**Documento de Visão**

**Projeto**: Sistema de Detecção Inteligente de Ameaças em Aplicações Web – Smart Detector

Controle de Versão

|  |  |  |  |
| --- | --- | --- | --- |
| **Versão** | **Data** | **Descrição da Alteração** | **Autor** |
| 1.0 | 10/09/2025 | Primeira versão do documento. Nesta versão foram inseridos os objetivos do presente documento, bem como os objetivos, escopo, premissas e tecnologias utilizadas no projeto. Além disso, também foram adicionados os requisitos funcionais e não funcionais do sistema. | Júlio Fraga |
| 1.1 | 11/09/2025 | Nesta versão foram adicionados os diagramas de caso de uso, classe e de sequência. | Júlio Fraga |
| 1.2 | 12/09/2025 | Nesta versão foi adicionado o diagrama de componentes e o modelo de entidade-relacionamento do banco de dados. Também foi atualizado o diagrama de classes. | Júlio Fraga |
| 1.3 | 16/09/2025 | Nesta versão foram alterados os diagramas de classe e o modelo de entidade-relacionamento do banco de dados. | Júlio Fraga |
| 1.4 | 18/09/2025 | Nesta versão foram alterados os requisitos RF006, RF008, RF009, RF010 e RNF004 e os diagramas de caso de uso e de componentes. Também foram adicionados os requisitos RF013 e RF014. | Júlio Fraga |
| 1.5 | 19/09/2025 | Nesta versão foram adicionados: estrutura de diretórios do projeto, as ferramentas utilizadas e os protótipos. | Júlio Fraga |
|  |  |  |  |

[1. Objetivo do Documento 3](#_Toc209177650)

[2. Objetivos do Projeto 3](#_Toc209177651)

[3. Escopo do Projeto 3](#_Toc209177652)

[4. Premissas do Projeto 3](#_Toc209177653)

[5. Tecnologias Utilizadas 4](#_Toc209177654)

[6. Requisitos do Sistema 4](#_Toc209177655)

[6.1 Requisitos funcionais 4](#_Toc209177656)

[6.2 Requisitos não funcionais 10](#_Toc209177657)

[7. Arquitetura do Sistema 12](#_Toc209177658)

[7.1. Diagrama de Caso de Uso 12](#_Toc209177659)

[7.2. Diagrama de Sequência 13](#_Toc209177660)

[7.3. Diagrama de Componentes 14](#_Toc209177661)

[7.4. Diagrama de Classes 15](#_Toc209177662)

[8. Banco de Dados – Entidade-Relacionamento 16](#_Toc209177663)

[9. Estrutura de Diretórios 18](#_Toc209177664)

[10. Protótipos 19](#_Toc209177665)

[11. Ferramentas Utilizadas no Projeto 21](#_Toc209177666)

# 1. Objetivo do Documento

Este documento tem como objetivo apresentar uma visão geral sobre o Sistema de Detecção Inteligente de Ameaças em Aplicações Web – Smart Detector, no que diz respeito ao painel de visualização do sistema. O foco está nos aspectos de interação, fluxo de dados e componentes envolvidos na exibição de eventos de anomalia em tempo real. O funcionamento do sistema é detalhado nas seções referentes aos requisitos funcionais e não funcionais.

# 2. Objetivos do Projeto

O projeto visa criar um painel que permita exibir eventos de possíveis anomalias detectadas por um IDS (Intrusion Detection System) em tempo real. Além disso, esses eventos devem ser armazenados para futuras consultas.

# 3. Escopo do Projeto

Fazem parte do escopo do projeto o painel web responsável por exibir os dados de eventos de possíveis anomalias, provenientes de um IDS, bem como armazenar esses dados no banco de dados. Também está no escopo do projeto um módulo de consulta de eventos passados e módulo de gestão de usuários.

Não fazem parte do escopo do projeto as definições de arquiteturas de IDS, bem como seu desenvolvimento.

# 4. Premissas do Projeto

Se tem como premissa para o projeto que um IDS esteja desenvolvido e disponibilize dados previamente analisados por IA, contendo informações sobre possíveis anomalias, que serão exibidas no painel e armazenadas no banco de dados. Enquanto o IDS não está disponível, o projeto pode utilizar dados simulados, seguindo a estrutura de dados previamente definidas para importação.

# 5. Tecnologias Utilizadas

* **Vue.js** (2.7.16) (Frontend): Interface de usuário que exibe eventos em tempo real, possibilita consulta de eventos anteriores e gerenciamento de usuários, além da tela de login no sistema.
* **PHP** (7.3.26) **/ Laravel** (8.83.29) (Backend): Responsável por autenticação, fornecimento de dados via endpoints, processamento e armazenamento no banco de dados de informações provenientes do serviço de detecção (IDS).
* **MySQL** (10.4.17 - MariaDB) (Banco de Dados): Utilizado para armazenamento de dados de usuários e histórico de eventos.
* **IDS** – Serviço de Detecção (Externo): Fonte de eventos de anomalias que alimentam o painel (não implementado neste projeto).

# 6. Requisitos do Sistema

## 6.1 Requisitos funcionais

|  |  |  |
| --- | --- | --- |
| **ID** | **TIPO** | **PRIORIDADE** |
| RF001 | Funcional | Alta |
| **TÍTULO** | | |
| Cadastro de Usuário | | |
| **DESCRIÇÃO** | | |
| O sistema deve permitir que um novo usuário seja cadastrado, fornecendo nome, e-mail, senha e perfil (usuário ou administrador). O e-mail deve ser único, portanto, não podem haver e-mails duplicados no sistema. | | |
| **CRITÉRIOS DE ACEITAÇÃO** | | |
| * Usuário preenche todos os campos e após submetê-los, o usuário é cadastrado com sucesso. * Usuário preenche todos os campos e após submetê-los, o sistema exibe um alerta, caso o e-mail já exista, não permitindo que o usuário seja cadastrado. | | |

|  |  |  |
| --- | --- | --- |
| **ID** | **TIPO** | **PRIORIDADE** |
| RF002 | Funcional | Alta |
| **TÍTULO** | | |
| Edição de Usuário | | |
| **DESCRIÇÃO** | | |
| O sistema deve permitir que um usuário existente tenha seus dados (exceto o e-mail) editados. | | |
| **CRITÉRIOS DE ACEITAÇÃO** | | |
| * Usuário existente no sistema é selecionado para edição e seus dados são alterados com sucesso. * Na tela de edição do usuário, o campo e-mail deve estar desabilitado para edição. | | |

|  |  |  |
| --- | --- | --- |
| **ID** | **TIPO** | **PRIORIDADE** |
| RF003 | Funcional | Alta |
| **TÍTULO** | | |
| Exclusão de Usuário | | |
| **DESCRIÇÃO** | | |
| O sistema deve permitir que um usuário existente seja excluído. | | |
| **CRITÉRIOS DE ACEITAÇÃO** | | |
| * Usuário existente no sistema é selecionado para exclusão e assim que a sua exclusão é confirmada, ele não deve mais estar cadastrado no sistema. | | |

|  |  |  |
| --- | --- | --- |
| **ID** | **TIPO** | **PRIORIDADE** |
| RF004 | Funcional | Média |
| **TÍTULO** | | |
| Consulta de Usuários | | |
| **DESCRIÇÃO** | | |
| O sistema deve permitir que o usuário administrador acesse uma tela de consulta de usuários, podendo filtrar por nome e/ou e-mail. | | |
| **CRITÉRIOS DE ACEITAÇÃO** | | |
| * Usuário administrador acessa a tela de consulta de usuários e, por padrão, serão exibidos 20 usuários, ordenados pela ordem alfabética do nome, de forma ascendente. * Usuário pode adicionar novo usuário. * Usuário pode editar um usuário existente. * Usuário pode excluir um usuário existente. | | |

|  |  |  |
| --- | --- | --- |
| **ID** | **TIPO** | **PRIORIDADE** |
| RF005 | Funcional | Alta |
| **TÍTULO** | | |
| Consulta de Usuários – Paginação | | |
| **DESCRIÇÃO** | | |
| A tela de consulta de usuários deve suportar paginação. | | |
| **CRITÉRIOS DE ACEITAÇÃO** | | |
| * Usuário administrador deve conseguir navegar pelos registros de usuários utilizando os controles de paginação. * Cada página deve exibir 20 registros de usuários. * Devem ser exibidos os seguintes controles de paginação: página atual (número), página anterior (número), próxima página (número), “Próximo »” e “« Anterior”. | | |

|  |  |  |
| --- | --- | --- |
| **ID** | **TIPO** | **PRIORIDADE** |
| RF006 | Funcional | Média |
| **TÍTULO** | | |
| Perfis de Usuários | | |
| **DESCRIÇÃO** | | |
| O sistema deve permitir dois tipos diferentes de perfis de usuários: administrador e usuário. Somente usuários com perfil de administrador poderão manipular usuários, além de todas as outras funcionalidades do sistema.  Usuários com perfil de usuário poderão visualizar o painel de eventos e fazer consultas no histórico de eventos, mas não poderão manipular usuários, exceto o seu próprio usuário. | | |
| **CRITÉRIOS DE ACEITAÇÃO** | | |
| * Usuário com perfil de administrador consegue manipular (inserir, editar e excluir) outros usuários no sistema. * Usuário com perfil de administrador consegue manipular (inserir, editar e excluir) classificações de risco e tipos de ameaças no sistema. * Usuário com perfil de usuário não consegue manipular (inserir, editar e excluir) outros usuários no sistema. * Usuário com perfil de usuário não consegue manipular (inserir, editar e excluir) classificações de risco e tipos de ameaças no sistema. * Usuário com perfil de usuário consegue editar o próprio usuário. | | |

|  |  |  |
| --- | --- | --- |
| **ID** | **TIPO** | **PRIORIDADE** |
| RF007 | Funcional | Alta |
| **TÍTULO** | | |
| Logout | | |
| **DESCRIÇÃO** | | |
| O sistema deve permitir que o usuário faça logout. | | |
| **CRITÉRIOS DE ACEITAÇÃO** | | |
| * Usuário faz logout do sistema e redirecionado para a tela de login. * Usuário não pode mais acessar o sistema até que faça login novamente. | | |

|  |  |  |
| --- | --- | --- |
| **ID** | **TIPO** | **PRIORIDADE** |
| RF008 | Funcional | Alta |
| **TÍTULO** | | |
| Armazenamento de Evento | | |
| **DESCRIÇÃO** | | |
| * O sistema deve armazenar no banco de dados cada evento identificado como possível anomalia/ameaça. O sistema deverá receber os dados do evento via um endopoint da API e processar a informação. Os dados armazenados para cada registro devem conter: Descrição, IP, Tipo de Ameaça, Classificação da Ameaça, Análise da IA, Origem Geográfica, Request, Data e hora do Evento. | | |
| **CRITÉRIOS DE ACEITAÇÃO** | | |
| * Após um evento ser recebido pelo endpoint da API, as informações do evento devem estar armazenadas no banco de dados. * Se a classificação de risco informada não estiver previamente cadastrada no sistema, o evento não deverá ser armazenado e a seguinte mensagem de erro deve ser retornada “Classificação informada não foi encontrada no sistema, tente novamente“. * Se o tipo de ameaça informado não estiver previamente cadastrado no sistema, o sistema deverá cadastrar automaticamente esse novo tipo de ameaça e então armazenar o evento. | | |

|  |  |  |
| --- | --- | --- |
| **ID** | **TIPO** | **PRIORIDADE** |
| RF009 | Funcional | Alta |
| **TÍTULO** | | |
| Painel de Exibição dos Eventos | | |
| **DESCRIÇÃO** | | |
| O sistema deve exibir em tempo real os últimos 100 eventos identificados como possíveis anomalias/ataques. Esses eventos devem exibir dados como Descrição, IP, Tipo de Ameaça, Classificação de Risco, Data e hora. | | |
| **CRITÉRIOS DE ACEITAÇÃO** | | |
| * Ao identificar um evento, ele deve ser imediatamente exibido no painel. * As informações do evento estão visíveis e claras ( Descrição, IP, Tipo de Ameaça e Classificação de Risco). * Devem ser listados os últimos 100 eventos (ordenados do mais recente para o mais antigo). | | |

|  |  |  |
| --- | --- | --- |
| **ID** | **TIPO** | **PRIORIDADE** |
| RF010 | Funcional | Média |
| **TÍTULO** | | |
| Consulta de Eventos | | |
| **DESCRIÇÃO** | | |
| O sistema deve permitir ao usuário acessar uma tela de consulta de eventos, podendo filtrar por Data, Hora, Descrição e Origem Geográfica. | | |
| **CRITÉRIOS DE ACEITAÇÃO** | | |
| * Usuário acessa a tela de consulta de eventos e, por padrão, serão exibidos os últimos 20 eventos, ordenados do mais recente para o mais antigo. * Usuário acessa a tela de consulta de eventos e informa os filtros desejados, então o sistema dever exibir os resultados correspondentes. | | |

|  |  |  |
| --- | --- | --- |
| **ID** | **TIPO** | **PRIORIDADE** |
| RF011 | Funcional | Alta |
| **TÍTULO** | | |
| Consulta de Eventos - Paginação | | |
| **DESCRIÇÃO** | | |
| A tela de consulta de eventos deve suportar paginação, devido ao grande número de eventos. | | |
| **CRITÉRIOS DE ACEITAÇÃO** | | |
| * Usuário deve conseguir navegar pelos registros de eventos utilizando os controles de paginação. * Cada página deve exibir 20 registros de eventos. * Devem ser exibidos os seguintes controles de paginação: página atual (número), página anterior (número), próxima página (número), “Próximo »” e “« Anterior”. | | |

|  |  |  |
| --- | --- | --- |
| **ID** | **TIPO** | **PRIORIDADE** |
| RF012 | Funcional | Média |
| **TÍTULO** | | |
| Consulta Detalhada de Evento | | |
| **DESCRIÇÃO** | | |
| Usuário pode consultar informações mais detalhadas de um evento acessando-o através da tela de consulta de eventos e/ou do painel principal de exibição de eventos. | | |
| **CRITÉRIOS DE ACEITAÇÃO** | | |
| * No painel de exibição de eventos, usuário clicará em uma opção que abrirá uma nova aba com todas as informações do evento. * Na tela de consulta de eventos, usuário clicará em uma opção que abrirá uma nova aba com todas as informações do evento. | | |

|  |  |  |
| --- | --- | --- |
| **ID** | **TIPO** | **PRIORIDADE** |
| RF013 | Funcional | Média |
| **TÍTULO** | | |
| Cadastro de Classificação de Risco | | |
| **DESCRIÇÃO** | | |
| O sistema deve permitir que o usuário administrador acesse uma tela de consulta de classificações de risco, podendo filtrar por descrição. Além disso, o usuário administrador poderá manipular (inserir, editar e excluir) uma classificação de risco. | | |
| **CRITÉRIOS DE ACEITAÇÃO** | | |
| * Usuário administrador acessa a tela de consulta de classificação de risco e, por padrão, serão exibidos 20 registros, ordenados pela data de criação, de forma ascendente. * Usuário pode adicionar nova classificação de risco. * Usuário pode editar uma classificação de risco existente. * Usuário pode excluir uma classificação de risco existente. * Página deve suportar paginação. | | |

|  |  |  |
| --- | --- | --- |
| **ID** | **TIPO** | **PRIORIDADE** |
| RF014 | Funcional | Média |
| **TÍTULO** | | |
| Cadastro de Tipo de Ameaça | | |
| **DESCRIÇÃO** | | |
| O sistema deve permitir que o usuário administrador acesse uma tela de consulta de tipos de ameaça, podendo filtrar por descrição. Além disso, o usuário administrador poderá manipular (inserir, editar e excluir) um tipo de ameaça. | | |
| **CRITÉRIOS DE ACEITAÇÃO** | | |
| * Usuário administrador acessa a tela de consulta de tipo de ameaça e, por padrão, serão exibidos 20 registros, ordenados pela data de criação, de forma ascendente. * Usuário pode adicionar novo tipo de ameaça. * Usuário pode editar um tipo de ameaça existente. * Usuário pode excluir um tipo de ameaça existente. * Página deve suportar paginação. | | |

## 6.2 Requisitos não funcionais

|  |  |  |
| --- | --- | --- |
| **ID** | **TIPO** | **PRIORIDADE** |
| RNF001 | Não Funcional | Alta |
| **TÍTULO** | | |
| Armazenar Senhas Criptografadas | | |
| **DESCRIÇÃO** | | |
| O sistema deve armazenar as senhas de usuários de forma criptografada. | | |
| **CRITÉRIOS DE ACEITAÇÃO** | | |
| * A senha não pode ser armazenada em texto no banco de dados. * O banco de dados deve conter apenas o hash da senha. * Ao acessar a tabela de usuários, no banco de dados, a senha não deve estar em um formato legível. | | |

|  |  |  |
| --- | --- | --- |
| **ID** | **TIPO** | **PRIORIDADE** |
| RNF002 | Não Funcional | Alta |
| **TÍTULO** | | |
| Autenticação JWT | | |
| **DESCRIÇÃO** | | |
| A API deve utilizar autenticação JWT. | | |
| **CRITÉRIOS DE ACEITAÇÃO** | | |
| * Todas as rotas protegidas da API devem rejeitar requisições sem um token válido. * O token deve expirar 2 horas após a sua criação. * Um token expirado deve ser considerado como token inválido, assim, não permitirndo que uma rota protegida da API possa ser acessada através dele. | | |

|  |  |  |
| --- | --- | --- |
| **ID** | **TIPO** | **PRIORIDADE** |
| RNF003 | Não Funcional | Média |
| **TÍTULO** | | |
| Utilizar Login Rate Limiting | | |
| **DESCRIÇÃO** | | |
| O sistema deve implementar login rate limiting, para evitar ataques de brute force no login. | | |
| **CRITÉRIOS DE ACEITAÇÃO** | | |
| * Após 5 tentativas consecutivas de login com falhas, o sistema deve bloquear novas tentativas do mesmo IP/Usuário por 10 minutos. * O bloqueio temporário deve ser informado ao usuário com uma mensagem clara. | | |

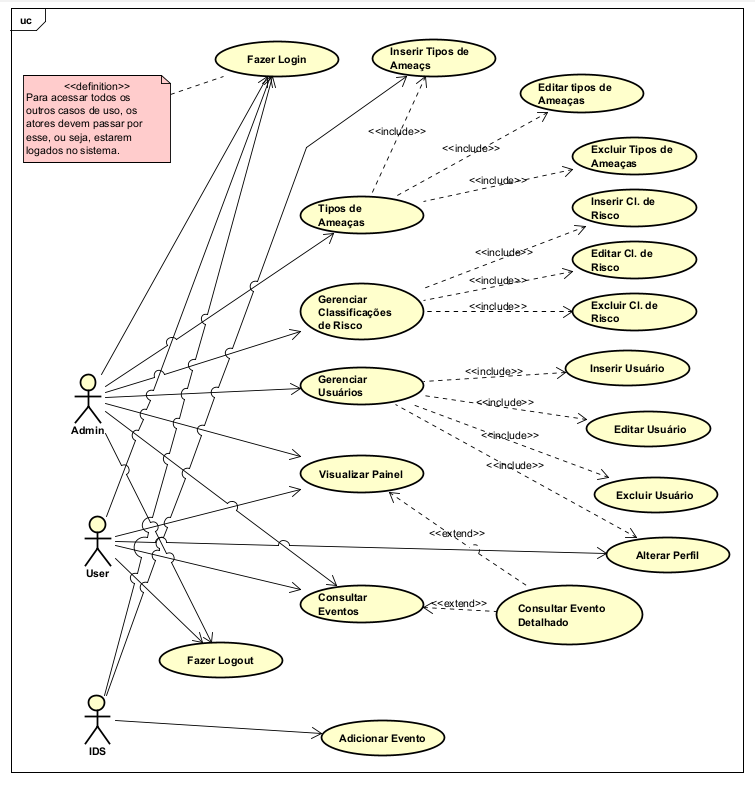
|  |  |  |
| --- | --- | --- |
| **ID** | **TIPO** | **PRIORIDADE** |
| RNF004 | Não Funcional | Alta |
| **TÍTULO** | | |
| Latência de Eventos | | |
| **DESCRIÇÃO** | | |
| O sistema deve ser capaz de exibir os eventos em tempo real, com latência máxima de 10 segundos entre o evento detectado e sua exibição. | | |
| **CRITÉRIOS DE ACEITAÇÃO** | | |
| * Eventos identificados pelo sistema devem aparecer no painel de eventos em até 10 segundos. * O painel deve atualizar automaticamente sem necessidade de refresh manual. | | |

|  |  |  |
| --- | --- | --- |
| **ID** | **TIPO** | **PRIORIDADE** |
| RNF005 | Não Funcional | Média |
| **TÍTULO** | | |
| Interface Responsiva | | |
| **DESCRIÇÃO** | | |
| O sistema deve ter uma interface responsiva, acessível em desktops, tablets e dispositivos móveis. | | |
| **CRITÉRIOS DE ACEITAÇÃO** | | |
| * O sistema ajusta automaticamente o layout confirme a largura da tela. * Em telas pequenas, o menu deve se adaptar ao estilo hamburguer. * Painel de Eventos, Consulta de Eventos e Gestão de Usuários estão acessíveis e funcionais, independentemente se acessos via desktop, tablet ou dispositivo móvel. | | |

# 7. Arquitetura do Sistema

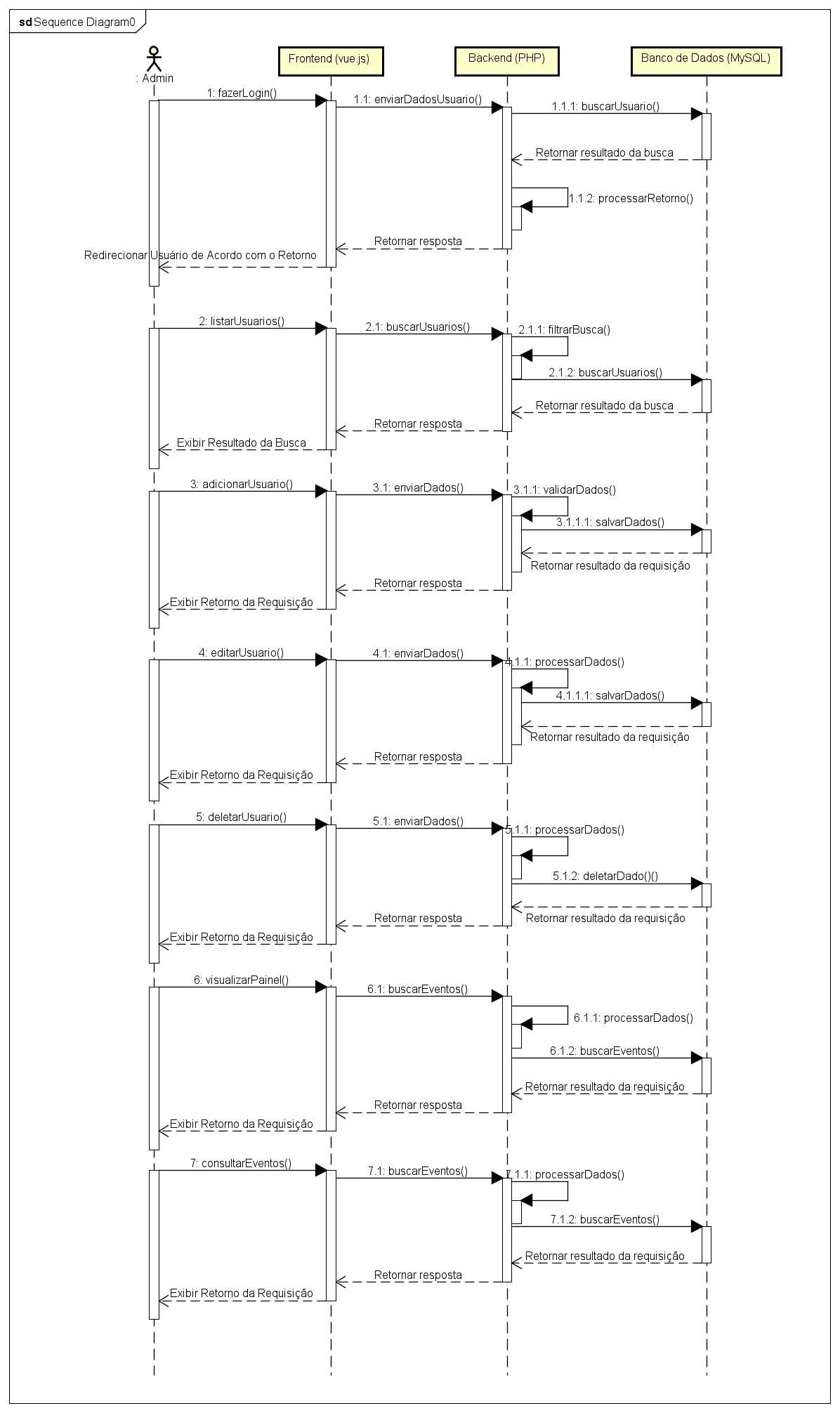
## 7.1. Diagrama de Caso de Uso

O diagrama de caso de uso representa as principais interações dos atores com o sistema. Para este projeto, as interações dos atores (tipos de usuários) estão demonstradas no diagrama abaixo:



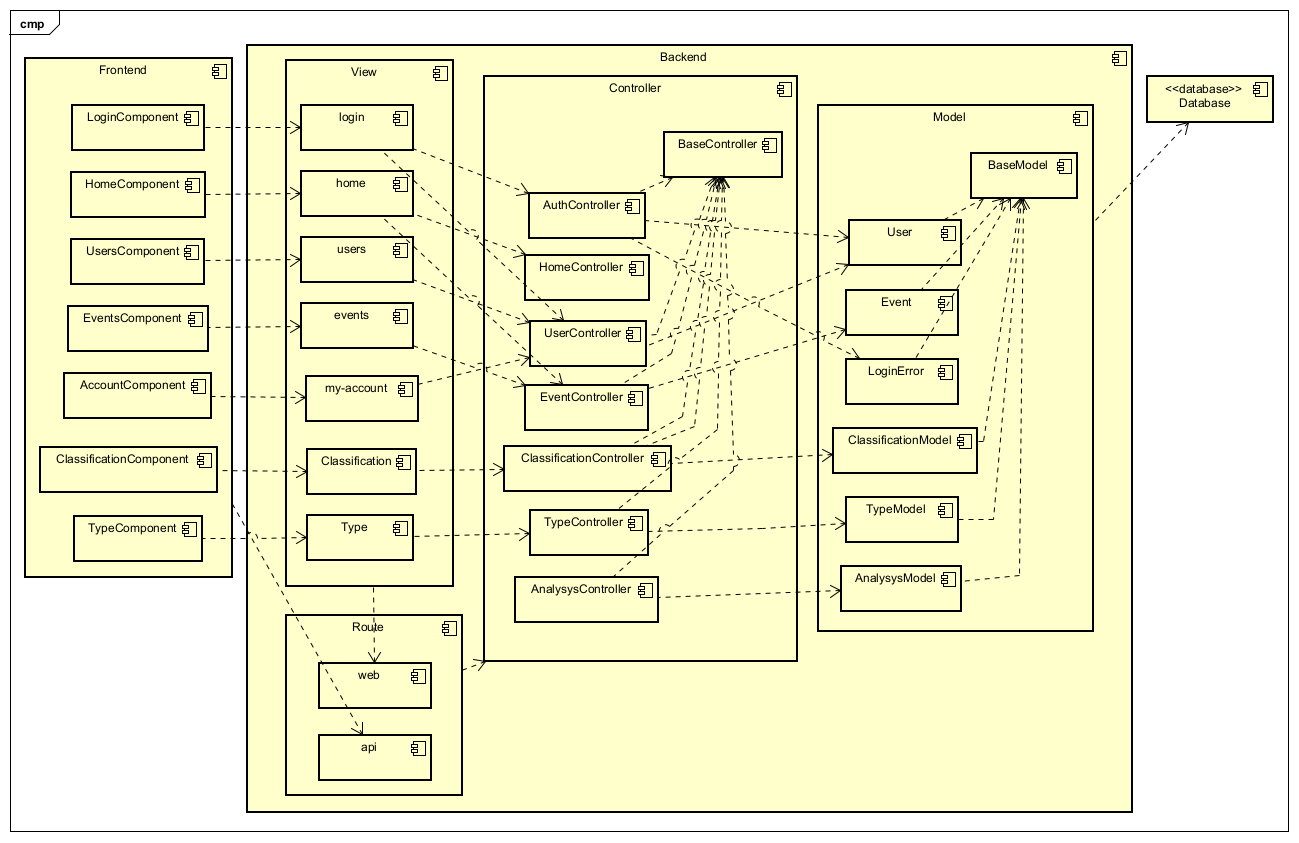
## 7.2. Diagrama de Sequência

O diagrama de sequência é utilizado para representar a sequência de processos em um sistema, demonstrando como funcionará a interação entre os atores e os componentes/objetos.



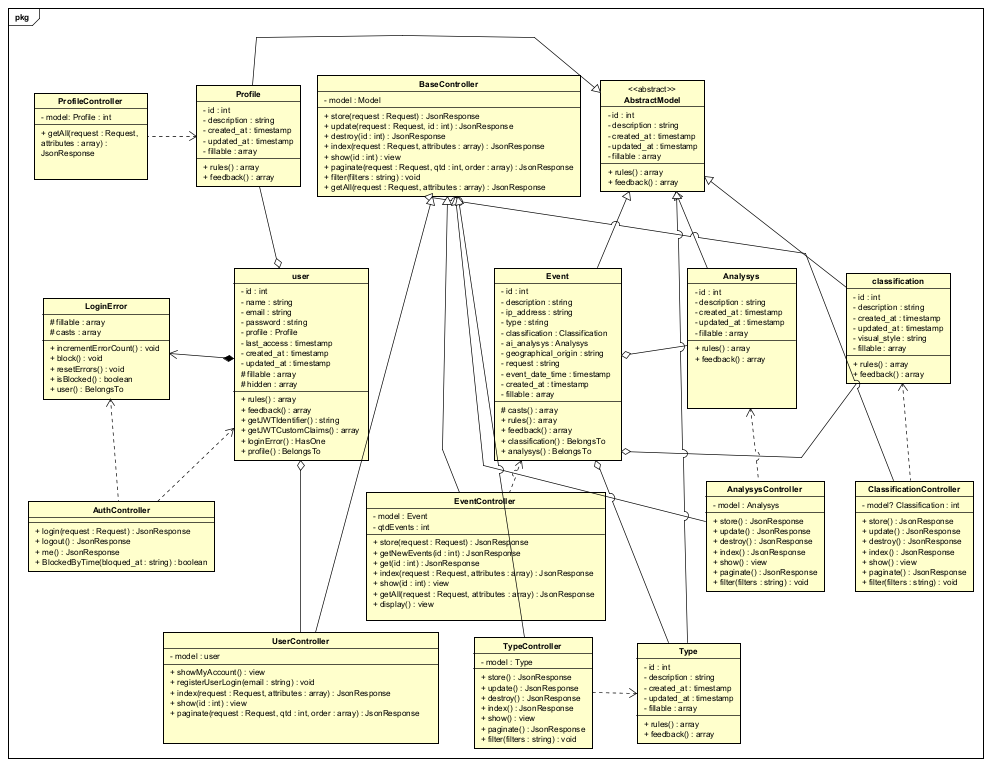
## 7.3. Diagrama de Componentes

O diagrama de componentes é uma visualização da estrutura de um sistema, onde são demonstrados os seus componentes e as relações ou dependências entre eles.



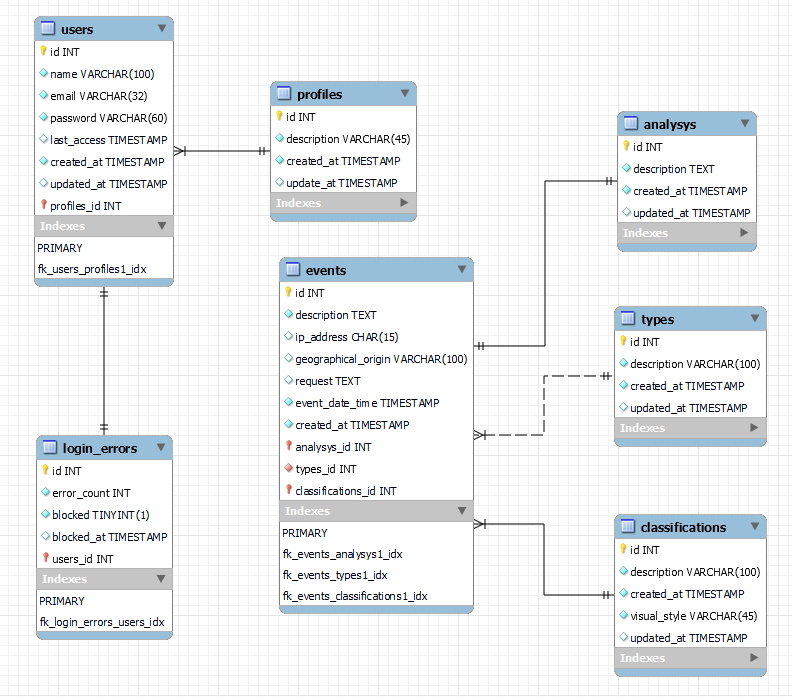
## 7.4. Diagrama de Classes

O diagrama de classes é uma demonstração da estrutura das classes do sistema, mostrando seus atributos e funções (métodos). Além disso, demonstra como elas estão relacionadas entre si.

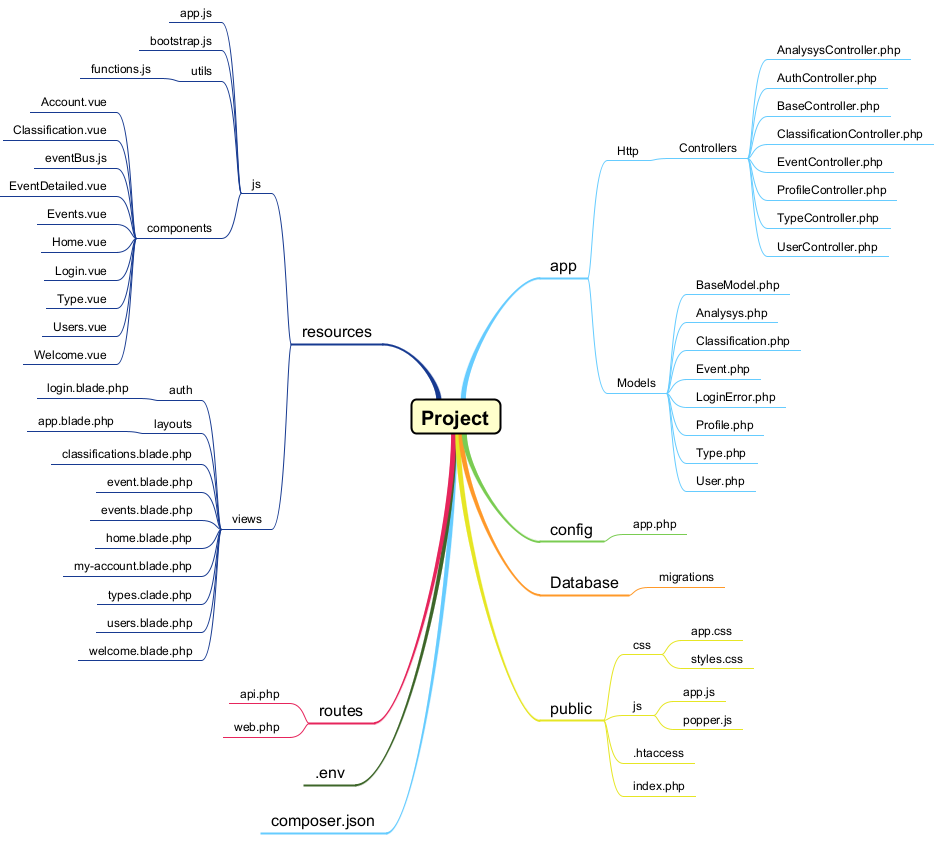


# 8. Banco de Dados – Entidade-Relacionamento

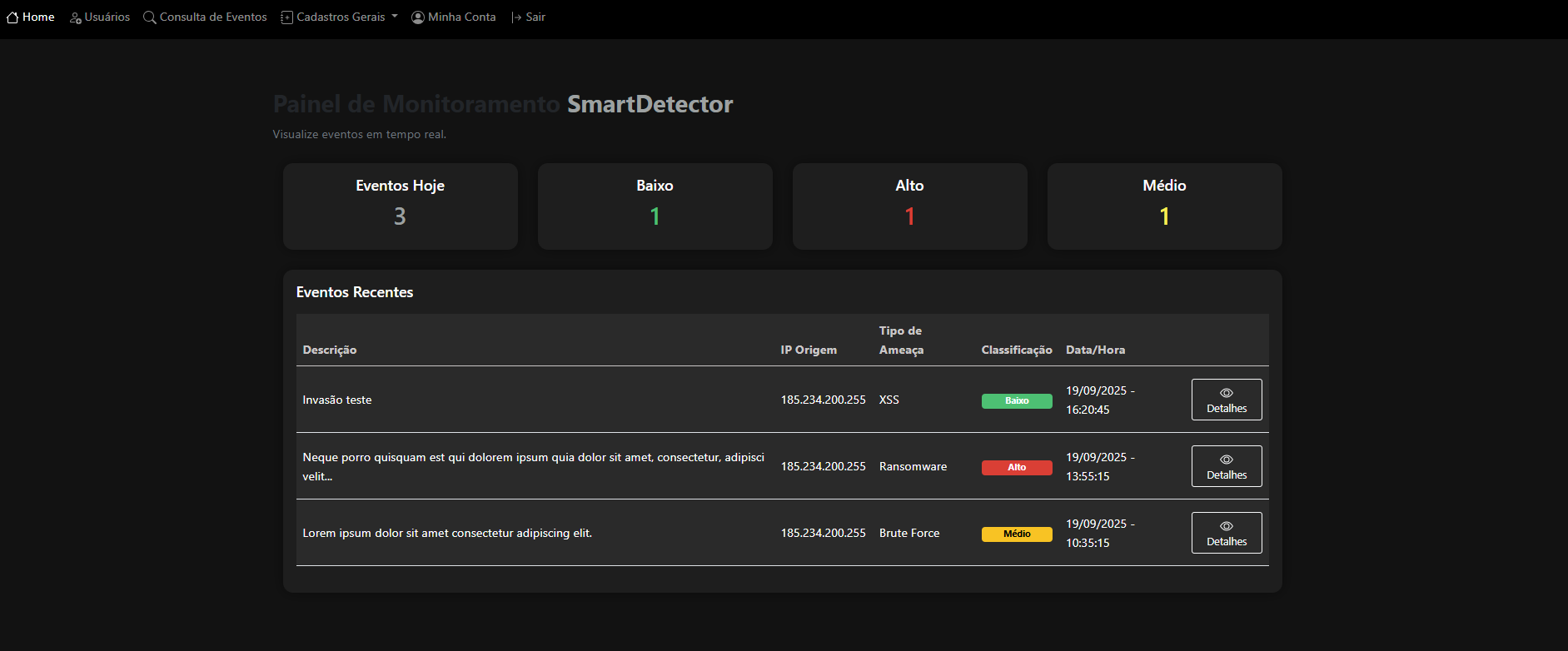
O modelo de entidade-relacionamento é uma representação que descreve as entidades de um sistema, seus atributos e os relacionamentos entre elas. Normalmente serve para compreender e estruturar os dados utilizados em um sistema e serve como base para a criação de um modelo lógico (tabelas, colunas e chaves).

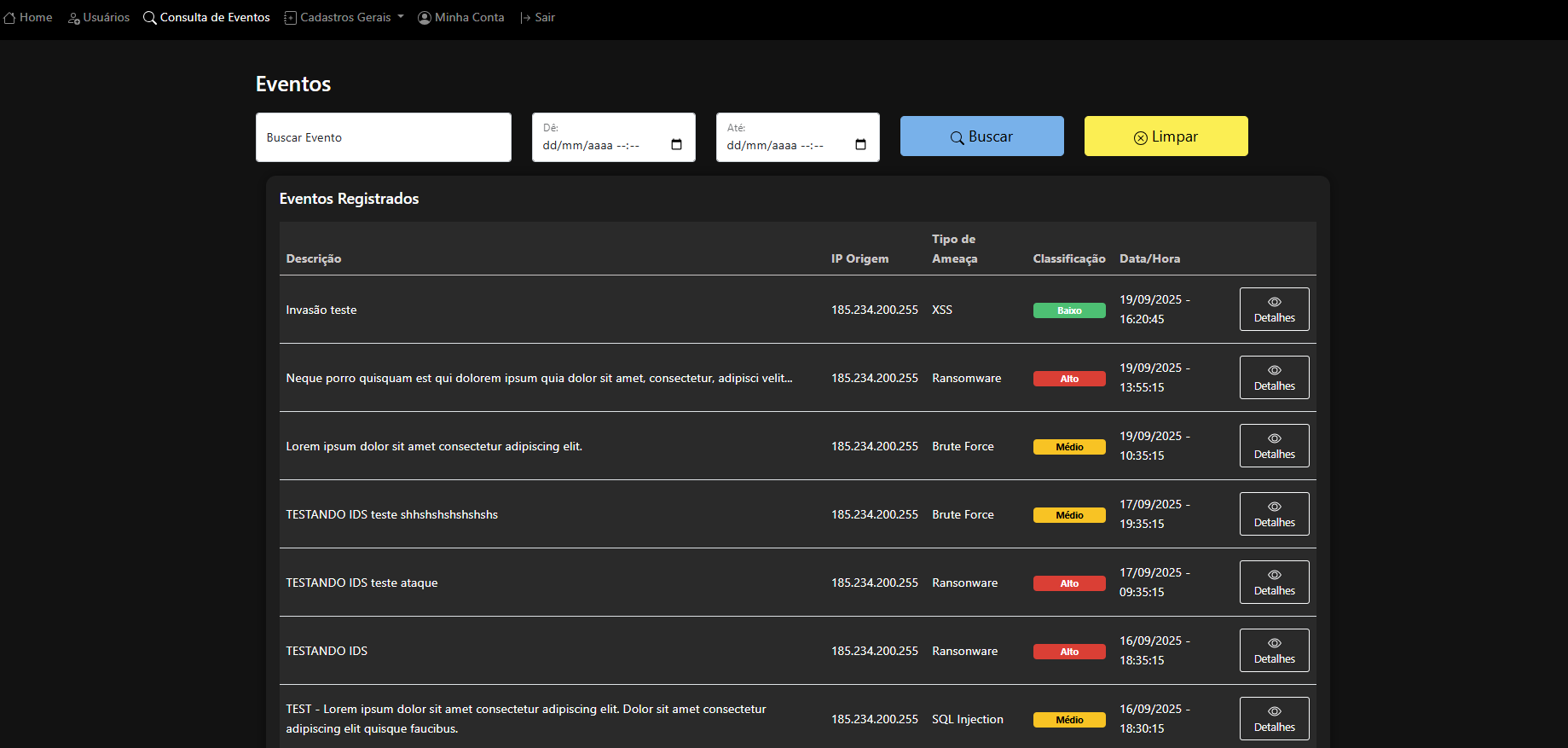


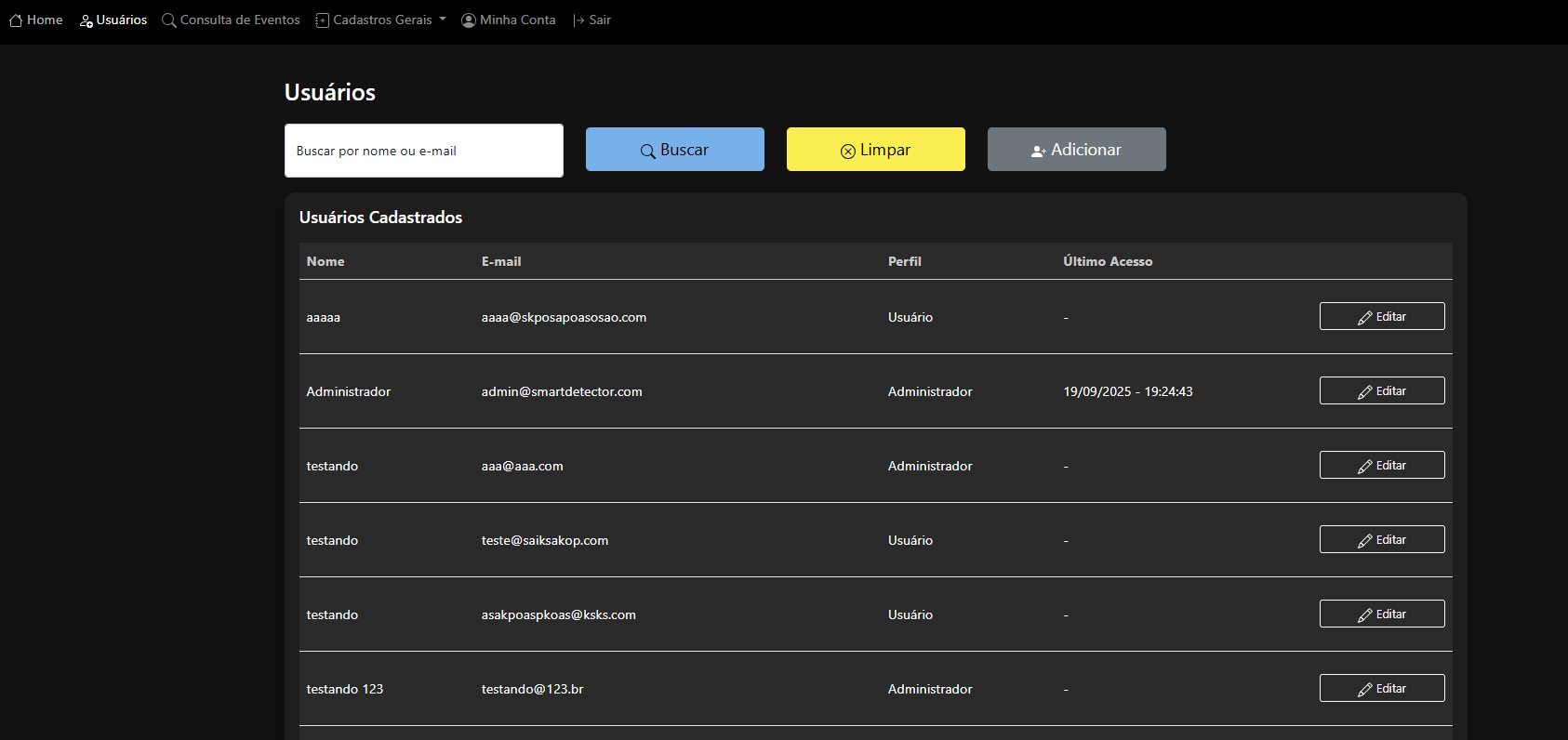
# 9. Estrutura de Diretórios

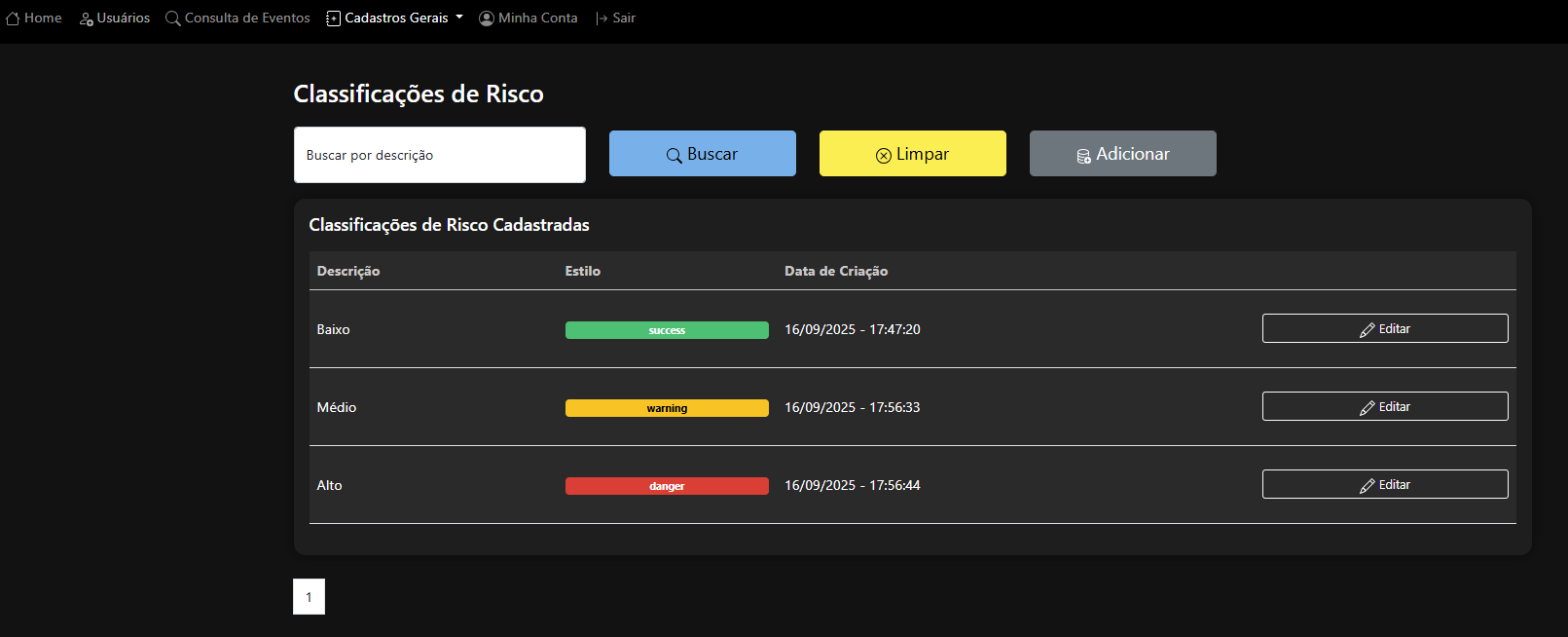
Abaixo, vemos como o projeto está estruturado no que diz respeito a estrutura dos diretórios e principais arquivos do projeto:  


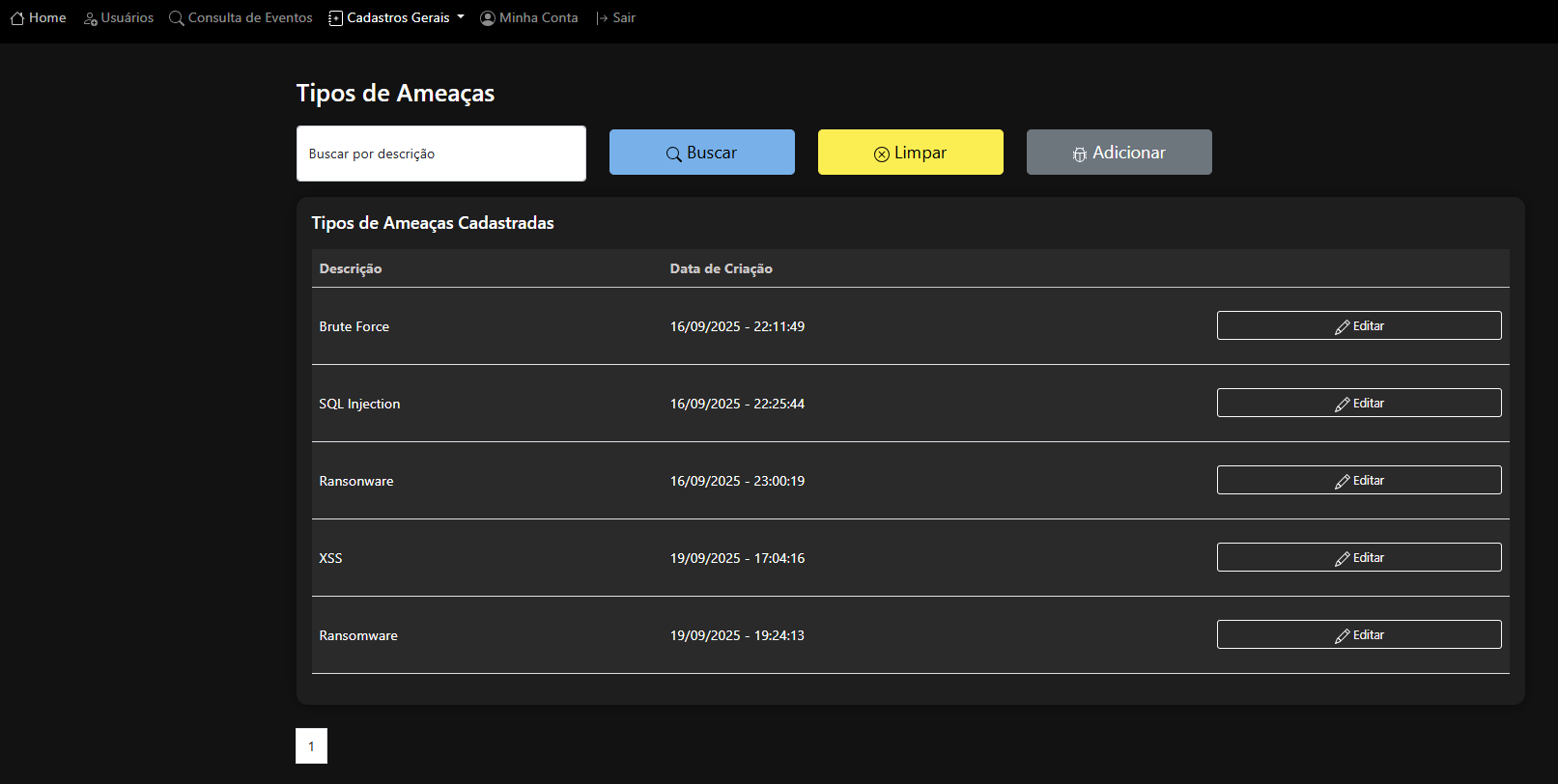
# 10. Protótipos

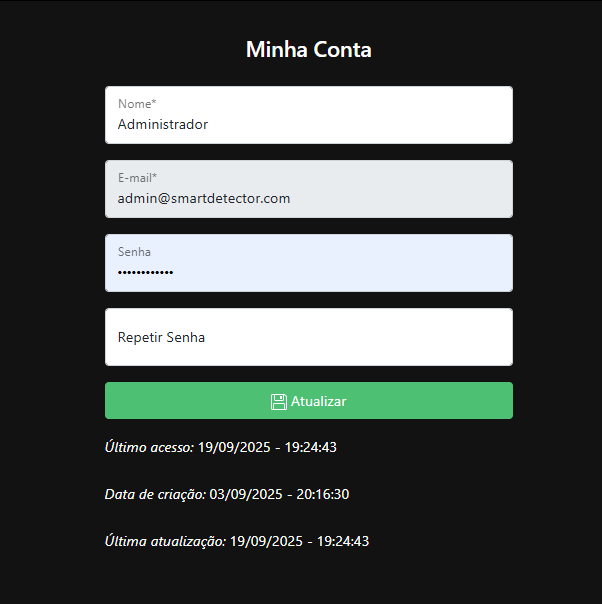












# 11. Ferramentas Utilizadas no Projeto

1. No desenvolvimento deste projeto, serão utilizadas as seguintes ferramentas:
   * Visual Studio Code: editor de código-fonte para o desenvolvimento do sistema;
   * GitHub: plataforma para versionamento de código;
   * Xampp: software que inclui servidor web Apache, PHP e banco de dados MySQL para criar um ambiente de desenvolvimento;
   * Astah UML: software para criação de diagramas UML;
   * MySQL Workbench: software para criação do modelo entidade-relacionamento do banco de dados e que, além disso, também possui um SGBD, que permite gerenciamento do banco de dados;
   * Postman: software para simular manipulação de dados através dos endpoints da API.