

# Documentação

- [Visão Geral do Produto](#)
- [Diagrama de Classe](#)
- [Histórias de Usuário](#)

# Visão Geral do Produto

## Problema

A Defensoria Pública do Distrito Federal (**DPDF**) realiza periodicamente eventos itinerantes e mutirões de atendimento para ampliar o acesso da população aos seus serviços. Estes eventos são caracterizados por um alto volume de cidadãos e uma dinâmica de atendimento rápida.

O método atual de registro e controle dos atendimentos realizados durante esses eventos é manual e fragmentado, frequentemente dependendo de formulários em papel ou planilhas eletrônicas não integradas. A ausência de um sistema centralizado e digital para capturar as informações em tempo real cria um gargalo operacional significativo.

## Impactos e Consequências

- **Inconsistência e Perda de Dados:** O registro manual está sujeito a erros de preenchimento, ilegibilidade e extravio de formulários, comprometendo a integridade e a confiabilidade das informações dos assistidos.
- **Falta de Rastreabilidade:** Torna-se extremamente difícil ou impossível realizar o acompanhamento de um caso registrado em um evento, pois não há um identificador único e um repositório central para consulta do histórico.
- **Dificuldade na Geração de Relatórios:** A consolidação dos dados para fins estatísticos e gerenciais é um processo lento, trabalhoso e impreciso.
- **Comprometimento da Análise Estratégica:** Sem dados estruturados, a gestão fica impossibilitada de analisar a demanda por tipo de serviço, perfil do público e localização geográfica, informações cruciais para o planejamento estratégico e a alocação de recursos em futuros eventos.

## Declaração de posição do produto

A plataforma Aurora foi concebida e implementada com sucesso para a gestão de atendimentos do evento **Dia da Mulher**. No entanto, a eficácia demonstrada pela solução gerou uma demanda expressiva por parte de outras coordenações da Defensoria Pública.

A nova versão do Aurora propõe uma evolução estratégica: transformar o sistema de uma aplicação de propósito específico para uma plataforma genérica e configurável. O objetivo é desacoplar as funcionalidades exclusivas do **Dia da Mulher**, convertendo-as em parâmetros e módulos customizáveis. Isso permitirá que o Aurora seja rapidamente adaptado e implantado em qualquer evento ou mutirão promovido pela DPDF, otimizando recursos e padronizando a coleta de dados em todas as iniciativas.

## Objetivos do Produto

Nosso objetivo é desenvolver uma nova versão da plataforma Aurora, uma solução abrangente que otimiza a eficiência operacional dos eventos da Defensoria Pública do Distrito Federal (DPDF) e moderniza a interação entre a instituição e os cidadãos assistidos em ações itinerantes.

Para alcançar este propósito, a plataforma é projetada para:

- **Digitalizar e centralizar o registro de atendimentos**, assegurando a rastreabilidade completa dos casos e a integridade das informações coletadas em campo.
- **Transformar dados operacionais em inteligência estratégica**, oferecendo ferramentas eficientes para a geração de relatórios e análises de impacto dos eventos.
- **Abstrair as particularidades dos eventos**, permitindo que a plataforma seja dinamicamente configurada para diferentes tipos de mutirões (saúde, consumidor, população de rua, etc.) sem a necessidade de novo desenvolvimento.
- **Capacitar os gestores de eventos com autonomia**, fornecendo uma interface administrativa onde possam configurar os fluxos e formulários de atendimento de acordo com as necessidades específicas de cada ação.

Com essas funcionalidades, buscamos não apenas otimizar a alocação de recursos e a eficiência dos mutirões, mas também fortalecer a capacidade da Defensoria Pública de tomar decisões estratégicas baseadas em dados, ampliando o alcance e a qualidade do serviço prestado à população do Distrito Federal.

## Tecnologias a Serem Usadas

### Frontend

- ReactJs com Vite
- Typescript
- TailwindCSS
- HeroUI (Biblioteca CSS)

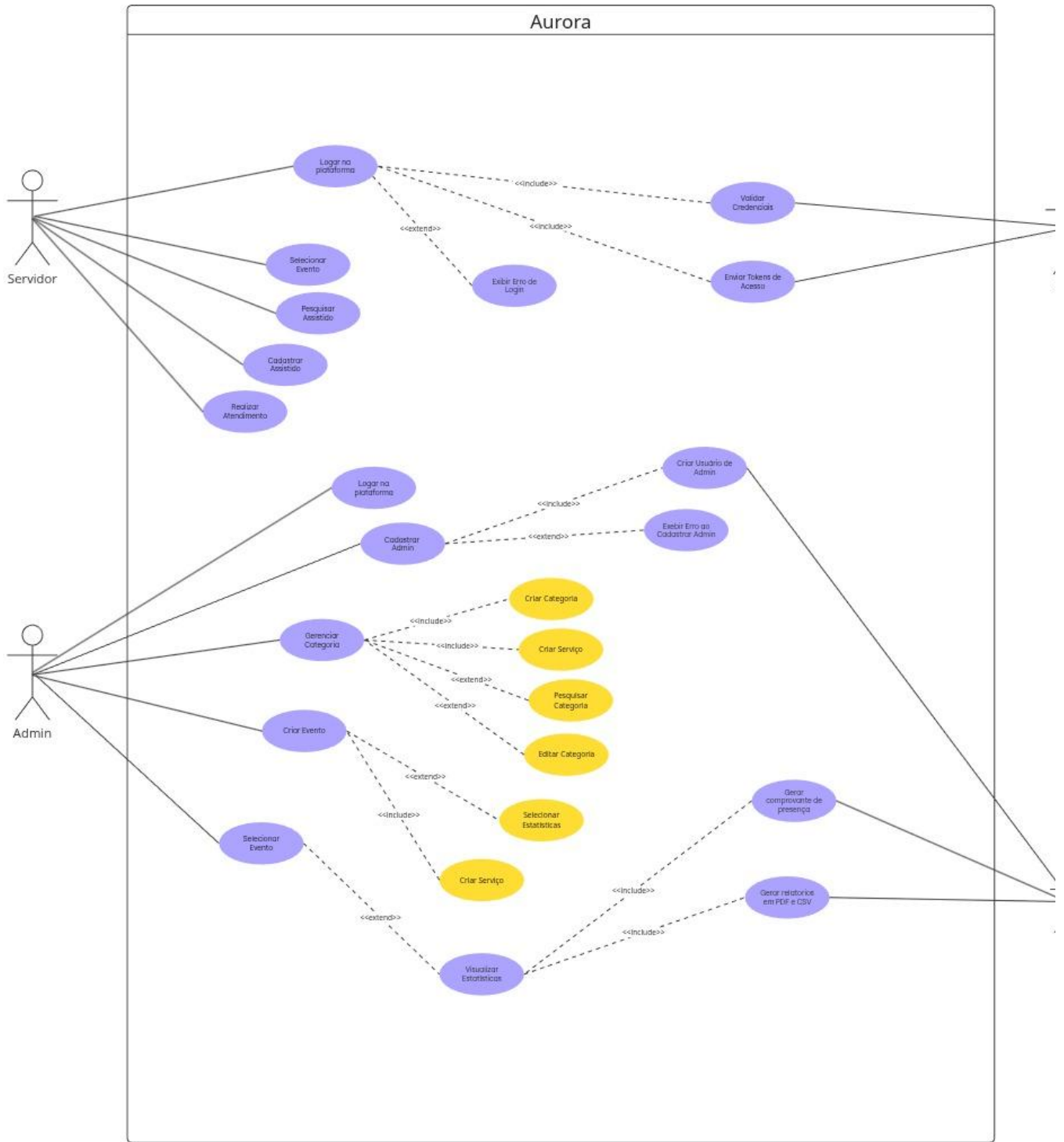
### API

- NodeJS com Typescript
- ExpressJS
- Swagger (Documentação)
- TypeORM (ORM para o banco de dados)
- JsonWebToken (JWT - Gerenciamento de Tokens de acesso)
- API REST

### Banco de Dados

- MySQL Server

## Diagrama de Casos de Uso



# Diagrama de Classe

# Histórias de Usuário

## US00 - Criação de Qualificações

**Eu como** administrador do Sistema,  
**quero** criar qualificações ao criar categorias  
**para** assim poder reaproveitá-las em eventos futuros.

### Qualificação:

- **Título:** Ex: SEMOB, CAESB
- **Categoria:** Ex: Dia da Mulher, Dia do Cidadão

### Critérios de aceitação:

- - Qualificações não podem ter o mesmo nome
  - 
  - Nenhum campo pode ser nulo
  - 
  - Toda qualificação deve estar associada a uma categoria
  - 
  - Deve ser possível deletar qualificações
- 

## US01 - Criação de Categorias

**Eu como** Administrador do Sistema,  
**quero** criar categorias  
**para** poder identificar tipos de eventos e suas particularidades, assim podendo reutilizar configurações para eventos recorrentes.

### Categoria:

- **Nome:** Ex: Dia da Mulher, Dia do Cidadão
- **Qualificações Base:**
  - Deverão ser cadastradas no momento da criação da categoria
  - Com ao menos 1 qualificação por categoria
  - Deverá ser possível deletar e criar qualificações
- **Estatísticas Relevantes:** Seleção de estatísticas relevantes para aquele tipo de evento:
  - Pessoas Atendidas
  - Parceiros
  - Serviços por Pessoa
  - Atendimentos
  - Voluntários
  - Lanches Entregues
- **Eventos Criados com a Categoria:** Apenas para facilitar eventuais queries

#### Critérios de aceitação:

- 
- Apenas usuários do Tipo Admin podem criar categorias
- 
- Nenhum campo pode ser nulo
- 
- Deverá ser possível editar as qualificações bases para eventos futuros
- 
- Deverá ser selecionado ao menos 6 estatísticas relevantes (já devem vir selecionadas por padrão)
- 
- As características serão criadas depois, em uma janela separada (página separada)

---

## US02 - Buscar Assistido

**Eu como** usuário Servidor (Atendente),

**quero** pesquisar um assistido

**para** que eu possa gerar sua ficha de atendimento para rastrear os serviços do meu evento.

#### Detalhes:

- Pesquisar um assistido que está no banco de dados do SOLAR.

#### Critérios de aceitação:

-

- Deverá ser possível pesquisar por:
    - Nome
    - CPF
  - 
  - Ambos os casos de pesquisa devem ser feitos pelo mesmo input
  - 
  - O sistema deverá apresentar todos os resultados encontrados, porém é necessário pensar em uma maneira (talvez paginação) para evitar pesar muito a página
- 

## US03 - Cadastrar Assistido

**Eu como** usuário Servidor (Atendente),

**quero** cadastrar um assistido

**para** que eu possa gerar sua ficha de atendimento para rastrear em meu evento.

### Detalhes:

- O cadastro deverá ser feito no sistema SOLAR, devendo estar disponível tanto no Aurora quanto no Solar

### Campos para cadastro:

- Nome
- Nome da Mãe
- CPF
- Telefone
- Data de Nascimento

### Critérios de aceitação:

- 
- Campos obrigatórios:
  - Nome
  - Nome da mãe
- 
- Nome e nome da mãe deverão aceitar apenas letras
- 
- CPF deverá aceitar apenas números e formatar o campo automaticamente para
-

- Telefone deverá aceitar apenas números e formatar o campo automaticamente para (xx) xxxxx-xxxx
- 

# US04 - Cadastro de Atendimento

**Eu como** servidor (Atendente),  
**quero** cadastrar um atendimento de um assistido  
**para** que eu possa rastrear as estatísticas do evento futuramente.

## Detalhes:

- Após a pesquisa de assistidos, o servidor deverá selecionar o assistido desejado e preencher quais serviços o mesmo irá usar durante o evento.

## Critérios de aceitação:

- 
- Um atendimento só pode ser feito em dia de evento e **SE O EVENTO ESTIVER COM O STATUS EM\_ANDAMENTO**

## △ Atenção:

O evento é identificado pela URL, portanto devem ser implementados avisos e travas para que, se o usuário tentar acessar um evento que não está em andamento diretamente pela URL, o sistema bloqueie a ação.

- 
- Apenas usuários logados e com sessão válida no Solar (5 min) poderão registrar atendimentos (implementar refresh do Solar)
- 
- O atendimento deverá ter no mínimo 1 serviço
- 
- O atendimento deverá vir com os dados do assistido já preenchidos e visíveis
- 
- O sistema deverá apresentar erros claros, indicando os motivos de tal erro
- 
- Caso o usuário crie um assistido, ele deverá ser automaticamente redirecionado para a tela de atendimento
-

Após a finalização, o atendimento deverá estar disponível imediatamente no banco de dados, para uso na página de estatísticas do admin

---