

# Requisitos Funcionais e Não Funcionais

- [Aurora - Requisitos](#)

# Aurora - Requisitos

## Requisitos Funcionais

- Servidor:
  - Login com credenciais do Solar
  - Seleciona o Evento
  - Pesquisa pelo assistido(a) (Solar)
    - Caso não encontre, pode o registrar (Solar)
  - Registra o atendimento
  - **Comprovantes de voluntário \***
  
- Admin
  - Login com credenciais próprias do Aurora
  - Acesso ao painel de Admin
  - Criação de eventos
    - O admin cria uma “Categoria” para o evento. Ex: Dia do Cidadão, Dia da Mulher
    - A categoria vai ser cadastrada e terá “Presets”, os presets serão qualificações pré-definidas, que podem ou não ser usadas em um cadastro de novo evento, servindo como base.
    - Após a criação da categoria, o evento poderá ser criado, usando da categoria como base, evitando o recadastro de serviços.
    - O evento contém:
      - Categoria
      - Data
      - Qualificações
  - Estatísticas
    - O admin deve ser capaz de visualizar estatísticas por evento.
    - Também deverá ser possível extrair as estatísticas para um PowerBI, por isso a aplicação deverá usar SQL Server como banco, para facilitar esse processo.
    - O admin seleciona um evento já cadastrado

- Cai no painel de admin, onde será possível visualizar as estatísticas de cada evento.
- Deverá ser possível ver:
  - Serviços associados
  - Ver atendimentos
  - Indicadores
    - Servidores
    - Serviços
    - Atendimentos gerados no Solar
    - Data do evento
  - Estatísticas detalhadas
    - Pessoas atendidas
    - Parceiros
    - Média de Serviços por pessoa
    - Atendimentos
    - Voluntários
  - Alguns serviços podem precisar de estatísticas diferentes das comuns a outros eventos, uma solução pode ser o sistema disponibilizar alguns campos que podem ser adicionados ou removidos das estatísticas, por exemplo:
    - Cor da Pele
    - Lanches Entregues.
  - Esses campos devem estar prontos no sistema, e devem funcionar de maneira modular, permitindo com que sejam selecionados ou não, sem prejuízos a outros eventos.
- Logs do sistema, toda operação deverá gerar um log que seja rastreável, assim podendo ser filtrado, pesquisado e etc.
- O log deve conter:
  - Rash (Identificador)
  - Ação
  - Usuário
  - Data e Hora

## Requisitos Não Funcionais

- Design Moderno, inspirado no Liquid Glass e Frosted Glass.

- O Banco de Dados deverá ser SQL Server, para facilitar a integração com o PowerBI.
- Aplicação deverá ser modular, evitando configurações hardcoded
- A aplicação deverá conter ampla segurança, portanto:
  - Operações devem ser realizadas na API, mesmo chamadas que sejam chamadas diretas ao solar.
  - Tokens JWT com assinatura
  - Verificação de permissões
  - A comunicação deverá ser feita por HTTPS, ao menos da comunicação interna do Aurora.
  - Rate Limiting, para evitar ataques de força bruta.
  - CORS restrito e bem configurado
  - Proteção CSRF
  - Proteção XSS
  - Headers de segurança (HelmetJS)
  - Não deve ser feito o armazenamento de qualquer informação sensível em localStorage/sessionStorage
  - Cookies apenas HTTPOnly, com flag secure (PROD)
  - Em produção, não expor portas do banco ou api, tudo ser feito pelo traefik ou nginx.
  - Travas de carregamento para evitar spam de requests.
- CI/CD para checagem automática de erros e segurança
- Testes unitários, especialmente em funções críticas para o sistema.
- Monitoramento de dependencias para garantir que não existam vulnerabilidades em bibliotecas de terceiros.
- Eslint e Prettier para melhorar o padrão de código.
- Tecnologias a serem usadas:
  - ReactJS com Vite e TypeScript
    - TailwindCSS
    - HeroUI
  - NodeJS com Express e TypeScript