



**CENTRO UNIVERSITÁRIO DE BELO HORIZONTE**

**Curso de Ciência da Computação**

Alexandre Guimarães Braga

Cauã Lucas Ribeiro Fraga

Fellipe Andrade Costa

Samuel Zappalá Batista

Vênica Angelos de Melo

**PROJETO "CIDADE SEMPRE ACESA": identificação de problemas e  
manutenção de iluminação em vias públicas**

Belo Horizonte

2024

## SUMÁRIO

<b>1 DEFINIÇÃO DO PROBLEMA.....</b>	<b>2</b>
<b>2 REQUISITOS DE USUÁRIOS.....</b>	<b>3</b>
2.1 CIDADÃO.....	3
2.2 ADMINISTRAÇÃO PÚBLICA (FUNCIONÁRIO PÚBLICO).....	3
2.3 EMPREITEIRO.....	4
<b>3 INTERAÇÃO COM IA GENERATIVA.....</b>	<b>5</b>
3.1 CIDADÃO.....	5
3.2 ADMINISTRAÇÃO PÚBLICA (FUNCIONÁRIO PÚBLICO).....	5
3.3 EMPREITEIRO.....	6
3.4 PRIORIZAÇÃO GERAL DOS REQUISITOS.....	6
<b>4 REQUISITOS FUNCIONAIS E NÃO FUNCIONAIS.....</b>	<b>7</b>
4.1 REQUISITOS FUNCIONAIS.....	8
4.1.1 Requisitos para o Cidadão.....	8
4.1.2 Requisitos para a Administração Pública.....	9
4.1.3 Requisitos para o Empreiteiro.....	10
4.2 REQUISITOS NÃO FUNCIONAIS.....	10
4.3 CLASSIFICAÇÃO E PRIORIZAÇÃO DOS REQUISITOS.....	11
<b>5 CASOS DE USO.....</b>	<b>13</b>
<b>6 DIAGRAMA DE CLASSES.....</b>	<b>14</b>
<b>7 DIAGRAMA DE ATIVIDADE.....</b>	<b>15</b>
<b>8 MODELO ENTIDADE-RELACIONAMENTO.....</b>	<b>16</b>
8.1 MODELO CONCEITUAL.....	16
8.2 MODELO LÓGICO.....	16
<b>9 MODELO FÍSICO DE DADOS COM SCRIPTS.....</b>	<b>17</b>
<b>10 ESTRATÉGIA DE MODELAGEM COM IA.....</b>	<b>20</b>
<b>11 CRÍTICA À IA.....</b>	<b>22</b>
<b>12 EVIDÊNCIAS DE INTERAÇÃO COM IA GENERATIVA.....</b>	<b>23</b>
<b>13 EVIDÊNCIAS DE TESTES REALIZADOS.....</b>	<b>35</b>
<b>14 REFERÊNCIAS BIBLIOGRÁFICAS.....</b>	<b>36</b>

## 1 DEFINIÇÃO DO PROBLEMA

O problema a ser solucionado pela aplicação é da ordem de *Atendimento ao Cidadão*: a prefeitura tem dificuldade em identificar rapidamente e reparar problemas de iluminação pública, tais como postes com lâmpadas queimadas e em mau funcionamento, decorrentes da falta de manutenção, acidentes ou outros fatores, trazendo diversos transtornos à comunidade, como aumento da insegurança e acidentes aos transeuntes. Com objetivo de solucionar esse problema, o cidadão terá meios de notificar a prefeitura para a troca dessas lâmpadas: uma forma intuitiva e eficiente para que o cidadão possa informar sobre ausência de manutenção nas vias públicas, propiciando respostas mais rápidas e satisfatórias à comunidade.

## 2 REQUISITOS DE USUÁRIOS

Para o levantamento de requisitos de usuários, foram considerados os seguintes perfis (atores): *cidadão*, *administração pública (funcionário público)* e *empreiteiro*.

### 2.1 CIDADÃO

1. Criar uma conta com seus dados pessoais;
2. Marcar sua localização atual ou selecionar no mapa;
3. Abrir um ticket para relatar o problema, anexar fotos;
4. Receber notificações para acompanhar a solicitação;
5. Avaliar o serviço e dar um feedback sobre a resolução do problema.

Por meio destes recursos, o cidadão poderia facilmente comunicar e acompanhar a manutenção da iluminação pública, garantindo um serviço mais eficiente e transparente.

### 2.2 ADMINISTRAÇÃO PÚBLICA (FUNCIONÁRIO PÚBLICO)

1. Acessar o sistema com login e senha;
2. Ter acesso ao *helpdesk*;
3. Filtrar e avaliar o nível de urgência dos tickets em aberto;
4. Agrupar os chamados de uma determinada região;
5. Gerar a ordem de serviço;
6. Contratar o empreiteiro;
7. Organizar as rotas para a realização dos serviços;
8. Fiscalizar a realização dos serviços;
9. Fornecer *feedback* ao cidadão;
10. Realizar o pagamento do empreiteiro.

Com essas funcionalidades, a administração pública poderia gerenciar de forma eficiente e transparente as solicitações de troca e manutenção, garantindo uma iluminação pública adequada e segura para os cidadãos.

## 2.3 EMPREITEIRO

1. Receber a ordem de serviço;
2. Receber a rota com as informações adicionais para a realização do serviço;
3. Enviar fotos e fornecer evidências da realização do serviço;
4. Gerar a NF (nota fiscal) do serviço e materiais utilizados;
5. Receber o pagamento.

O empreiteiro terceirizado, por sua vez, pode gerenciar de forma eficiente as solicitações de serviço, garantindo uma resposta rápida e eficaz às necessidades de manutenção da iluminação das vias públicas.

### 3 INTERAÇÃO COM IA GENERATIVA

Para realização da nossa entrevista identificamos os possíveis pontos de vista, onde os analistas se reuniram e utilizaram o *brainstorming* para identificar os serviços em potencial, usuários do sistema que interagem com o sistema. Em uma segunda etapa estruturamos os pontos de vista, relacionando-os e perguntamos a IA e foram gerados os seguintes requisitos de usuários a partir da interação:

#### 3.1 CIDADÃO

1. Relatar Problemas Facilmente (Alta Prioridade)
  - 1.1. Funcionalidade para relatar problemas de iluminação pública através de um aplicativo ou site;
  - 1.2. Possibilidade de anexar fotos e a localização exata no mapa.
2. Rastreamento de Solicitações (Alta Prioridade)
  - 2.1. Ferramenta para acompanhar o status da solicitação desde o recebimento até a resolução.
3. Notificações (Média Prioridade)
  - 3.1. Sistema de notificações para atualizar o cidadão sobre o progresso da solicitação via e-mail ou aplicativo.
4. Histórico de Solicitações (Média Prioridade)
  - 4.1. Acesso a um histórico de todas as solicitações feitas pelo cidadão e suas resoluções.
5. Feedback (Baixa Prioridade)
  - 5.1. Funcionalidade para permitir que o cidadão dê feedback sobre o serviço prestado após a resolução do problema.

#### 3.2 ADMINISTRAÇÃO PÚBLICA (FUNCIONÁRIO PÚBLICO)

1. Painel de Controle (Alta Prioridade)
  - 1.1. Um painel centralizado onde possam ser visualizadas todas as solicitações de manutenção em tempo real, com detalhes como localização, tipo de problema, e urgência.
2. Gestão de Tarefas (Alta Prioridade)

- 2.1. Ferramenta para designar tarefas aos empreiteiros, definir prazos, e acompanhar o progresso de cada tarefa.
- 3. Relatórios e Análises (Média Prioridade)
  - 3.1. Capacidade de gerar relatórios sobre frequência dos problemas, tempo médio de resolução, e outras métricas.
- 4. Comunicação Interna (Média Prioridade)
  - 4.1. Sistema de comunicação interna para interagir com cidadãos e empreiteiros, permitindo atualizações rápidas e troca de informações.
- 5. Integração com Outras Bases de Dados (Baixa Prioridade)
  - 5.1. Integração com outros sistemas da prefeitura para verificar orçamentos, aprovar despesas, e coordenar com outras áreas como segurança pública.

### 3.3 EMPREITEIRO

- 1. Recebimento de Ordens de Serviço (Alta Prioridade)
  - 1.1. Receber ordens de serviço diretamente pelo sistema com todos os detalhes necessários, como localização exata, tipo de problema, e prazo para reparo.
- 2. Ferramentas de Gestão (Alta Prioridade)
  - 2.1. Capacidade de organizar e priorizar tarefas, alocar recursos, e monitorar o progresso das atividades em campo.
- 3. Comunicação com a Administração (Média Prioridade)
  - 3.1. Canal de comunicação direto com os funcionários da prefeitura para resolver dúvidas rapidamente e reportar o andamento dos trabalhos.
- 4. Geolocalização (Média Prioridade)
  - 4.1. Utilização de geolocalização para encontrar rapidamente os locais dos problemas e otimizar as rotas de reparo.
- 5. Feedback e Avaliação (Baixa Prioridade)
  - 5.1. Ferramenta para receber feedback sobre o trabalho realizado para melhorar a qualidade do serviço.

### 3.4 PRIORIZAÇÃO GERAL DOS REQUISITOS

A interação com a IA generativa também resultou em uma compilação automática de todos os requisitos listados, ordenados de acordo com o nível de prioridade:

Com base na importância e impacto para cada grupo de usuário, os requisitos podem ser priorizados da seguinte forma:

1. Relatar Problemas Facilmente (Alta Prioridade)
2. Rastreamento de Solicitações (Alta Prioridade)
3. Painel de Controle (Alta Prioridade)
4. Gestão de Tarefas (Alta Prioridade)
5. Recebimento de Ordens de Serviço (Alta Prioridade)
6. Ferramentas de Gestão (Alta Prioridade)
7. Notificações (Média Prioridade)
8. Histórico de Solicitações (Média Prioridade)
9. Relatórios e Análises (Média Prioridade)
10. Comunicação Interna (Média Prioridade)
11. Comunicação com a Administração (Média Prioridade)
12. Geolocalização (Média Prioridade)
13. Integração com Outras Bases de Dados (Baixa Prioridade)
14. Feedback (Baixa Prioridade)
15. Feedback e Avaliação (Baixa Prioridade)

Esses requisitos, organizados e priorizados, fornecem uma base sólida para o desenvolvimento do sistema, atendendo às necessidades mais críticas de cada perfil de usuário e garantindo que os recursos mais importantes sejam implementados primeiro.



## 4 REQUISITOS FUNCIONAIS E NÃO FUNCIONAIS

A partir da interação com a IA generativa<sup>1</sup>, foram identificados e classificados os seguintes requisitos funcionais e não funcionais:

### 4.1 REQUISITOS FUNCIONAIS

#### 4.1.1 Requisitos para o Cidadão

RF1. Cadastro de Usuário:

RF1.1. O sistema deve permitir que o cidadão crie uma conta fornecendo dados pessoais (nome, endereço, email, telefone).

RF1.2. O sistema deve validar os dados fornecidos durante o cadastro.

RF2. Relato de Problemas:

RF2.1. O sistema deve permitir que o cidadão marque sua localização atual ou selecione um ponto no mapa para relatar um problema.

RF2.2. O sistema deve permitir que o cidadão abra um ticket para relatar problemas de iluminação pública, incluindo a descrição do problema.

RF2.3. O sistema deve permitir que o cidadão anexe fotos ao ticket.

RF3. Notificações e Acompanhamento:

RF3.1. O sistema deve notificar o cidadão sobre o status do seu ticket.

RF3.2. O sistema deve permitir que o cidadão acompanhe o progresso da sua solicitação através de uma interface.

RF4. Feedback:

RF4.1. O sistema deve permitir que o cidadão avalie o serviço após a resolução do problema.

RF4.2. O sistema deve permitir que o cidadão forneça feedback textual sobre o serviço prestado.

---

<sup>1</sup> <https://chatgpt.com/share/b1fe7a8f-980d-4d52-bde9-dd1b60c3375b>

#### **4.1.2 Requisitos para a Administração Pública**

##### **RF5. Acesso e Helpdesk:**

RF5.1. O sistema deve permitir que funcionários públicos façam login com senha.

RF5.2. O sistema deve fornecer uma interface de helpdesk para os funcionários públicos gerenciarem os tickets.

##### **RF6. Gerenciamento de Tickets:**

RF6.1. O sistema deve permitir que os funcionários filtrem e avaliem o nível de urgência dos tickets em aberto.

RF6.2. O sistema deve permitir o agrupamento de chamados por região.

##### **RF7. Ordem de Serviço e Contratação:**

RF7.1. O sistema deve gerar ordens de serviço baseadas nos tickets.

RF7.2. O sistema deve permitir a contratação de empreiteiros para a realização dos serviços.

##### **RF8. Organização de Rotas:**

RF8.1. O sistema deve otimizar e organizar as rotas para a realização dos serviços baseados na localização dos problemas.

##### **RF9. Fiscalização e Feedback:**

RF9.1. O sistema deve permitir que funcionários públicos fiscalizem a realização dos serviços.

RF9.2. O sistema deve permitir que funcionários forneçam feedback sobre a execução do serviço.

##### **RF10. Pagamento:**

RF10.1. O sistema deve permitir a realização do pagamento aos empreiteiros após a conclusão do serviço.

RF10.2. O sistema deverá receber e conferir se a Nota Fiscal emitida está em conformidade.

### 4.1.3 Requisitos para o Empreiteiro

#### RF11. Recebimento de Ordem de Serviço:

- RF11.1. O sistema deve notificar os empreiteiros sobre novas ordens de serviço.
- RF11.2. O sistema deve fornecer informações detalhadas sobre a rota e instruções adicionais para a execução do serviço.

#### RF12. Evidências de Serviço:

- RF12.1. O sistema deve permitir que os empreiteiros enviem fotos e outras evidências da realização do serviço.

#### RF13. Nota Fiscal e Pagamento:

- RF13.1. O sistema deve permitir que os empreiteiros gerem e submetam notas fiscais pelo serviço e materiais utilizados.
- RF13.2. O sistema deve permitir que as notas fiscais sejam enviadas em formato JPG ou PDF.
- RF13.3. O sistema deve processar o pagamento ao empreiteiro após a verificação das evidências e da nota fiscal.

## 4.2 REQUISITOS NÃO FUNCIONAIS

#### RNF1. Nota Fiscal e Pagamento:

- RNF1.1. O sistema deve utilizar autenticação segura (ex. OAuth, autenticação multifator) para acesso de cidadãos, funcionários públicos e empreiteiros.
- RNF1.2. Os dados dos usuários devem ser armazenados de forma criptografada.

#### RNF2. Desempenho:

- RNF2.1. O sistema deve ser capaz de processar e responder a solicitações de usuários em tempo real.
- RNF2.2. O sistema deve suportar um grande número de acessos simultâneos sem degradação de performance.

RNF3. Usabilidade:

RNF3.1. A interface do sistema deve ser intuitiva e fácil de usar para todos os tipos de usuários (cidadão, funcionário público e empreiteiro).

RNF3.2. O sistema deve ser acessível a partir de dispositivos móveis e desktops.

RNF4. Disponibilidade:

RNF4.1. O sistema deve ter uma disponibilidade mínima de 99.9%, garantindo que os serviços estejam acessíveis 24/7.

RNF5. Escalabilidade:

RNF5.1. O sistema deve ser escalável para suportar o crescimento do número de usuários e volume de dados.

RNF6. Manutenibilidade:

RNF6.1. O sistema deve ser fácil de manter e atualizar, permitindo a adição de novas funcionalidades sem interrupções significativas nos serviços.

RNF7. Conformidade:

RNF7.1. O sistema deve estar em conformidade com as regulamentações e leis locais sobre proteção de dados e privacidade (ex. LGPD no Brasil).

#### 4.3 CLASSIFICAÇÃO E PRIORIZAÇÃO DOS REQUISITOS

De forma similar ao que ocorreu com os requisitos de usuário, aqui a interação com a IA generativa também resultou em uma ordenação automática dos itens conforme o nível de prioridade:

Os requisitos devem ser classificados e priorizados com base na criticidade e impacto no funcionamento do sistema. Priorizar inicialmente os requisitos funcionais que garantem a criação e gerenciamento de tickets, pois eles são essenciais para a operação básica do sistema. Requisitos não funcionais de segurança e disponibilidade também devem ter alta prioridade para assegurar a integridade e confiabilidade do sistema.

#### Matriz de Priorização

##### Alta Prioridade:

- RF1.1, RF1.2, RF2.1, RF2.2, RF2.3, RF3.1, RF3.2, RF5.1, RF5.2, RF6.1, RF7.1, RF11.1
- RNF1.1, RNF1.2, RNF2.1, RNF4.1

##### Média Prioridade:

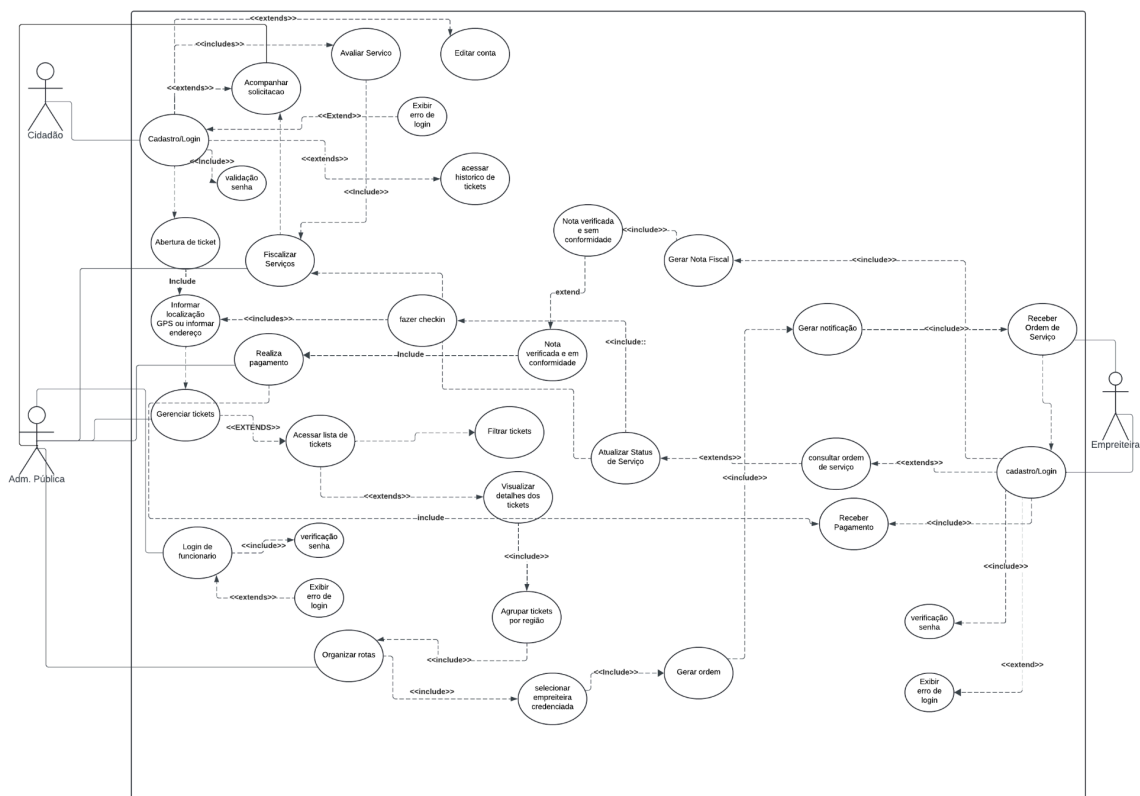
- RF4.1, RF4.2, RF6.2, RF8.1, RF9.1, RF9.2, RF10.1, RF12.1, RF13.1
- RNF3.1, RNF3.2, RNF5.1

##### Baixa Prioridade:

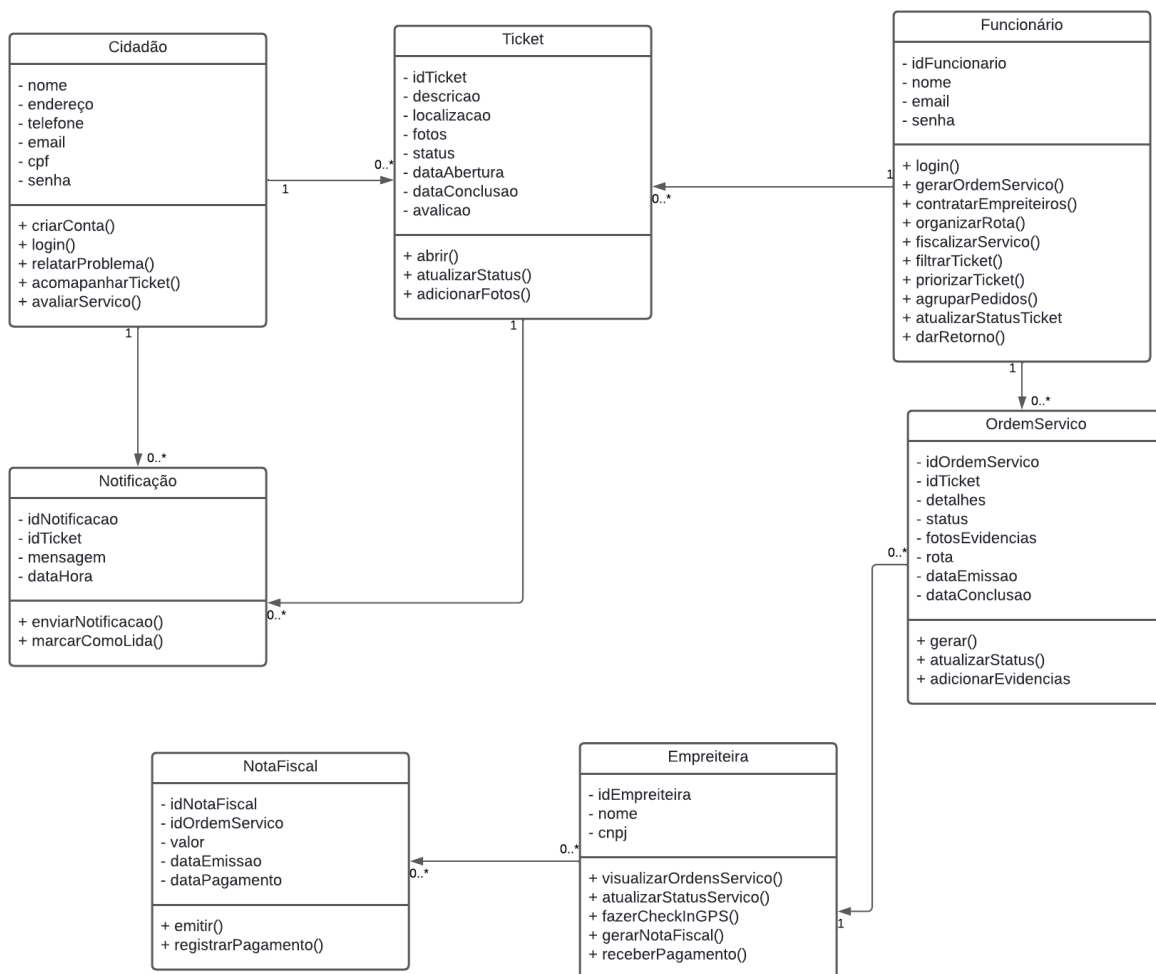
- RF7.2, RF11.2, RF13.2
- RNF6.1, RNF7.1

Esse detalhamento dos requisitos de sistema permite uma compreensão clara das necessidades dos usuários e as funcionalidades e restrições que o sistema deve atender para solucionar o problema de iluminação pública.

## 5 CASOS DE USO



## 6 DIAGRAMA DE CLASSES

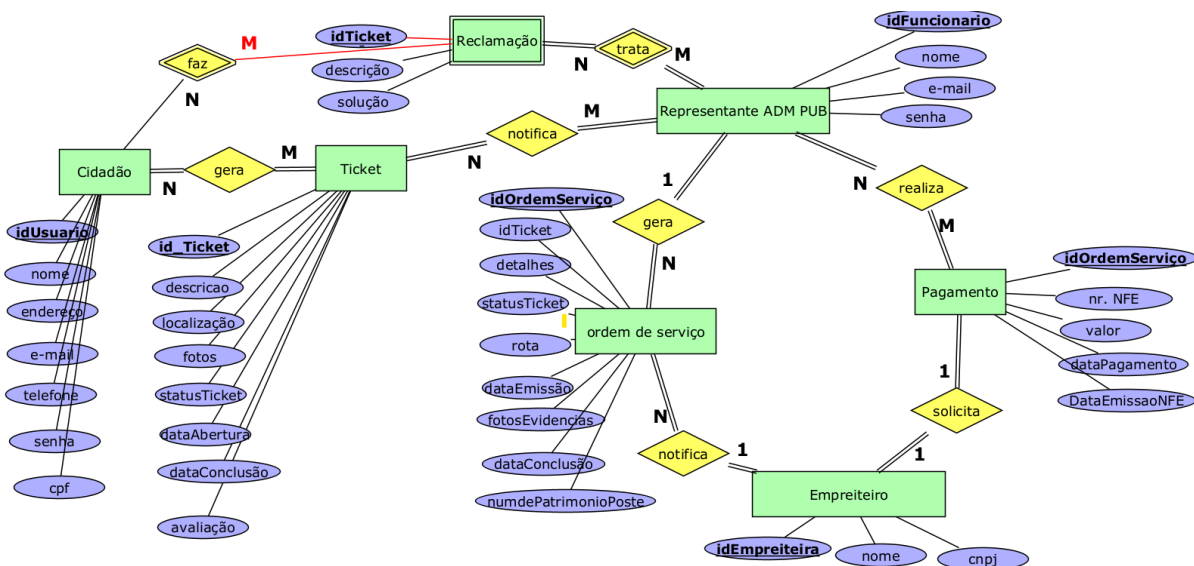




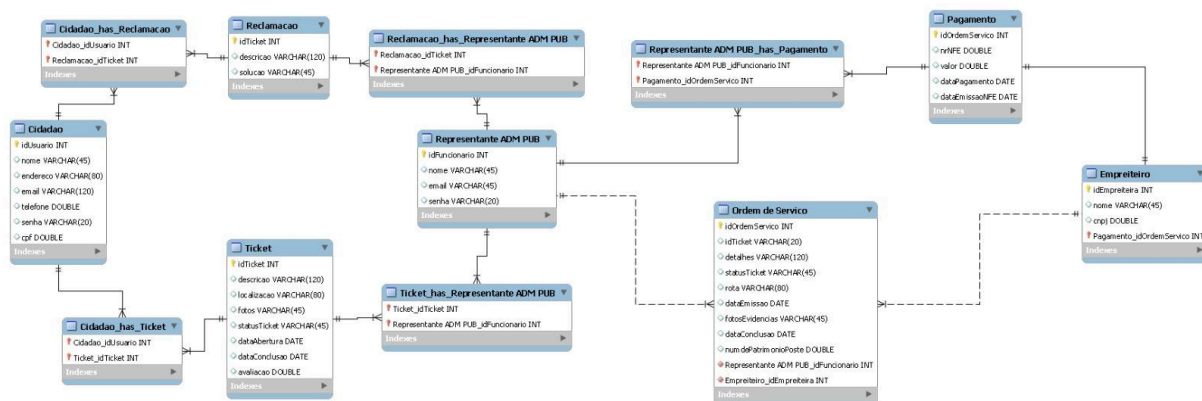


## 8 MODELO ENTIDADE-RELACIONAMENTO

### 8.1 MODELO CONCEITUAL



### 8.2 MODELO LÓGICO



## 9 MODELO FÍSICO DE DADOS COM SCRIPTS

```
CREATE TABLE cidadao (  
  idUsuario INT PRIMARY KEY,  
  nome VARCHAR(100),  
  endereco VARCHAR(200),  
  email VARCHAR(100),  
  telefone VARCHAR(20),  
  senha VARCHAR(50),  
  cpf VARCHAR(20)  
);
```

```
CREATE TABLE ticket (  
  idTicket INT PRIMARY KEY,  
  descricao TEXT,  
  localizacao VARCHAR(200),  
  fotos VARCHAR(255),  
  statusTicket VARCHAR(50),  
  dataAbertura DATE,  
  dataConclusao DATE,  
  avaliacao INT  
);
```

```
CREATE TABLE representanteAdmPub (  
  idFuncionario INT PRIMARY KEY,  
  nome VARCHAR(100),  
  email VARCHAR(100),  
  senha VARCHAR(50)  
);
```

```
CREATE TABLE reclamacao (  
  idTicket INT,  
  idUsuario INT,  
  descricao TEXT,
```

```

solucao TEXT,
PRIMARY KEY (idTicket, idUsuario),
FOREIGN KEY (idTicket) REFERENCES ticket(idTicket),
FOREIGN KEY (idUsuario) REFERENCES Cidadao(idUsuario)
);

```

```

CREATE TABLE ordemDeServico (
idOrdemServico INT PRIMARY KEY,
idTicket INT,
detalhes TEXT,
statusTicket VARCHAR(50),
rota VARCHAR(200),
dataEmissao DATE,
dataConclusao DATE,
numDePatrimonioPoste VARCHAR(50),
idFuncionario INT,
FOREIGN KEY (idTicket) REFERENCES ticket(idTicket),
FOREIGN KEY (idFuncionario) REFERENCES
RepresentanteAdmPub(idFuncionario)
);

```

```

CREATE TABLE empreiteiro (
idEmpreiteira INT PRIMARY KEY,
nome VARCHAR(100),
cnpj VARCHAR(20)
);

```

```

CREATE TABLE pagamento (
idOrdemServico INT,
idEmpreiteira INT,
nrNFE VARCHAR(50),
valor DECIMAL(10, 2),
dataPagamento DATE,
dataEmissaoNFE DATE,

```

```
PRIMARY KEY (idOrdemServico, idEmpreiteira),  
FOREIGN KEY (idOrdemServico) REFERENCES  
OrdemDeServico(idOrdemServico),  
FOREIGN KEY (idEmpreiteira) REFERENCES empreiteiro(idEmpreiteira)  
);
```

## 10 ESTRATÉGIA DE MODELAGEM COM IA

A utilização de ferramentas de IA generativa para identificação dos requisitos de usuário, requisitos de sistema e estratégias de modelagem, apesar de ter apresentado falhas, entregou resultados que podem ser considerados satisfatórios. A começar pelo levantamento dos requisitos de usuário, ainda que haja pequenas divergências, tanto do ponto de vista objetivo quanto estrutural—ao contrário da abordagem mais concisa adotada pelo grupo, a IA apresentou descrições um pouco mais amplas, geralmente englobando diversos requisitos em um único item—e de ter falhado em identificar alguns requisitos básicos anteriormente levantados pelo grupo—como a capacidade de poder se cadastrar e conectar-se ao sistema—a IA foi capaz de apontar necessidades relevantes dos usuários que ainda não haviam sido consideradas. Nota-se, também, que a IA apresentou, adequadamente, uma das características básicas dos requisitos de usuário: uma descrição de alto nível do sistema e suas funcionalidades em linguagem natural, direcionada ao público-alvo correto. Contudo, a resposta fornecida limita-se às funcionalidades que atendem às necessidades dos usuários, falhando ao não incluir descrições acerca das restrições sob as quais o sistema deve operar. Conforme Sommerville (2016, p. 102):

Os requisitos do usuário são declarações, em linguagem natural e diagramas, de quais serviços o sistema deve fornecer aos usuários do sistema e as restrições sob as quais ele deve operar. Os requisitos do usuário podem variar desde declarações amplas dos recursos necessários do sistema até descrições detalhadas e precisas da funcionalidade do sistema. (Tradução própria<sup>2</sup>)

Já no caso dos requisitos funcionais e não funcionais, a situação impôs algumas dificuldades, o que acabou acarretando em uma mudança na estratégia adotada na interação, a ser discutida mais adiante. Inicialmente, a IA forneceu uma resposta majoritariamente insatisfatória, listando os requisitos de sistema com descrições genéricas e simples, similares à forma como apresentado nos requisitos de usuário. O prompt inicial consistia em uma contextualização acerca do problema a ser resolvido pelo sistema, seus atores e a lista de requisitos de usuário

---

<sup>2</sup> User requirements are statements, in a natural language plus diagrams, of what services the system is expected to provide to system users and the constraints under which it must operate. The user requirements may vary from broad statements of the system features required to detailed, precise descriptions of the system functionality.

previamente estabelecida. A julgar pelos resultados, podemos observar que, após ser alimentada com essas informações, a IA praticamente apenas devolveu os mesmos requisitos com alterações insignificantes na escrita. Para tentar contornar o problema, uma segunda tentativa foi realizada: alteramos o prompt de modo a solicitar explicitamente uma resposta mais técnica e detalhada.

Prompt inicial (trecho):

Com base nas informações acima, ajude a definir os requisitos de sistema, dividindo o problema em requisitos funcionais e não funcionais, e auxilie na identificação e classificação dos requisitos, garantindo que todas as funcionalidades e restrições do sistema sejam consideradas.

Prompt editado (trecho):

Com base nas informações acima, ajude a definir os requisitos de sistema, dividindo o problema em requisitos funcionais e não funcionais, e auxilie na identificação e classificação dos requisitos, garantindo que todas as funcionalidades e restrições do sistema sejam consideradas. Os requisitos devem ser apresentados de forma técnica e detalhada, destrinchando os requisitos de usuário em quantos requisitos funcionais e não funcionais forem necessários.

A segunda tentativa de interação resultou em uma resposta mais satisfatória no que se refere à forma de apresentação dos requisitos de modo geral, mas com poucos ganhos na qualidade das descrições dos requisitos funcionais. A exceção fica por conta dos requisitos não funcionais, onde pode-se observar uma pequena melhora no detalhamento das descrições, que passam a ser ligeiramente mais específicas e adotam uma linguagem um pouco mais técnica.

## 11 CRÍTICA À IA

Pode-se dizer, portanto, que a divisão dos problemas em requisitos funcionais e não funcionais teve um resultado misto, com muito espaço para melhora mesmo após a segunda tentativa de interação. A IA falhou em não expandir os requisitos de usuário em requisitos mais específicos, deixando de fornecer descrições mais detalhadas das funções do sistema, seus serviços e restrições operacionais, conforme proposto por Sommerville (2016), praticamente limitando-se ao escopo abrangente definido anteriormente.

É importante destacar que uma contribuição interessante, ainda que não solicitada no prompt, tanto no caso dos requisitos de usuário quanto na segunda interação para a geração dos requisitos de sistema foi a determinação do nível de prioridade dos requisitos com base na sua importância e impacto, e a organização dos mesmos seguindo este critério. Tal contribuição, como bem apontado pela própria IA, auxilia diretamente no processo de desenvolvimento do sistema, garantindo que as necessidades mais críticas sejam implementadas primeiro.

## 12 EVIDÊNCIAS DE INTERAÇÃO COM IA GENERATIVA

Primeira interação para a geração dos requisitos de sistema:

### Requisitos Funcionais e Não Funcionais

June 13, 2024

Considere o seguinte contexto referente à modelagem de uma aplicação:

"O problema a ser solucionado pela aplicação é da ordem de Atendimento ao Cidadão: a prefeitura tem dificuldade em identificar rapidamente e reparar problemas de iluminação pública, tais como postes com lâmpadas queimadas e em mau funcionamento, decorrentes da falta de manutenção, acidentes ou outros fatores, trazendo diversos transtornos à comunidade, como aumento da insegurança e acidentes aos transeuntes."

Agora, considere os seguintes requisitos de usuário adquiridos por meio de um levantamento feito com diferentes tipos de usuário do sistema (cidadão, funcionário público e empreiteiro):

#### 2.1 CIDADÃO

- Criar uma conta com seus dados pessoais;
- Marcar sua localização atual ou selecionar no mapa;
- Abrir um ticket para relatar o problema, anexar fotos;
- Receber notificações para acompanhar a solicitação;
- Avaliar o serviço e dar um feedback sobre a resolução do problema.

#### 2.2 ADMINISTRAÇÃO PÚBLICA (FUNCIONÁRIO PÚBLICO)

- Acessar o sistema com login e senha;
- Ter acesso ao helpdesk;
- Filtrar e avaliar o nível de urgência dos tickets em aberto;
- Agrupar os chamados de uma determinada região;
- Gerar a ordem de serviço;
- Contratar o empreiteiro;



Organizar as rotas para a realização dos serviços;  
 Fiscalizar a realização dos serviços;  
 Fornecer feedback ao cidadão;  
 Realizar o pagamento do empreiteiro.

### 2.3 EMPREITEIRO

Receber a ordem de serviço;  
 Receber a rota com as informações adicionais para a realização do serviço;  
 Enviar fotos e fornecer evidências da realização do serviço;  
 Gerar a NF (nota fiscal) do serviço e materiais utilizados;  
 Receber o pagamento.

Com base nas informações acima, ajude a definir os requisitos de sistema, dividindo o problema em requisitos funcionais e não funcionais, e auxiliando na identificação e classificação dos requisitos, garantindo que todas as funcionalidades e restrições do sistema sejam consideradas.

Para definir os requisitos de sistema para a aplicação de Atendimento ao Cidadão, precisamos identificar os requisitos funcionais e não funcionais com base nos diferentes tipos de usuários (cidadão, funcionário público e empreiteiro) e nas funcionalidades e restrições que a aplicação deve atender.

## Requisitos Funcionais

### Cidadão

#### 1. Cadastro e Autenticação:

- O cidadão deve poder criar uma conta com seus dados pessoais.
- O cidadão deve poder fazer login e logout no sistema.

#### 2. Reportar Problemas:

- O cidadão deve poder marcar sua localização atual ou selecionar um ponto no mapa.
- O cidadão deve poder abrir um ticket para relatar problemas de iluminação pública.

- O cidadão deve poder anexar fotos ao ticket.

### **3. Acompanhamento e Feedback:**

- O cidadão deve receber notificações sobre o status de sua solicitação.
- O cidadão deve poder avaliar o serviço e fornecer feedback após a resolução do problema.

## **Administração Pública (Funcionário Público)**

### **1. Acesso ao Sistema:**

- O funcionário público deve poder acessar o sistema com login e senha.

### **2. Gestão de Tickets:**

- O funcionário deve poder acessar o helpdesk.
- O funcionário deve poder filtrar e avaliar o nível de urgência dos tickets em aberto.
- O funcionário deve poder agrupar os chamados de uma determinada região.

### **3. Ordem de Serviço e Contratação:**

- O funcionário deve poder gerar ordens de serviço.
- O funcionário deve poder contratar empreiteiros.

### **4. Organização e Fiscalização:**

- O funcionário deve poder organizar as rotas para a realização dos serviços.
- O funcionário deve poder fiscalizar a realização dos serviços.

### **5. Comunicação e Pagamento:**

- O funcionário deve poder fornecer feedback ao cidadão.
- O funcionário deve poder realizar o pagamento ao empreiteiro.

## **Empreiteiro**

### **1. Recebimento de Ordem de Serviço:**

- O empreiteiro deve receber a ordem de serviço.
- O empreiteiro deve receber a rota com informações adicionais para a realização do serviço.

**2. Execução e Evidências:**

- O empreiteiro deve poder enviar fotos e fornecer evidências da realização do serviço.

**3. Faturamento:**

- O empreiteiro deve poder gerar a nota fiscal do serviço e dos materiais utilizados.
- O empreiteiro deve poder receber o pagamento.

**Requisitos Não Funcionais****1. Segurança:**

- O sistema deve garantir a segurança dos dados pessoais dos cidadãos e funcionários.
- O sistema deve assegurar que apenas usuários autenticados possam acessar as funcionalidades restritas.

**2. Performance:**

- O sistema deve ser capaz de processar e responder às solicitações dos usuários em tempo hábil, mesmo em picos de uso.
- O sistema deve suportar um grande número de usuários simultâneos sem degradação significativa de performance.

**3. Usabilidade:**

- A interface do usuário deve ser intuitiva e fácil de usar para todos os tipos de usuários (cidadãos, funcionários públicos e empreiteiros).
- O sistema deve fornecer feedback claro ao usuário sobre as ações realizadas e o status das solicitações.

**4. Escalabilidade:**

- O sistema deve ser escalável para atender a um número crescente de usuários e solicitações sem perda de performance.

**5. Confiabilidade:**

- O sistema deve ser altamente disponível e ter um plano de recuperação em caso de falhas.
- O sistema deve garantir a integridade dos dados em todas as transações.

#### 6. **Compatibilidade:**

- O sistema deve ser compatível com diferentes dispositivos e navegadores, permitindo o acesso tanto via desktop quanto dispositivos móveis.

### **Classificação dos Requisitos**

- **RF01:** O sistema deve permitir ao cidadão criar uma conta com seus dados pessoais.
- **RF02:** O sistema deve permitir ao cidadão fazer login e logout.
- **RF03:** O sistema deve permitir ao cidadão marcar sua localização atual ou selecionar um ponto no mapa.
- **RF04:** O sistema deve permitir ao cidadão abrir um ticket para relatar problemas.
- **RF05:** O sistema deve permitir ao cidadão anexar fotos ao ticket.
- **RF06:** O sistema deve enviar notificações ao cidadão sobre o status da solicitação.
- **RF07:** O sistema deve permitir ao cidadão avaliar o serviço e fornecer feedback.
- **RF08:** O sistema deve permitir ao funcionário público acessar o sistema com login e senha.
- **RF09:** O sistema deve fornecer acesso ao helpdesk para o funcionário público.
- **RF10:** O sistema deve permitir ao funcionário público filtrar e avaliar tickets.
- **RF11:** O sistema deve permitir ao funcionário público agrupar chamados por região.
- **RF12:** O sistema deve permitir ao funcionário público gerar ordens de serviço.
- **RF13:** O sistema deve permitir ao funcionário público contratar empreiteiros.
- **RF14:** O sistema deve permitir ao funcionário público organizar rotas.

- **RF12:** O sistema deve permitir ao funcionário público gerar ordens de serviço.
- **RF13:** O sistema deve permitir ao funcionário público contratar empreiteiros.
- **RF14:** O sistema deve permitir ao funcionário público organizar rotas.
- **RF15:** O sistema deve permitir ao funcionário público fiscalizar serviços.
- **RF16:** O sistema deve permitir ao funcionário público fornecer feedback ao cidadão.
- **RF17:** O sistema deve permitir ao funcionário público realizar pagamento ao empreiteiro.
- **RF18:** O sistema deve permitir ao empreiteiro receber ordens de serviço e rotas.
- **RF19:** O sistema deve permitir ao empreiteiro enviar fotos e evidências do serviço.
- **RF20:** O sistema deve permitir ao empreiteiro gerar notas fiscais.
- **RF21:** O sistema deve permitir ao empreiteiro receber pagamento.
- **RNF01:** O sistema deve garantir a segurança dos dados pessoais.
- **RNF02:** O sistema deve ser responsivo e performático.
- **RNF03:** O sistema deve ter uma interface intuitiva.
- **RNF04:** O sistema deve ser escalável.
- **RNF05:** O sistema deve ser confiável e altamente disponível.
- **RNF06:** O sistema deve ser compatível com diversos dispositivos e navegadores.

## **Conclusão**

Com esses requisitos definidos, a equipe de desenvolvimento poderá elaborar um plano de implementação que abranja todas as funcionalidades necessárias para os diferentes tipos de usuários, assegurando que o sistema atenda às expectativas de desempenho, segurança e usabilidade.

Segunda interação para a geração dos requisitos de sistema:

# Requisitos Funcionais e Não Funcionais

June 13, 2024

Considere o seguinte contexto referente à modelagem de uma aplicação:

"O problema a ser solucionado pela aplicação é da ordem de Atendimento ao Cidadão: a prefeitura tem dificuldade em identificar rapidamente e reparar problemas de iluminação pública, tais como postes com lâmpadas queimadas e em mau funcionamento, decorrentes da falta de manutenção, acidentes ou outros fatores, trazendo diversos transtornos à comunidade, como aumento da insegurança e acidentes aos transeuntes."

Agora, considere os seguintes requisitos de usuário adquiridos por meio de um levantamento feito com diferentes tipos de usuário do sistema (cidadão, funcionário público e empreiteiro):

## 2.1 CIDADÃO

- Criar uma conta com seus dados pessoais;
- Marcar sua localização atual ou selecionar no mapa;
- Abrir um ticket para relatar o problema, anexar fotos;
- Receber notificações para acompanhar a solicitação;
- Avaliar o serviço e dar um feedback sobre a resolução do problema.

## 2.2 ADMINISTRAÇÃO PÚBLICA (FUNCIONÁRIO PÚBLICO)

- Acessar o sistema com login e senha;
- Ter acesso ao helpdesk;
- Filtrar e avaliar o nível de urgência dos tickets em aberto;
- Agrupar os chamados de uma determinada região;
- Gerar a ordem de serviço;
- Contratar o empreiteiro;

Organizar as rotas para a realização dos serviços;  
Fiscalizar a realização dos serviços;  
Fornecer feedback ao cidadão;  
Realizar o pagamento do empreiteiro.

### 2.3 EMPREITEIRO

Receber a ordem de serviço;  
Receber a rota com as informações adicionais para a realização do serviço;  
Enviar fotos e fornecer evidências da realização do serviço;  
Gerar a NF (nota fiscal) do serviço e materiais utilizados;  
Receber o pagamento.

Com base nas informações acima, ajude a definir os requisitos de sistema, dividindo o problema em requisitos funcionais e não funcionais, e auxilie na identificação e classificação dos requisitos, garantindo que todas as funcionalidades e restrições do sistema sejam consideradas. Os requisitos devem ser apresentados de forma técnica e detalhada, destrinchando os requisitos de usuário em quantos requisitos funcionais e não funcionais forem necessários.

Para atender aos requisitos de usuário e resolver o problema da identificação e reparo rápido de problemas de iluminação pública, os requisitos de sistema podem ser divididos em requisitos funcionais e não funcionais conforme detalhado abaixo:

## Requisitos Funcionais

### Requisitos para o Cidadão:

#### 1. Cadastro de Usuário:

- RF1.1: O sistema deve permitir que o cidadão crie uma conta fornecendo dados pessoais (nome, endereço, email, telefone).
- RF1.2: O sistema deve validar os dados fornecidos durante o cadastro.

**2. Relato de Problemas:**

- RF2.1: O sistema deve permitir que o cidadão marque sua localização atual ou selecione um ponto no mapa para relatar um problema.
- RF2.2: O sistema deve permitir que o cidadão abra um ticket para relatar problemas de iluminação pública, incluindo a descrição do problema.
- RF2.3: O sistema deve permitir que o cidadão anexe fotos ao ticket.

**3. Notificações e Acompanhamento:**

- RF3.1: O sistema deve notificar o cidadão sobre o status do seu ticket.
- RF3.2: O sistema deve permitir que o cidadão acompanhe o progresso da sua solicitação através de uma interface.

**4. Feedback:**

- RF4.1: O sistema deve permitir que o cidadão avalie o serviço após a resolução do problema.
- RF4.2: O sistema deve permitir que o cidadão forneça feedback textual sobre o serviço prestado.

**Requisitos para a Administração Pública:****1. Acesso e Helpdesk:**

- RF5.1: O sistema deve permitir que funcionários públicos façam login com senha.
- RF5.2: O sistema deve fornecer uma interface de helpdesk para os funcionários públicos gerenciarem os tickets.

**2. Gerenciamento de Tickets:**

- RF6.1: O sistema deve permitir que os funcionários filtrem e avaliem o nível de urgência dos tickets em aberto.
- RF6.2: O sistema deve permitir o agrupamento de chamados por região.



**3. Ordem de Serviço e Contratação:**

- RF7.1: O sistema deve gerar ordens de serviço baseadas nos tickets.
- RF7.2: O sistema deve permitir a contratação de empreiteiros para a realização dos serviços.

**4. Organização de Rotas:**

- RF8.1: O sistema deve otimizar e organizar as rotas para a realização dos serviços baseados na localização dos problemas.

**5. Fiscalização e Feedback:**

- RF9.1: O sistema deve permitir que funcionários públicos fiscalizem a realização dos serviços.
- RF9.2: O sistema deve permitir que funcionários forneçam feedback sobre a execução do serviço.

**6. Pagamento:**

- RF10.1: O sistema deve permitir a realização do pagamento aos empreiteiros após a conclusão do serviço.

**Requisitos para o Empreiteiro:****1. Recebimento de Ordem de Serviço:**

- RF11.1: O sistema deve notificar os empreiteiros sobre novas ordens de serviço.
- RF11.2: O sistema deve fornecer informações detalhadas sobre a rota e instruções adicionais para a execução do serviço.

**2. Evidências de Serviço:**

- RF12.1: O sistema deve permitir que os empreiteiros enviem fotos e outras evidências da realização do serviço.

### 3. Nota Fiscal e Pagamento:

- RF13.1: O sistema deve permitir que os empreiteiros gerem e submetam notas fiscais pelo serviço e materiais utilizados.
- RF13.2: O sistema deve processar o pagamento ao empreiteiro após a verificação das evidências e da nota fiscal.

## Requisitos Não Funcionais

### 1. Segurança:

- RNF1.1: O sistema deve utilizar autenticação segura (ex. OAuth, autenticação multifator) para acesso de cidadãos, funcionários públicos e empreiteiros.
- RNF1.2: Os dados dos usuários devem ser armazenados de forma criptografada.

### 2. Desempenho:

- RNF2.1: O sistema deve ser capaz de processar e responder a solicitações de usuários em tempo real.
- RNF2.2: O sistema deve suportar um grande número de acessos simultâneos sem degradação de performance.

### 3. Usabilidade:

- RNF3.1: A interface do sistema deve ser intuitiva e fácil de usar para todos os tipos de usuários (cidadão, funcionário público e empreiteiro).
- RNF3.2: O sistema deve ser acessível a partir de dispositivos móveis e desktops.

### 4. Disponibilidade:

- RNF4.1: O sistema deve ter uma disponibilidade mínima de 99.9%, garantindo que os serviços estejam acessíveis 24/7.

### 5. Escalabilidade:

- RNF5.1: O sistema deve ser escalável para suportar o crescimento do número de usuários e volume de dados.

#### 6. **Manutenibilidade:**

- RNF6.1: O sistema deve ser fácil de manter e atualizar, permitindo a adição de novas funcionalidades sem interrupções significativas nos serviços.

#### 7. **Conformidade:**

- RNF7.1: O sistema deve estar em conformidade com as regulamentações e leis locais sobre proteção de dados e privacidade (ex. LGPD no Brasil).

### **Classificação e Priorização dos Requisitos**

Os requisitos devem ser classificados e priorizados com base na criticidade e impacto no funcionamento do sistema. Priorizar inicialmente os requisitos funcionais que garantem a criação e gerenciamento de tickets, pois eles são essenciais para a operação básica do sistema. Requisitos não funcionais de segurança e disponibilidade também devem ter alta prioridade para assegurar a integridade e confiabilidade do sistema.

### **Matriz de Priorização**

#### 1. **Alta Prioridade:**

- RF1.1, RF1.2, RF2.1, RF2.2, RF2.3, RF3.1, RF3.2, RF5.1, RF5.2, RF6.1, RF7.1, RF11.1
- RNF1.1, RNF1.2, RNF2.1, RNF4.1

#### 2. **Média Prioridade:**

- RF4.1, RF4.2, RF6.2, RF8.1, RF9.1, RF9.2, RF10.1, RF12.1, RF13.1
- RNF3.1, RNF3.2, RNF5.1

#### 3. **Baixa Prioridade:**

- RF7.2, RF11.2, RF13.2
- RNF6.1, RNF7.1

Esse detalhamento dos requisitos de sistema permite uma compreensão clara das necessidades dos usuários e as funcionalidades e restrições que o sistema deve atender para solucionar o problema de iluminação pública.

### **13 EVIDÊNCIAS DE TESTES REALIZADOS**

## 14 REFERÊNCIAS BIBLIOGRÁFICAS

1. CHATGPT. Resposta gerada pela IA. OpenAI, 13 jun. 2024. Disponível em: <https://chatgpt.com/share/b1fe7a8f-980d-4d52-bde9-dd1b60c3375b>. Acesso em: 13 jun. 2024.
2. SOMMERVILLE, Ian. Requirements engineering. *In*: SOMMERVILLE, Ian. **Software Engineering, Global Edition**. 10. ed. Inglaterra: Pearson, 2016. cap. 4, p. 102-137. ISBN 978-1292096131.