**Documento de Requisitos de Software**

**CIDADON**

Versão 1.1

**Desenvolvedores/Analistas**

GABRIEL FIGUEIREDO BEZERRA

MATEUS SILVA COSTA

SALOMÃO MACHADO MAFALDA

**Rio Branco – AC**

**2018**

**Histórico de Alterações**

|  |  |  |  |
| --- | --- | --- | --- |
| **Data** | **Versão** | **Descrição** | **Autor** |
| 03/11/2018 | 0.1.0 | Início da edição do documento | Gabriel Figueiredo Bezerra e Mateus da Silva Costa |
| 05/11/2018 | 0.1.1 | Edição dos requisitos | Gabriel Figueiredo Bezerra e Mateus da Silva Costa |
| 05/12/2018 | 1 | Revisão e adequação das avaliações do cliente/orientador | Mateus da Silva Costa |
| 20/12/2018 | 1.1 | Inclusão de requisitos | Salomão Machado Mafalda |

1. **Análise do Problema**

É notório que a administração pública tem dificuldade em notar e resolver os problemas estruturais das cidades como a estrutura viária, a rede de esgoto e água, a iluminação pública e a rede de distribuição de energia. A população sofre com esses problemas que causam prejuízos materiais e imateriais. A dificuldade na comunicação entre a população e as prefeituras, e muitas vezes o descaso da administração pública como os problemas da cidade, muitas vezes acaba passando despercebido e faz com que os problemas estruturais perdurem por muito tempo. Deste modo a solução proposta ao cliente visa disponibilizar uma plataforma que permita aos cidadãos listarem os problemas estruturais existentes na cidade, e que propicie ao cliente o levantamento desses dados.

1. **Necessidades Básicas do Cliente**

O cliente necessita de uma aplicação mobile que permita a usuários cadastrados inserir e manter descrições de problemas estruturais em um mapa da cidade onde mora. O problema deve pertencer a um de cinco tipos: estrutura viária, rede de esgoto, rede de água e iluminação pública. Usuários não cadastrados podem consultar o mapa com os problemas, mas não poderão inseri novos sem cadastro. A aplicação deve garantir que o usuário esteja próximo à região onde o problema vai ser inserido e uma mecanismo de avaliação dos problemas criados por outros, criticando-os ou favorecendo-os. Os problemas devem ser ocultados do mapa depois de um certo período de tempo, para cada tipo de problema

1. **Estudo de Viabilidade**

O desenvolvimento do software se mostra extremamente viável por ser amparada por ferramentas gratuitas. A implantação depende de investimento financeiro por parte do cliente.

**3.1. Viabilidade Técnica**

A implementação da aplicação necessita de conhecimento na framework ReactNative, feita na linguagem JavaScript e nas API’s Google Maps e Firebase do Google. O conhecimento sobre a utilização do *framework* e das API’s será abrangida durante o processo de desenvolvimento com cursos.

**3.2. Viabilidade Econômica**

Os gastos necessários envolvem a compra de cursos online sobre o framework e de armazenamento, após implementação, em um servidor privado, durante o processo de desenvolvimento será utilizado as versões gratuitas para testes.

**3.3. Viabilidade Legal**

Por meio do que se constata nada impede legalmente o desenvolvimento de tal solução proposta, uma vez que ela não fere nenhuma lei municipal, estatual ou federal nem tratados internacionais relacionados a utilização da internet. As ferramentas utilizam dação possuem licenças gratuitas de utilização dentro dos limites legais de autoria.

1. **Missão do Software**

O software tem por objetivo propiciar aos cidadãos uma plataforma que permita, após cadastro e autenticação, inserir e consultar problemas em um mapa além de permitir avaliação de problemas já inseridos. Usuários não cadastrados/autenticados poderão apenas consultar o mapa.

1. **Limites do Sistema**

|  |  |  |
| --- | --- | --- |
| **ID** | **Funcionalidade** | **Justificativa** |
| **L1** | Cadastro, edição e exclusão de contas de usuários | É necessário um cadastro e autenticação para inserir um problema infraestrutural no mapa da aplicação |
| **L2** | Cadastro e edição de problema | Para levantamento, atualização e exibição dos problemas. |
| **L3** | Consulta de problemas | Para permitir aos usuários visualizar os problemas |
| **L4** | Avaliar problemas | Para garantia a efetividade |

## Benefícios Gerais

|  |  |
| --- | --- |
| **ID** | **Benefício** |
| **B1** | O sistema proverá uma voz aos problemas dos cidadãos |
| **B2** | Prover a prefeitura do município facilidades na administração |
| **B3** | Levantamento de dados referentes a administração pública |

1. **Restrições**

|  |  |  |
| --- | --- | --- |
| **ID** | **Restrição** | **Descrição** |
| **R1** | Entrega | O software deve ser apresentado ao cliente, com todos os requisitos funcionais, até o dia 11 de fevereiro de 2018 |

1. **Atores**

|  |  |  |
| --- | --- | --- |
| **ID** | **Atores** | **Descrição** |
| **A1** | Usuário | Aquele que irá se manifestar referente a determinado problema em seu meio |
| **A2** | Administrador | Responsável pela manutenção do software |

1. **Requisitos Funcionais**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Funcionalidade** | **Necessidades** | **Classificação** |
| RF1 | Cadastro de conta de usuário | O cliente deve realizar um cadastro de seus dados para dar credibilidade as informações geradas |  |
| **RF2** | Edição de conta de usuário | Manter os dados dos cliente atualizados |  |
| **RF3** | Exclusão de conta de usuário | Permitir que um usuário exclua seu cadastro |  |
| **RF4** | Autenticar usuário | Para garantir que apenas usuários cadastros possam executar certas funcionalidades |  |
| **RF4** | Cadastro de Problema | E necessário que o usuário esteja logado para que seja possível se inserir um problema |  |
| **RF5** | Edição de Problema | Permitir a edição de problemas após sua criação |  |
| **RF6** | Exclusão de Problema | Para permitir ao usuário excluir problemas criados por ele e que por ventura não representem a realidade |  |
| **RF7** | Avaliação Problema | Definir métricas de prioridade |  |
| **RF8** | Denunciar Problema | Para garantir a veracidade dos dados apresentados no sistema |  |
| RF9 | Gerenciar ações de usuário | Permite ao usuário logado diversas funcionalidades, além de mostrar o mapa. |  |
| RF10 | Filtrar Problemas | Permite o usuário pesquisar problemas inserindo filtros. |  |
| RF11 | Visualizar problemas de usuário | Permite o usuário logado ver os problemas por ele cadastrado |  |

## Requisitos Não-Funcionais

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Requisito** | **Categoria** | **Classificação** |
| **NRF1** | Os problemas devem ser automaticamente ocultados depois de dois meses | manutenibilidade | Alta prioridade |
| **NRF2** | Cada tipo de problema deve ter um ícone diferente | usabilidade | Prioritário |
| **NRF3** | Ao entrar no sistema deve ser mostrado o mapa com todos problemas já registrados | usabilidade | prioritário |
| **NRF4** | Após o cadastro mostrar tela com feedback | usabilidade | prioritário |

## Requisitos de Hardware

**11.1. Configuração Mínima**

Dispositivo mobile com sistema operacional Android ou IOS.

**11.2. Configuração Recomendada**

* Sistema operacional atualizado

1. **Ferramentas de Desenvolvimento e Licença de Uso**
   1. ReactNative, licença gratuita;
   2. JavaScrip, licença gratuita;
   3. Google Maps API, gratuita de desenvolvimento;
   4. Atom, licença gratuita;
   5. SubliText, licença gratuita;
   6. Dispositivos mobiles;
   7. VSCode, licença gratuita;