

**PLATAFORMA *WEB* PARA GESTÃO
DE DADOS SOBRE PROBLEMAS URBANOS
COLETADOS VIA *API***

WILLIAN PEREIRA DE JESUS

POSSE- GO
2022

WILLIAN PEREIRA DE JESUS

**PLATAFORMA *WEB* PARA GESTÃO
DE DADOS SOBRE PROBLEMAS URBANOS
COLETADOS VIA *API***

Trabalho apresentado como requisito parcial para a Conclusão do Curso de Bacharelado em Sistemas de Informação da Universidade Estadual de Goiás.

Orientadora: Prof.^a Esp. Cristiane Batista Xavier
Coorientador: Prof. MSc. Ronaldo Ferreira da Silva

POSSE– GO
2023

COMISSÃO EXAMINADORA

Prof.^a Esp. Cristiane Batista Xavier
Universidade Estadual de Goiás

Prof. Dr. Roberto Felicio de Oliveira
Universidade Estadual de Goiás

Prof. Esp. Givanilde de Assis dos Santos Oliveira
Universidade Estadual de Goiás

Posse, 01 de Fevereiro de 2023.

Elaborada pelo Sistema de Geração Automática de Ficha Catalográfica da UEG
com os dados fornecidos pelo(a) autor(a).

PP436 Pereira de Jesus, Willian
p PLATAFORMA WEB PARA GESTÃO DE DADOS SOBRE PROBLEMAS
 URBANOS COLETADOS VIA API / Willian Pereira de Jesus;
 orientador Cristiane Batista Xavier; co-orientador
 Ronaldo Ferreira da Silva. -- Posse, 2023.
 46 p.

 Graduação - Sistemas de Informação -- Unidade de
 Posse, Universidade Estadual de Goiás, 2023.

 1. Tecnologia da Informação e comunicação . 2.
 Desenvolvimento. 3. Gestão de problemas urbano. 4. API.
 I. Batista Xavier, Cristiane, orient. II. Ferreira da
 Silva, Ronaldo, co-orient. III. Título.

DEDICATÓRIA

A Deus, por ter me dado forças, sabedoria, graça, ânimo.
A minha esposa que sempre me deu forças, sempre esteve ao meu lado.

AGRADECIMENTOS

A meu Deus por até aqui estar ao meu lado e ter chegado até aqui.

A minha querida Esposa Daniely, que me motivou a cada dia.

A minha família, pela confiança e motivação.

Aos amigos e colegas, pela força e pela vibração em relação a esta jornada.

Aos professores e colegas de Curso, pois juntos trilhamos uma etapa importante de nossas vidas.

A minha orientadora que me incentivou a ler mais, a ter mais atenção, para que essa monografia fosse a melhor.

EPÍGRAFE

“O temor do Senhor é o princípio da sabedoria,
e o conhecimento do Santo é prudência”.

Provérbios 9.10

RESUMO

Com a perspectiva de possibilitar comunicação, transparência, centralização, o projeto de software apresentado a seguir visa unir tecnologia da informação e aspectos de comunicação governamental e cidadão. Esse processo resultou em uma análise de possíveis melhorias a serem aplicadas por meio de soluções tecnológicas. Após uma pesquisa aplicando um questionário na prefeitura municipal de Simolândia-GO, o resultado apontou que havia a necessidade de um software para centralizar a resolução de problemas, a partir disso, o presente documento visa apresentar o desenvolvimento de um software, onde o mesmo é alimentado por uma *API* que vincula o aplicativo a um painel *Web* para gerenciamento de problemas, que por sua vez permite gerenciá-los. Em suma, esta modularização tem como resultado final um controle centralizado sobre as manutenções expostas pelos cidadãos. Esta plataforma visa conectar gestores municipais e cidadãos, mas com capacidade de atuação em qualquer município.

Palavras-chave: Comunicação, Gestão da Informação, *Web*, Centralização.

ABSTRACT

With the perspective of enabling communication, transparency, centralization, the software project presented below aims to unite information technology and aspects of government communication and citizens. This process resulted in an analysis of possible improvements to be applied through technological solutions. After a survey applying a questionnaire in the city hall of Simolândia-GO, the result pointed out that there was a need for a software to centralize the resolution of problems, from this, this present document aims to present the development of a software, where the same is powered by an API that links the application to a Web panel for managing issues, which in turn allows managing them. In short, this modularization has as its final result a centralized control over the maintenance exposed by citizens. This platform aims at connecting municipal managers and citizens, but with the ability to act in any municipality..

Key Words: *Communication, Information Management, Web, Centralization.*

LISTA DE TABELAS

Tabela 1. Requisitos Funcionais	24
Tabela 2. Requisitos Não Funcionais	25
Tabela 3. Regras de Negócio	26
Tabela 4. Documentação do Caso de Uso Ativar/Inativar	28
Tabela 5. Documentação do Caso de Uso Alterar Status.....	29
Tabela 6. Documentação do Caso de Uso Cadastro de Entidades	30

LISTA DE FIGURAS

Figura 1. Funcionamento da Plataforma <i>Web</i> com a <i>API</i>	19
Figura 2. Procedimentos Metodológicos	22
Figura 3. Diagrama de Caso de Uso	27
Figura 4. Diagrama de Classe	31
Figura 5. Diagrama de Sequência de Login	32
Figura 6. Diagrama de Sequência do Uso do Sistema	32
Figura 7. Diagrama de Mapeamento Objeto Relacional (ORM)	33
Figura 8. Tela de Login.....	34
Figura 9. Tela de <i>dashboard</i>	35
Figura 10. Tela de listagem de usuário ativos/inativos	35
Figura 11. Tela de listagem de usuário ativos	36
Figura 12. Tela de cadastro e edição de usuário.....	36
Figura 13. Continuação da tela anterior	37
Figura 14. Continuação da tela anterior	37
Figura 15. Tela de listagem de problemas	38
Figura 16. Tela de cadastro de entidades	38
Figura 17. Continuação da tela anterior	39
Figura 18. Tela de listagem das entidades.....	39
Figura 19. Tela de edição do problema	40

LISTA DE ANEXOS

Anexo 1. Questionário para analisar e definir requisitos	46
---	----

Sumário

1 INTRODUÇÃO	14
1.2 Justificativa	15
1.3 Objetivos	15
1.3.1 Geral.....	15
1.3.2 Específicos.....	15
2 REVISÃO TEORICA	16
2.1 O uso da tecnologia para gestão pública	16
2.2 tecnologias e padrões de software utilizados.....	18
2.2.1 Barramentos de Dados.....	18
2.2.2 Interface de Programação de Aplicação (API)	18
2.2.3 Web services	19
2.2.4 Framework, bibliotecas e linguagens de programação.....	20
3 METODOLOGIA.....	22
4 RESULTADOS	23
4.1 Elementos da Documentação do software.....	23
4.1.1 Requisitos do software.....	23
4.1.1.1 Requisitos funcionais	23
4.1.2 Requisitos não funcionais.....	24
4.1.3 Regra de negócio	25
4.1.4 Diagrama de Caso e Uso	26
4.1.5 Documentação dos Caso de Uso.....	28
4.1.6 Diagrama de Classe	30
4.1.7 Diagrama de Sequência	31
4.1.8 Diagrama de Mapeamento Objeto-Relacional (ORM)	33
4.2 Telas da Plataforma.....	34
6 CONCLUSÃO	41
7 REFERÊNCIAS	42
8 ANEXOS	46

1 INTRODUÇÃO

Como a evolução do homem está em constante avanço, vem surgindo a necessidade de locais mais adequados (DOBZHANSKY, 1956). Como por exemplo, cidades bem estruturadas, com o objetivo de corresponder a realidade contemporânea, sendo assim, centros urbanos com infraestrutura de qualidade. No entanto, isso na realidade não acontece. Com relação a este fato, o autor (DHEIN, 2009) afirma que as cidades começaram a sofrer grandes impactos, principalmente com problemas da gestão de problemas urbanos, esses impactos podem estar atrelados com a expansão das áreas urbanas.

Logo, surge a necessidade de buscar soluções para esses problemas. E uma possível solução seria o uso da tecnologia (MONTEIRO, VASQUES, 2016). Visto que através das tecnologias é possível desenvolver ferramentas a fim de proporcionar um viés prático de inovação, facilidades à informação e melhorias em diversos âmbitos, como exemplo disso, na gestão de problemas municipais (AGUNE, 2006). Uma ferramenta de planejamento para enfrentar os problemas de gestão, poderá auxiliar na melhor execução de atividades pendentes, controlar e ter conhecimento de todos os déficits urbanos e uma possível comunicação com os cidadãos (BALBE, 2010).

Com isso o objetivo deste trabalho é o desenvolvimento de uma plataforma *Web* para gestão de dados sobre problemas urbanos coletados via *API* (interface de programação de aplicação). Uma *API* diz respeito a um código que permite que dois programas de software se comuniquem entre si (MORAIS, BATISTA, 2018). Essa plataforma permitirá um controle conciso em sua gestão e no planejamento de como os problemas serão enfrentados. Além disso, essa plataforma possibilitará uma melhor organização de acordo com cada necessidade apresentada, tendo assim um melhor controle na sua gestão.

Para alcançar os objetivos deste trabalho foram realizadas pesquisa bibliográfica, visando consultar o plano diretor municipal para o levantamento de normas, e entender a forma de como são tratados os problemas municipais. Além disso, foi realizado uma pesquisa de campo para identificar as metodologias que a gestão pública utiliza para enfrentar os problemas urbanos objetivando definir a análise de requisitos. Logo após, foi feita toda a documentação da plataforma, codificação e a fase de testes, até a disponibilização da plataforma a gestão.

1.2 Justificativa

Os problemas urbanos são aqueles que ocorrem nas zonas urbanas devido ao crescimento desordenado das cidades (MARTINS, 2012). Por isso, a gestão das áreas urbanas tem sido apontada como uma das questões mais desafiantes e pertinentes (PEREIRA, 2017). Pois, as gestões públicas, enfrentam dificuldades em gerenciar e resolver os problemas urbanos. E o uso da tecnologia pode auxiliar na gestão desses problemas.

Logo, torna-se necessário uma plataforma para gestão desses dados, pois pode ocasionar diversos problemas, como por exemplo a utilização de planilhas manuais para controlar os problemas urbanos relatados pelos cidadãos, assim como atrasos e falta de conhecimento dos problemas.

Por isso, o objetivo desse trabalho é desenvolver uma plataforma *Web* para gerir tais problemas. A plataforma será responsável pelo gerenciamento de dados disponibilizados em um barramento de dados abertos sobre problemas urbanos, seguindo um layout preestabelecido. Desta forma, quaisquer sistemas de gestão pública, ou mesmo aplicativos de terceiros, poderão alimentar a plataforma.

Assim, essa plataforma *Web* irá auxiliar os gestores a: (i) executar melhor as atividades pendentes, (ii) controlar e ter conhecimento de todos os problemas urbanos, (iii) possibilitar respostas as reclamações dos cidadãos e (iv) proporcionar uma plataforma mais ampla e completa para transformar dados em informações.

1.3 Objetivos

1.3.1 Geral

Afim de auxiliar a gestão pública a gerir os problemas urbanos, o objetivo principal desse trabalho é desenvolver uma plataforma *Web* para gerenciamento de dados sobre problemas urbanos coletados via *API*. Cujas finalidades de melhorar a gestão dos dados, vislumbrando a agilidade no trabalho da gestão.

1.3.2 Específicos

A fim de estender o alcance do objetivo geral, foi definido alguns Objetivos Específicos, os quais são descritos abaixo:

- Realizar um levantamento sobre a legislação que regulamenta normas em relação a gestão de problemas e diretrizes dos municípios brasileiros;

- Consultar as metodologias utilizadas pelos gestores para coletar e gerenciar as reclamações sobre problemas urbanos feito pelos cidadãos, visando definir os requisitos funcionais, não funcionais e regras de negócio da plataforma *Web*;
- Consultar e propor um barramento de dados para comunicação da plataforma e aplicativo mobile;
- Desenvolver uma plataforma *Web*, para coletar dados via *API*.
- Realizar testes de usabilidade objetivando verificar a experiência do usuário;
- Realizar testes de performance aspirando verificar se os comandos da plataforma *Web* respondem rapidamente a aplicação proposto por um barramento de dados abertos sobre problemas urbanos, seguindo um layout preestabelecido;
- Disponibilizar a plataforma *Web* para a gestão pública.

2 REVISÃO TEORICA

2.1 O uso da tecnologia para gestão pública

Um dos grandes pilares da gestão pública atual é a inovação, utilizando as TIC (Tecnologias da Informação e Comunicação) (BARBOSA, 2009). As TICs podem ser definidas como o conjunto total de tecnologias que permitem a produção, o acesso e a propagação de informações, assim como tecnologias que permitem a comunicação entre pessoas (RODRIGUES et al., 2014).

As TICs voltado a gestão pública, tem o objetivo é desenvolver condições para que o País possa evoluir tecnologicamente. Existem vários exemplos de iniciativas na administração pública em diferentes estados e municípios (BRAGA, 2008). Alguns desses exemplos incluem inovações na gestão da informação, atendimento ao usuário/cidadão, simplificação e modernização dos processos, avaliação de desempenho e controle de resultados, entre outros (BARBOSA, 2009).

No Brasil, existem meios tecnológicos que agregam em melhorias tecnologias como: O Siafi (Sistema Integrado de Administração Financeira do Governo Federal) é uma TIC do governo federal que foi implementado a partir de 1986, visando organizar e consolidar os registros das atividades orçamentária, financeira e contábil da União. (Suzart, 2010). O portal gov.br que por sua vez é uma forma de simplificar e unificar

canais e serviços digitais, de acordo com o decreto n. 9.756/1998.

Fica instituído o portal único “gov.br”, no âmbito dos órgãos e das entidades da administração pública federal direta, autárquica e fundacional do Poder Executivo Federal, por meio do qual informações institucionais, notícias e serviços públicos prestados pelo Governo federal serão disponibilizados de maneira centralizada. (PLANALTO, 2019).

Com a evolução do mundo foram surgindo novos conceitos, como a TICs (Tecnologia da Informação e Comunicação) que se tornou um assunto de grande relevância nos dias atuais com por exemplo na pandemia recente da COVID 19. De acordo com a pesquisa da Abradisti (Associação Brasileira de Distribuição de Tecnologia e Informação), o setor cresceu 13% em 2020.

O Estudo Setorial, que nesse ano também analisou o segmento de comunicação, registrou um crescimento de 13% para os distribuidores de TIC em 2020. Ao considerar apenas o setor de TI, o crescimento foi maior, de 23%. (CONTEÚDO EDITORIAL, 2021).

Estudos esses mostram o quão são importantes esse crescimento, pois trazem maior conectividade quando falamos de comunicação remota, redução de custos por conta de que não havendo necessidade de trabalhar presencialmente, mas *home-office*, diminuindo gastos com funcionários, evitando estresse no deslocamento do mesmo. Aumento na produtividade devido a possibilidade de reuniões instantâneas.

A tecnologia vem crescendo e cada dia ela contribui, melhora, implementa maneiras de organizar melhor seja no setor privado ou no público. De acordo com o autor o governo federal implementou a partir do ano de 1986 um software que visando organizar, consolidar orçamento e a contabilidade da União.

O governo viu a necessidade de um organizador de informação como os orçamentos e finanças eram manuais, informações desatualizadas, visando esses déficits gerados pelo fator de ser manual veio o sistema para desburocratizar e facilitar o acompanhamento e atualização de cada passo que a união dava em finanças.

Com isso o governo visa centralizar os meios de comunicação fazendo com que os documentos pessoais de cada cidadão fiquem em um só lugar serviços para o cidadão e informações sobre a atuação de todas as áreas do governo. Onde o cidadão pode de um único lugar utilizar de ferramentas para ter um controle sobre seus dados.

2.2 tecnologias e padrões de software utilizados

2.2.1 Barramentos de Dados

Barramento de dados também conhecido como *data bus* pode ser entendida como uma característica que se refere à capacidade de diversos sistemas e organizações trabalharem em conjunto (interoperar) de modo a garantir que pessoas, organizações e sistemas computacionais interajam para trocar informações de maneira eficaz e eficiente (ALENCAR, MAGALHÃES, VASCONCELOS, 2022). Cada vez mais, esse tipo de sistema vem sendo utilizado por entidades públicas, como por exemplo, o governo federal.

A utilização de um barramento de troca de dados que dê suporte à uma melhor interoperabilidade vem trazendo sucesso a modelos de Governos Eletrônicos em países como Estônia e Finlândia. (ALMEIDA. COSTA, p. 04).

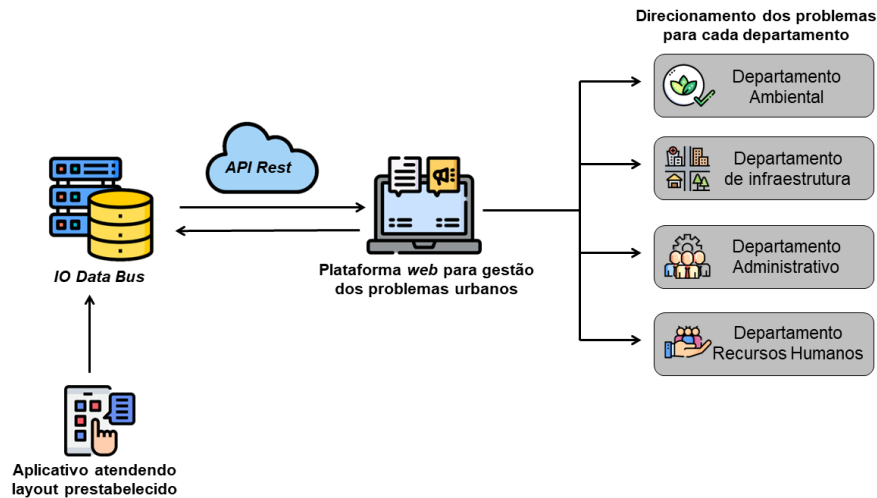
A interoperabilidade de barramento de dados é apresentada como uma proposta para unificação da oferta de serviços governamentais, no decreto nº 10.332, de 28 de abril de 2020 (GOV.BR, 2020). Por exemplo, hoje já é comum encontrar documentos de forma digital com o portal do gov.br, onde é possível fazer um único cadastro e ter todas as informações através de aplicativos no seu *smartphone*. Com isso, a proposta dessa aplicação é utilizar do conceito interoperável pra que a plataforma se comunique não somente com o local, mas também sirva para ampliar e melhorar as informações.

2.2.2 API (Interface de Programação de Aplicação)

Uma *API* (Interface de Programação de Aplicação) é um código que permite que dois programas de *software* se comuniquem entre si (COMPUTERWEECLE, 2021). A *API* determina a maneira correta para que um desenvolvedor escreva um programa que solicite os serviços de um SO (sistema operacional) ou de outro aplicação (COMPUTERWEECLE, 2021). A utilização desse recurso pode ajudar a economia do tempo e dinheiro, possibilitando o desenvolvimento de novas ferramentas e soluções. Além disso, com uma *API* é possível fazer o gerenciamento de sistemas existentes e simplificar a comunicação (RED HAT, 2017).

A seguir a **Figura 7**, apresenta uma representação de como nossa plataforma *Web* irá se comportar utilizando a *API* em conjunto com a interoperabilidade de barramento de dados.

Figura 1. Funcionamento da Plataforma Web com a API



Fonte: O autor

A plataforma Web funcionará seguindo alguns princípios: (i) como uma base de dados (*IO DATA BUS*) e (ii) uma *API*. Sendo assim a base de dados será alimentado e preenchido por um barramento de dados abertos sobre problemas urbanos, seguindo um layout preestabelecido. Após o mesmo ser alimentado, a *API* fornecerá os dados em um formato JSON (*JavaScript Object Notation*) que diz respeito a uma notação que permite estruturar dados em *formato* texto para serem utilizados em diferentes tipos de sistemas. Com isso, os dados serão tratados pela plataforma, onde será feito uma separação e indicação relativo a cada problema, onde o gestor então irá direcioná-los para cada departamento correspondente.

2.2.3 Web services

Web services é um meio utilizado para integração de sistemas e de comunicação entre aplicações heterogêneas (EULALIO, CORDEIRO, SOUZA, 2017). Para utiliza-lo são necessárias uma linguagem que intermediária para que possa haver uma comunicação entre a plataforma Web e outras aplicações, podendo ser ela através aplicativos móveis ou *desktop*. Sendo assim a plataforma proposta utilizará desses meios, para padronizar sua requisição e resposta ao banco de dados, utilizando o protocolo *REST* abaixo descrito.

O *REST* (*Representational State Transfer*), é um modelo arquitetural de software que tem objetivo de definir características para desenvolvimento tanto de aplicações *Web*, *mobile*, *desktop*, utilizando sé de princípios e definições necessárias para a criação de um projeto bem definido.

O *REST* é um “estilo arquitetônico” que basicamente explora a tecnologia existente e protocolos da *Web*. Enquanto *RESTful* é um serviço *Web* que utiliza esse paradigma, ou seja, é o termo normalmente usado para referir a implementação de *Web Services* que utilizam tal arquitetura. (OLIVEIRA, 2014; DURÃES, 2020; PIRES, 2018).

Para realizar a comunicação existem protocolos de comunicação como o *SOAP* (*Simple Object Access Protocol*) de acordo com o autor BERÇA (2021) é um protocolo simples, comercialmente independente que não define a semântica da aplicação onde será utilizado. Outrossim, ele expressa sua a semântica da aplicação através de uma estrutura simples.

Ambos protocolos são alimentados pelo *HTTP* (*Hyper Text Transfer Protocol*), um protocolo de camada de aplicação para transmissão de documentos hipermídia (MDN, 2022). Que foi desenvolvido especificamente para a comunicação entre navegadores *Web* e servidores *Web*, porém pode ser utilizado para outros propósitos também (MDN, 2022).

2.2.4 Framework, bibliotecas e linguagens de programação

Nesta seção será apresentada as tecnologias que irá compor o sistema *Web*, tais como: *frameworks*, bibliotecas. Para o desenvolvimento do sistema foi escolhido tecnologias que teria uma curva de aprendizagem simples, segura, ou seja, sempre buscando o equilíbrio para não se tornar algo inviável de desenvolver dentro do prazo já definido para a apresentação e funcionamento em ambiente de produção desse trabalho.

Ao escolher as tecnologias empregadas foram considerados tanto a manutenções futuras, bem como uma ampla documentação bem estruturada, atualizações que visam auxiliar na melhoria da tecnologia. Visando tais observações, definimos que a linguagem para o desenvolvimento *backend* é o *PHP*.

PHP significa *Hypertext Preprocessor* (pré-processador de hipertexto), essa linguagem de programação foi inicialmente criada inicialmente, mas não lançada, pelo o autor *Rasmus Lerdorf*, utilizava em uma página inicial para ficar de olho no seu currículo virtual.

O *PHP* foi concebido em meados de 1994 por *Rasmus Lerdorf*. Versões iniciais não lançadas foram usadas em sua home page para manter uma trilha de quem estava olhando o seu currículo virtual. (FACIVI, 2001/2002)

No ano seguinte ele liberou o código fonte do *PHP* que inicialmente se chamava *PHP Tools*. para que houvesse interação do público geral, assim abriu-se

leques de oportunidades para que surgissem novas ideias, otimizações pois seu código fonte estava livremente acessível.

Em 1995, Rasmus então liberou o código fonte do PHP Tools para o público geral. Dessa forma, abriu-se as portas para que ideias de outros desenvolvedores pudessem mexer e otimizar a estrutura do código-fonte livremente. Foi assim que a linguagem foi se aperfeiçoando. (ALURA, 2022).

Com o passar dos anos a linguagem de programação vem sofrendo alterações em seu nome até chegar no que hoje é conhecido como *PHP Hypertext Preprocessor*, ou simplesmente *PHP*. Atualmente essa linguagem de programação está em sua versão 8.1, e de acordo com o site da *W3TECHS*, o *PHP*, é a linguagem de programação que mais abrange o mercado *Web*, contando com mais de 77,5% dos sites desenvolvidos por ela.

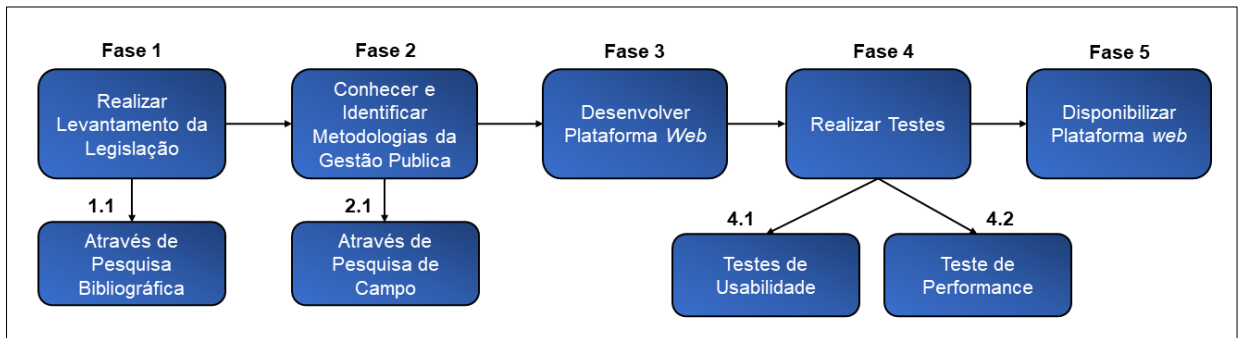
O presente sistema foi desenvolvido nesta linguagem por ser uma linguagem que tem uma longa jornada desde a sua criação até os dias atuais. Para ajudar no desenvolvimento do sistema foi adotado o *framework Laravel*, que é baseado nessa linguagem de programação. Os *frameworks* desta linguagem foram criados com intuito de padronizar o desenvolvimento, já que a linguagem não tinha ferramenta que sugeriam como desenvolver, como ter boas praticas

Então surgiram vários *frameworks* com intuito de suprir essas necessidades que hoje já existe, como *phpfig*, *phptherightway*, ferramentas essas desenvolvidas para boas práticas, código limpo, desenvolvidas por uma comunidade vasta que engloba os principais criadores dos *frameworks* e *CMS* mais utilizados no mercado atualmente, a comunidade e também do próprio fundador da linguagem.

3 METODOLOGIA

Para a realização deste trabalho foi definido 5 (cinco) fases que estão representadas na **Figura 1**. Cada fase é representada por retângulos e as setas indicam a ligação de uma fase a outra.

Figura 2. Procedimentos Metodológicos



Fonte: Autor

Na **Fase 1**, foi realizar levantamento da legislação, será realizado uma pesquisa bibliográfica consultando o plano diretor municipal para levantamentos de normas, para verificar como são tratados os problemas municipais, leis estaduais e federais, para serem levantados e identificados. Na **Fase 2**, foi feito Conhecimento e Identificação as metodologias da gestão pública, será realizado uma pesquisa de campo com observação, coleta, análise e interpretação de fatos que a gestão utiliza, para definir nossos requisitos funcionais, não funcionais e regras de negócio.

Na **Fase 3**, começamos a desenvolver Plataforma *Web* será com base nos requisitos apontados nas fases 1 e 2, consumindo dados de uma *API* disponibilizada. Na **Fase 4** foi realizado a fase de testes, que consiste em teste de usabilidade e teste de performance. O teste de usabilidade visa observar usuários reais usando o produto para descobrir problemas e pontos de melhorias. Já o teste de performance, será para verificar respostas concernentes a *API* disponibilizada por barramento de dados abertos sobre problemas urbanos. Na **Fase 5** foi disponibilizado a plataforma para o uso do sistema através da *Web*, não sendo necessária a instalação do mesmo.

4 RESULTADOS

4.1 Elementos da Documentação do software

Essa seção contém toda a documentação do sistema de como ele foi projetado desde a sua pesquisa para os requisitos, os diagramas que é feito para entender como um usuário irá interagir dentro do sistema, as tecnologias utilizadas em sua construção, bem como padrões utilizados.

4.1.1 Requisitos do software

4.1.1.1 Requisitos funcionais

Essa seção visa apresentar as representações dos Requisitos Funcionais do sistema proposto. RF (Requisitos funcionais) é uma declaração de como um sistema deve se comportar (VISURE, 2022). Define o que o sistema deve fazer para atender às necessidades ou expectativas do usuário (VISURE, 2022).

A seguir a **tabela 1** apresenta os RFs, que foram identificados através de aplicação de um formulário aplicado na prefeitura municipal de Simolândia, estado de Goiás, levando em consideração ferramentas, tecnologias e práticas de desenvolvimento de software ágil. Sendo que, a primeira coluna apresenta a identificação dos requisitos funcionais, a segunda apresenta sua descrição, a terceira apresenta o módulo em que ela se encaixa, e a quarta a sua prioridade de acordo com sua identificação.

Tabela 1. Requisitos Funcionais

Identificação	Descrição do requisito	Módulo	Prioridade
RF001 - Efetuar Login	O sistema deve permitir efetuar <i>login/logout</i> para acessar o sistema.	Cadastro	Alta
RF002 – Cadastro de Usuários	O administrador do sistema poderá cadastrar, editar, desativar ou consultar os usuários que tem ou terão acesso ao sistema.	Cadastro	Alta
RF003 - Cadastro Entidades	O administrador do sistema poderá cadastrar, editar, desativar ou consultar entidades no sistema.	Cadastro	Alta
RF004 - Gestão de Problemas	O administrador do sistema poderá gerenciar e direcionar os problemas para as entidades.	Gestão	Média
RF005 - <i>Feedback</i>	O administrador do sistema poderá alterar o <i>status</i> do problema, para aberto e/ou fechado.	Interno	Alto

Fonte: O Autor

4.1.2 Requisitos não funcionais

Essa seção visa apresentar as representações dos Requisitos não Funcionais do sistema proposto. RNF (Requisitos não funcionais) são as restrições impostas a um sistema que definem seus atributos de qualidade (VISURE, 2022). Eles geralmente são indicados por adjetivos como segurança, desempenho e escalabilidade (VISURE, 2022).

A seguir a **Tabela 2** apresenta os RNFs identificados através de aplicação de um formulário aplicado na prefeitura municipal local, levando em consideração ferramentas, tecnologias e práticas de desenvolvimento de software ágil. Sendo que, a primeira coluna apresenta a identificação dos requisitos não funcionais, a segunda apresenta sua descrição.

Tabela 2. Requisitos Não Funcionais

Identificação	Descrição do Requisito
RNF001	O sistema deve fornecer mecanismos de segurança e autenticação.
RNF002	O sistema deve estar disponível 24h por dia e 7 dias por semana.
RNF003	O sistema deve ser compatível com os navegadores: Google Chrome, Firefox e Safari.
RNF004	Somente o administrador pode gerenciar os usuários que estejam abaixo de seu nível do sistema.
RNF005	O sistema deve ter cores neutras para não ocasionar cansaço visual.
RNF006	O sistema deve possuir uma interface responsiva.
RNF007	O sistema deve possuir um super usuário, para recuperação de senha.

Fonte: O Autor

4.1.3 Regra de negócio

Regras de Negócio são padrões que condicionam o funcionamento do negócio, sendo comumente aplicadas no contexto da arquitetura de softwares, (MOVIDESK, 2020). Regras essas que são essenciais para que o sistema esteja alinhado com o planejamento da organização. Regras essas apresentadas que fazem parte da documentação do sistema e são estritamente fundamentais direcionar estrategicamente o sistema.

A **tabela 3** apresenta as Regras de Negócio identificados através dos requisitos funcionais e não funcionais descrito na seção anterior. Sendo que, a primeira coluna apresenta a identificação das regras de negócio, a segunda apresenta sua descrição.

Tabela 3. Regras de Negócio

Identificação	Descrição da Regra
RN001. Acesso ao sistema.	Para que o bom funcionamento e segurança do mesmo somente pessoas que tem um nível específico de acesso terá responsabilidades no sistema.
RN002. Nível de <i>superAdmin</i>	É o nível de acesso mais alto do sistema, ele é responsável pelo cadastro do <i>Admin</i> , recuperação de senha, ativar e/ou inativar qualquer usuário dentro do sistema.
RN003. Nível <i>Admin</i>	É o responsável por gerenciar qualquer usuário no sistema, podendo designar tarefas para cada usuário dentro do sistema, podendo também editar cadastro, cadastrar, inativar e/ou ativar usuários, atualizar <i>status</i> de cada problemas, redirecionar para os departamentos competentes.
RN004. Nível <i>collaborate</i>	É o nível de usuário para auxiliar o administrador, para realizar tarefas designada pelo <i>Admin</i> .
RN005. Cadastro de Entidades	Somente usuários com permissão de <i>Admin</i> poderá fazer o cadastro, edição, ativar e/ou inativar entidades.

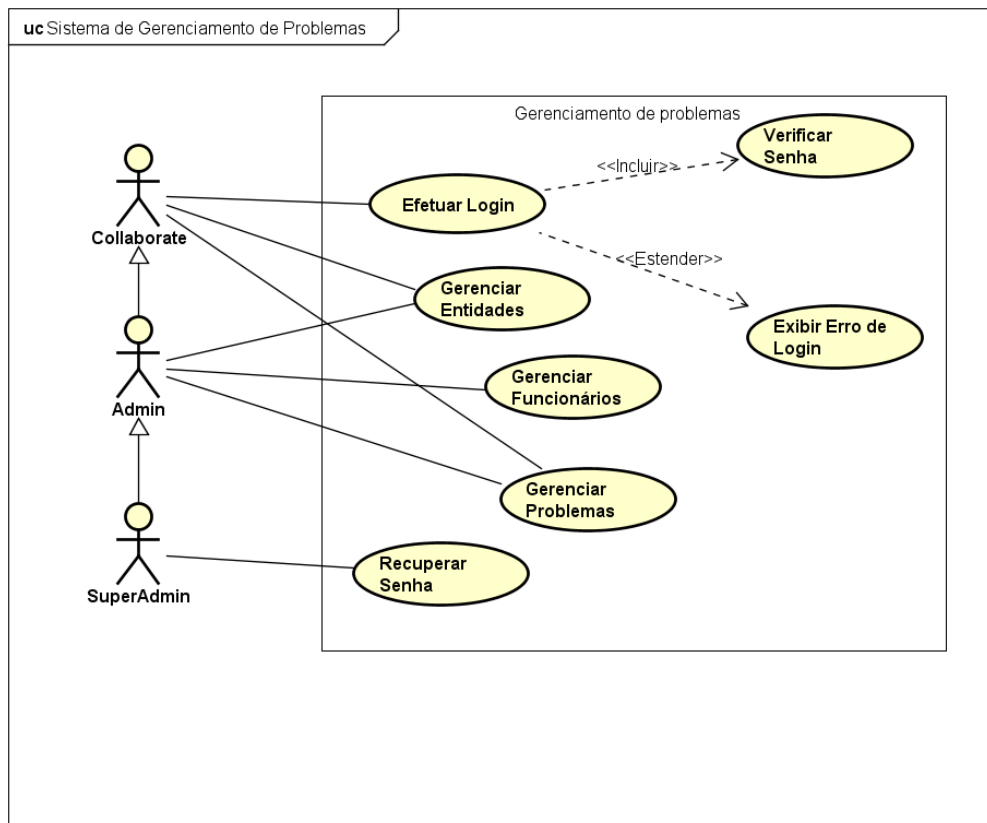
Fonte: O Autor

4.1.4 Diagrama de Caso e Uso

Nesta seção é apresentado o diagrama de caso de uso, documento este que compõe a documentação do sistema de software proposto a ser desenvolvido. De acordo com Guedes (2018) caso de uso possibilita a compreensão do comportamento do sistema por qualquer pessoa com conhecimento sobre o problema apresentado, tentando representart o sistmea na perspectiva do usuário.

Na figura **Figura 2** temos os atores representado os usuários, nas formas elípticas temos os casos de uso representando as funções do sistema e as linhas representado a ligação entre eles. Foram identificados 9 casos de uso para representar as principais funções propostas no levantamento de requisitos, realizada anteriormente.

Figura 3. Diagrama de Caso de Uso



powered by Astah

Fonte: Adaptado de GUEDES (2018).

A hierarquia do sistema foi dividida em 3, onde cada ator terá funções de acordo com o seu nível hierárquico e somente o *SuperAdmin* poderá gozar de todas as funções do sistema. O ator intitulado como *Admin* será responsável por quase todas as funções exceto recuperação de senha. O *Collaborate* tendo por sua vez limitações poderá listar todos os problemas e cadastrar entidade, demais funções somente com autorização do *Admin*.

4.1.5 Documentação dos Caso de Uso

Tabela 4. Documentação do Caso de Uso Ativar/Inativar

Nome do Caso de Uso	Ativar/Inativar
Atores principais	SuperAdmin, Admin
Atores Secundários	
Resumo	Esse caso de uso descreve as possíveis atividades na gestão de usuários, permitindo o SuperAdmin: ativar e/ou inativar usuários abaixo de seu cargo.
Pré-Condições	SuperAdmin, Admin devem estar logado.
Pós-Condições	Página de listagem de problemas.
Fluxo Principal	
Ações do Ator	Ações do Sistema
Selecionar a opção cadastrar usuário.	
	Apresentar a tela de cadastro de um novo usuário
3. Preencher os campos requeridos	
	4. Validar os campos
Fluxo de exceção – Dados inválidos	
Ações do Ator	Ações do sistema
	Indicar mensagem de erro.

Fonte: O Autor

Tabela 5. Documentação do Caso de Uso Alterar Status

Nome do Caso de Uso	Alterar Status
Atores principais	<i>Admin, Collaborate.</i>
Atores Secundários	
Resumo	Esse caso de uso descreve as possíveis atividades na gestão de status dos problemas, permitindo o <i>Admin</i> e <i>Collaborate</i> : alterar status dos problemas.
Pré-Condições	<i>Admin, Collaborate</i> devem estar logado.
Pós-Condições	Página de listagem do problema.
Fluxo Principal	
Ações do Ator	Ações do Sistema
Selecionar o problema.	
	Apresentar a tela de para alteração de status do problema.
3. Preencher os campos requeridos	
	4. Validar os campos
Fluxo de exceção – Dados inválidos	
Ações do Ator	Ações do sistema
	Indicar mensagem de erro.

Fonte: O Autor

Tabela 6. Documentação do Caso de Uso Cadastro de Entidades

Nome do Caso de Uso	Cadastro de Entidades
Atores principais	<i>Admin, Collaborate</i>
Atores Secundários	
Resumo	Esse caso de uso descreve as possíveis atividades na gestão de entidades, permitindo o <i>Admin</i> e <i>Collaborate</i> : Cadastrar entidades, alterar entidades, editar entidades.
Pré-Condições	<i>Admin. Collaborate</i> devem estar logado.
Pós-Condições	Página de listagem do problema.
Fluxo Principal	
Ações do Ator	Ações do Sistema
1. Selecionar o problema.	
	2. Apresentar a tela de para alteração de status do problema.
3. Preencher os campos requeridos	
	4. Validar os campos
Fluxo de exceção – Dados inválidos	
Ações do Ator	Ações do sistema
	1. Indicar mensagem de erro.

Fonte: O Autor

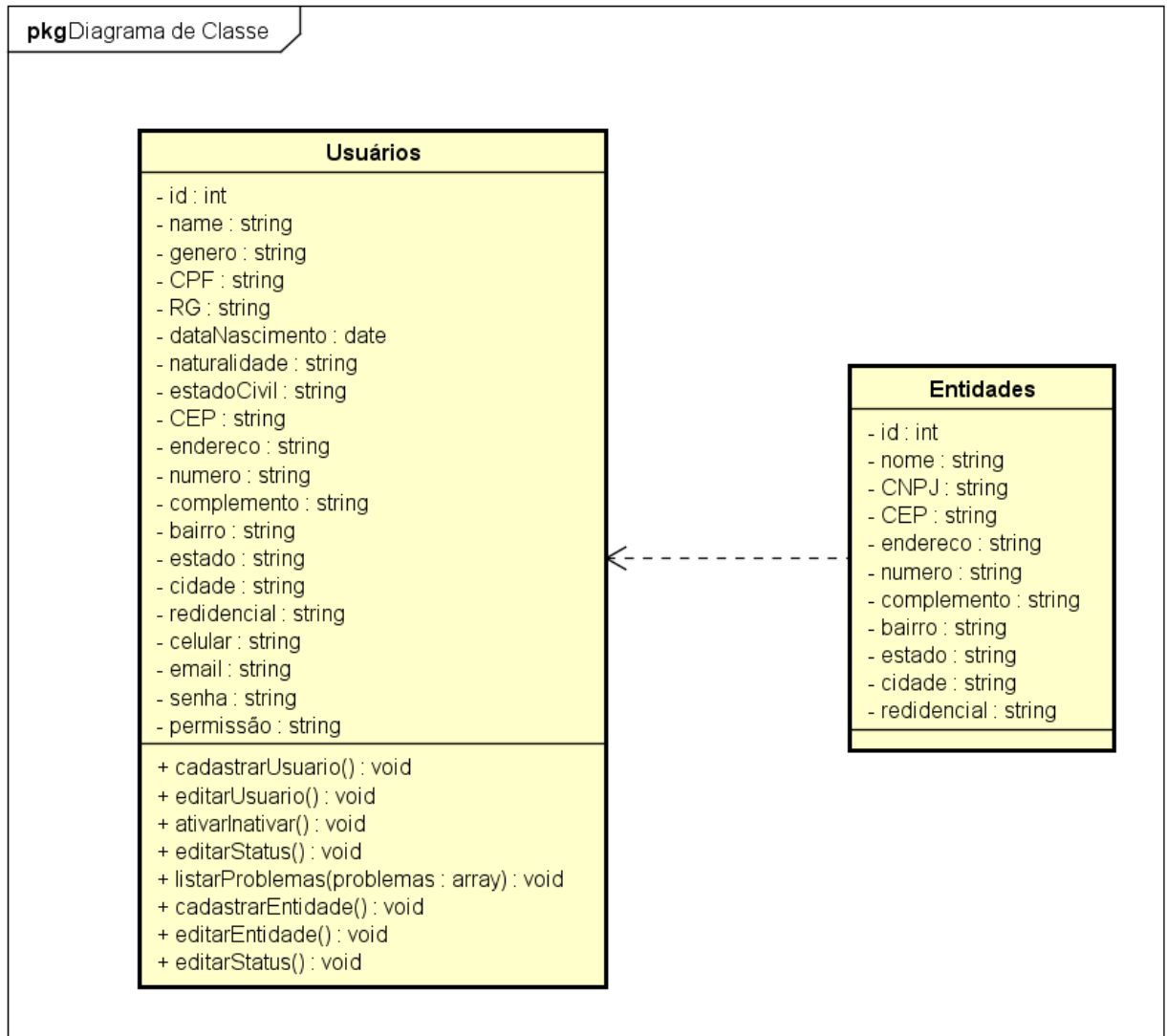
4.1.6 Diagrama de Classe

Nesta seção é apresentado o diagrama classe é uma representação estrutural e relacional das classes que servem de modelo para objetos. Os diagramas de classe são úteis em muitos estágios do design do sistema (IBM, 2022), pois ele é fundamental para descrever o que compoe o sistema que será desenvolvido, como ompreender os requisitos do domínio do problema e a identificar seus componentes.

A seguir na **Figura 3** temos duas classe definidas, cuja cada classe tem seus repectivos atributos, seus metodos para efetuar funções específicas. Nesse digrama

existe uma seta tracejada que indica dependência onde ela precisa de informações de outra classe.

Figura 4. Diagrama de Classe



powered by Astah

Fonte: Adaptado de GUEDES (2018).

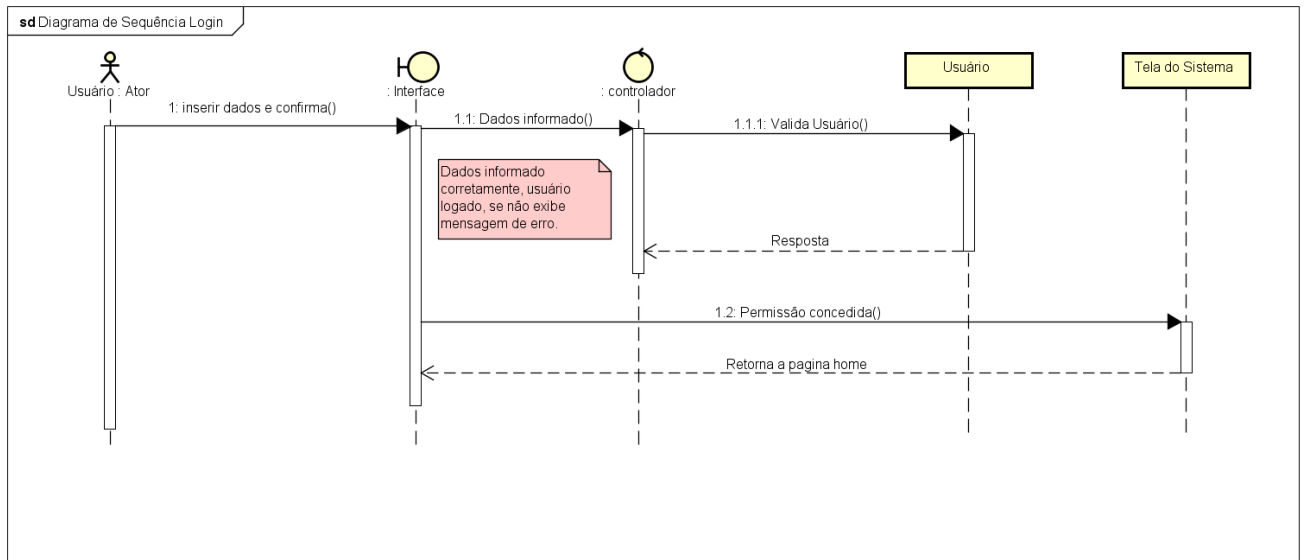
4.1.7 Diagrama de Sequência

Nesta seção é apresentado o diagrama sequência que descreve como, e em qual ordem, um grupo de objetos trabalha em conjunto, indicando cada ciclo de vida que cada interação, de acordo com o sistema desenvolvido (IBM, 2021). Estes diagramas são usados para entender as necessidades de um novo sistema ou para documentar um processo existente (IBM, 2021).

Na **Figura 4** é apresentado o diagrama de sequência do login do usuário, onde

contém o passo a passo para realizar suas interações com o sistema bem como mensagens de erros se usuário não tiver permissão ou não for cadastrado, e retorno de suas respectivas telas.

Figura 5. Diagrama de Sequência de Login

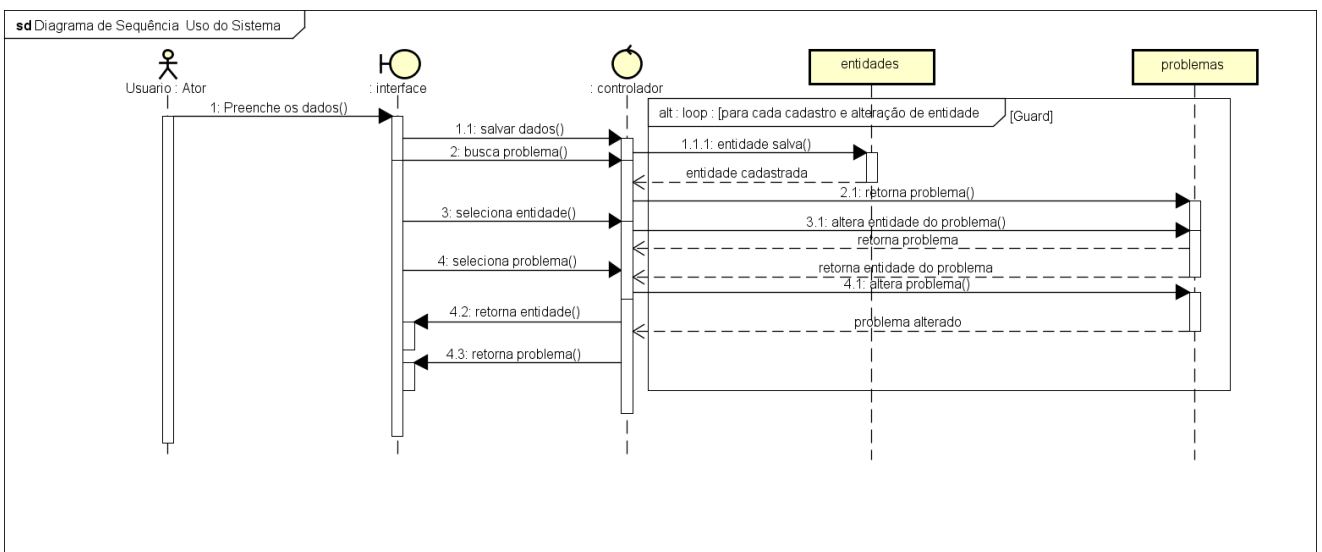


powered by Astah

Fonte: Adaptado de GUEDES (2018).

Na **Figura 5** temos o diagrama de sequência do uso do sistema, onde contém o passo a passo para utilizar o sistema bem como: realizar cadastros de entidades, atualizações de problemas, atualização dos problemas indicando qual entidade ele pertence.

Figura 6. Diagrama de Sequência do Uso do Sistema



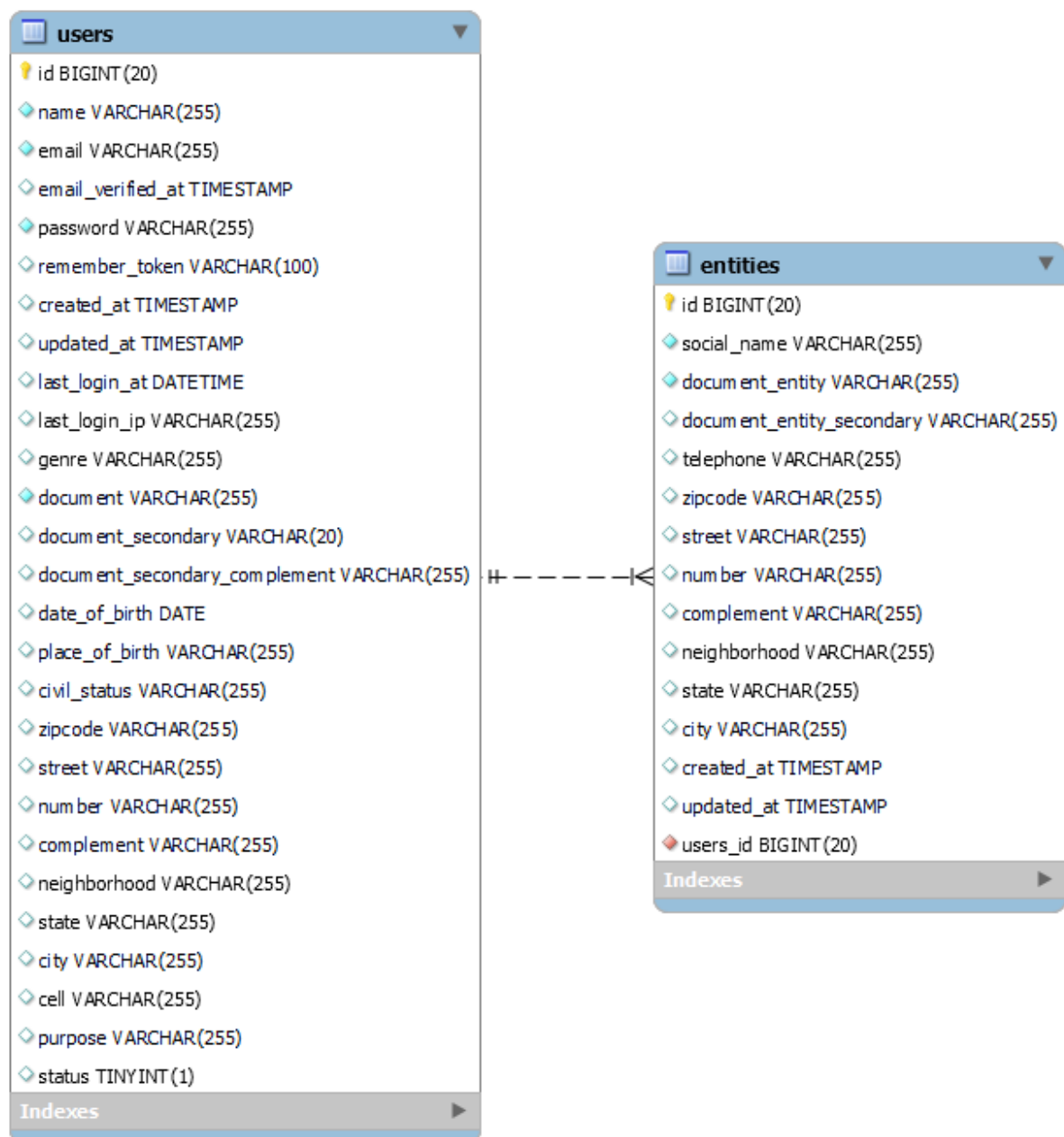
powered by Astah

Fonte: Adaptado de GUEDES (2018).

4.1.8 Diagrama de Mapeamento Objeto-Relacional (ORM)

Nesta seção é apresentado o diagrama MOR (Mapeamento Objeto Relacional) que é definido como o ato de conversão de objetos em memória para dados relacionais, e vice-versa. Em geral, pressupõe-se que o modelo de dados já existe, e que temos que adaptar nosso sistema orientado a objetos para trabalhar com este esquema pré-existente (GEKHUNTER, 2019). A seguir a **Figura 5**, apresenta o Diagrama de Mapeamento Objeto Relacional do nosso sistema:

Figura 7. Diagrama de Mapeamento Objeto Relacional (ORM)

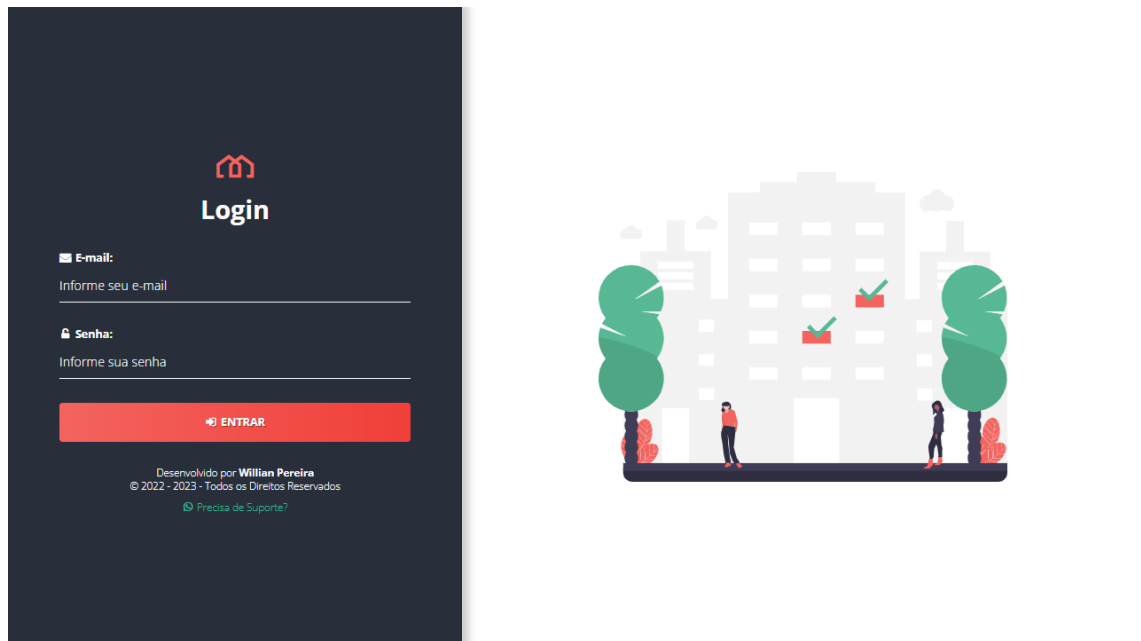


Fonte: Adaptado de GUEDES (2018).

4.2 Telas da Plataforma

A seguir nós temos a **figura 8**, que representa a tela de login da plataforma. Solicitando dados como e-mail e senha para acessar o sistema.

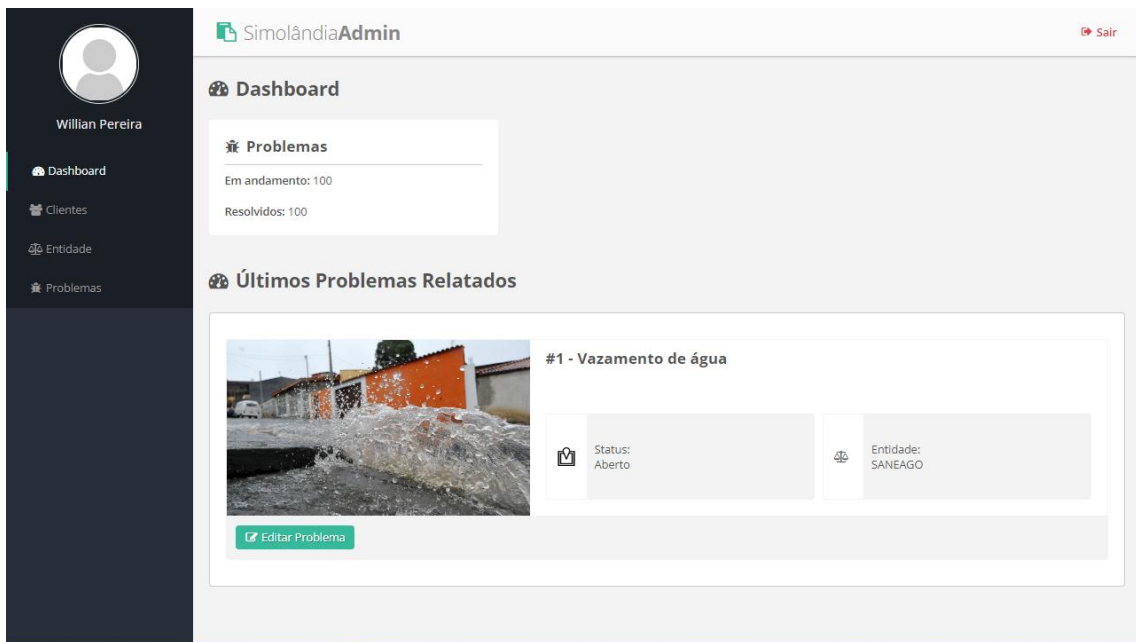
Figura 8. Tela de Login



Fonte: O Autor

A seguir temos a **figura 9**, que representa a tela de *dashboard* da plataforma. Tela essa que é exibida após o login. Com listagem dos 3 últimos problemas, problemas solucionados e não solucionados.

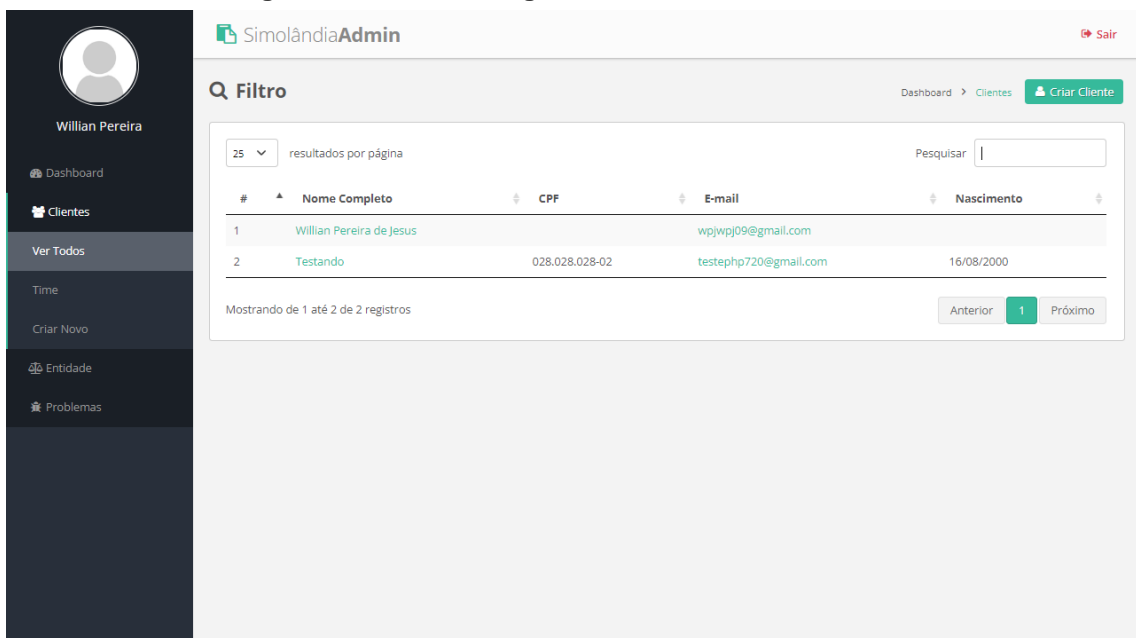
Figura 9. Tela de dashboard



Fonte: O Autor

A seguir temos a **figura 10**, que representa a tela de listagem de todos os usuários da plataforma ativos e inativos. Tela essa que pode ser acessada somente por usuários autenticados no sistema.

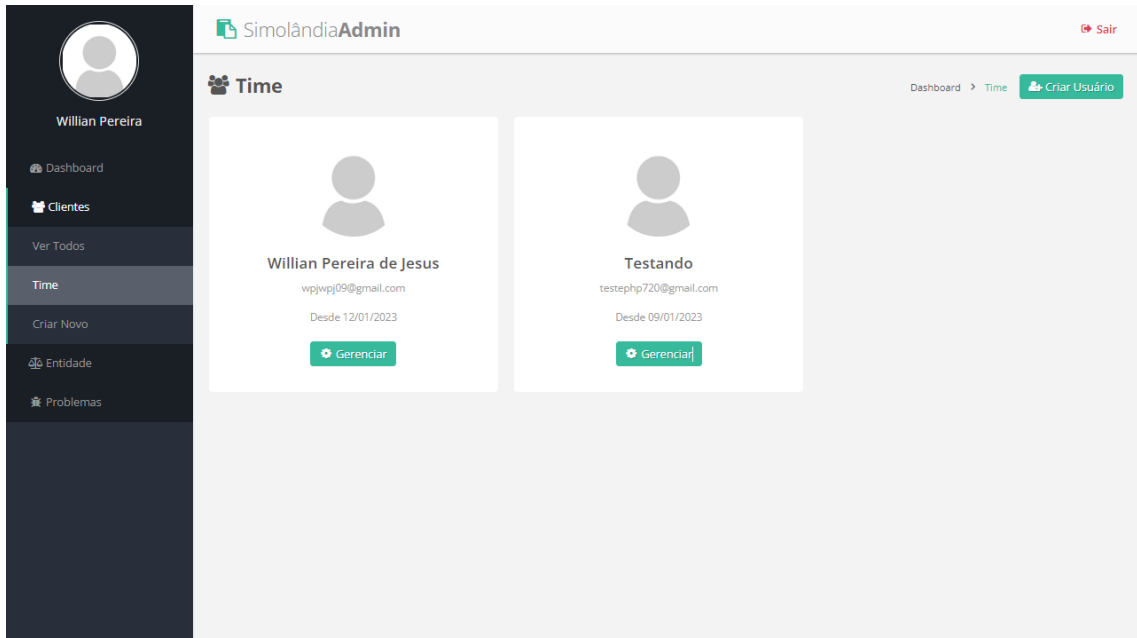
Figura 10. Tela de listagem de usuário ativos/inativos



Fonte: O Autor

A seguir temos a **figura 11**, que representa a tela de listagem somente dos usuários ativos. Tela essa que pode ser acessada somente por usuários autenticados no sistema.

Figura 11. Tela de listagem de usuário ativos



Fonte: O Autor

A seguir temos a **figura 12**, que representa o cadastro de usuários e também de edição. Tela essa que pode ser acessada somente pelo *Superadmin* e pelo *Admin* do sistema.

Figura 12. Tela de cadastro e edição de usuário

The screenshot shows the 'Novo Cliente' (New Client) form. The sidebar is identical to Figure 11, with 'Criar Novo' selected. The main area is titled 'Novo Cliente' and includes a breadcrumb 'Dashboard > Clientes > Novo Cliente'. A 'Dados Cadastrais' tab is active. At the top, there are radio buttons for 'Administrativo' (selected) and 'Colaborador'. The form contains several input fields:

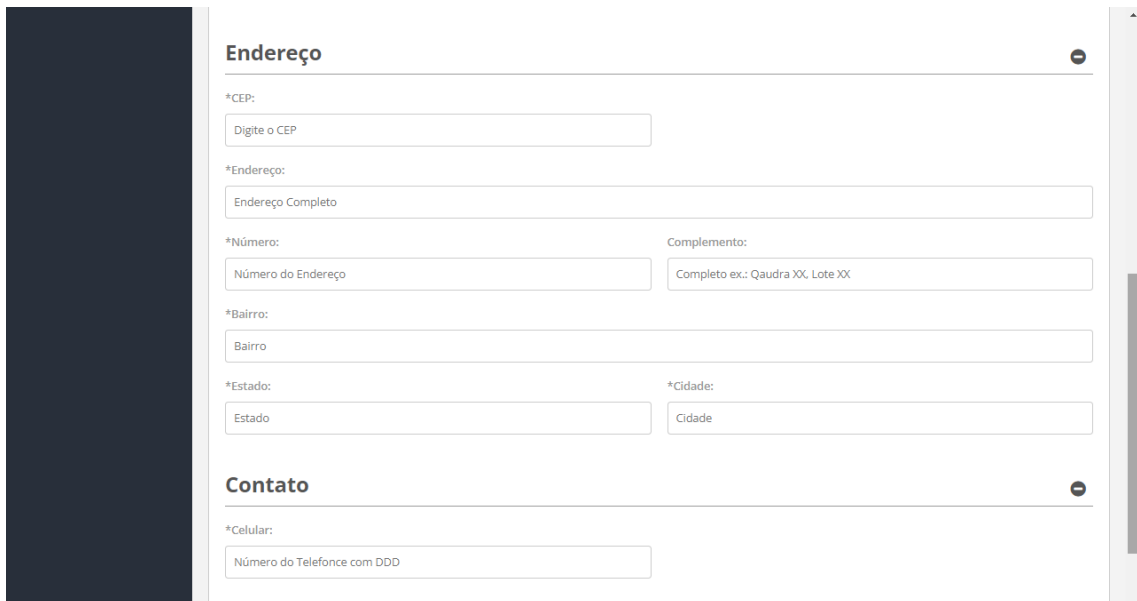
- *Nome: A text input field with the placeholder 'Nome Completo'.
- *Genero: A dropdown menu with 'Escolha...' selected.
- *CPF: A text input field with the placeholder 'CPF do Cliente'.
- *RG: A text input field with the placeholder 'RG do Cliente'.
- Órgão Expedidor: A text input field with the placeholder 'Expedição'.
- *Data de Nascimento: A text input field with the placeholder 'Data de Nascimento'.
- *Naturalidade: A text input field with the placeholder 'wpjwpj09@gmail.com'.
- *Estado Civil: A dropdown menu with 'Escolha...' selected.

 At the bottom, there is a section for 'Endereço' with a minus sign icon to collapse it.

Fonte: O Autor

Continuação da tela anterior, **figura 13**.

Figura 13. Continuação da tela anterior



Endereço

*CEP:
Digite o CEP

*Endereço:
Endereço Completo

*Número: Número do Endereço
Complemento: Completo ex.: Qaudra XX, Lote XX

*Bairro:
Bairro

*Estado: Estado
*Cidade: Cidade

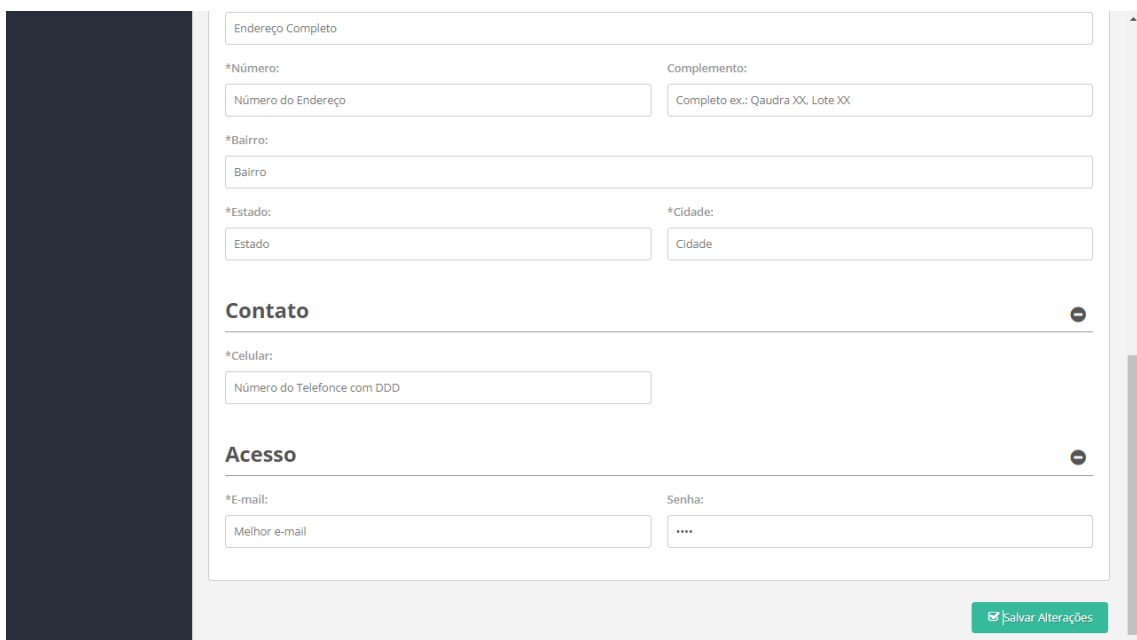
Contato

*Celular:
Número do Telefone com DDD

Fonte: O Autor

Continuação da tela anterior, **figura 14**.

Figura 14. Continuação da tela anterior



Endereço Completo

*Número: Número do Endereço
Complemento: Completo ex.: Qaudra XX, Lote XX

*Bairro:
Bairro

*Estado: Estado
*Cidade: Cidade

Contato

*Celular:
Número do Telefone com DDD

Acesso

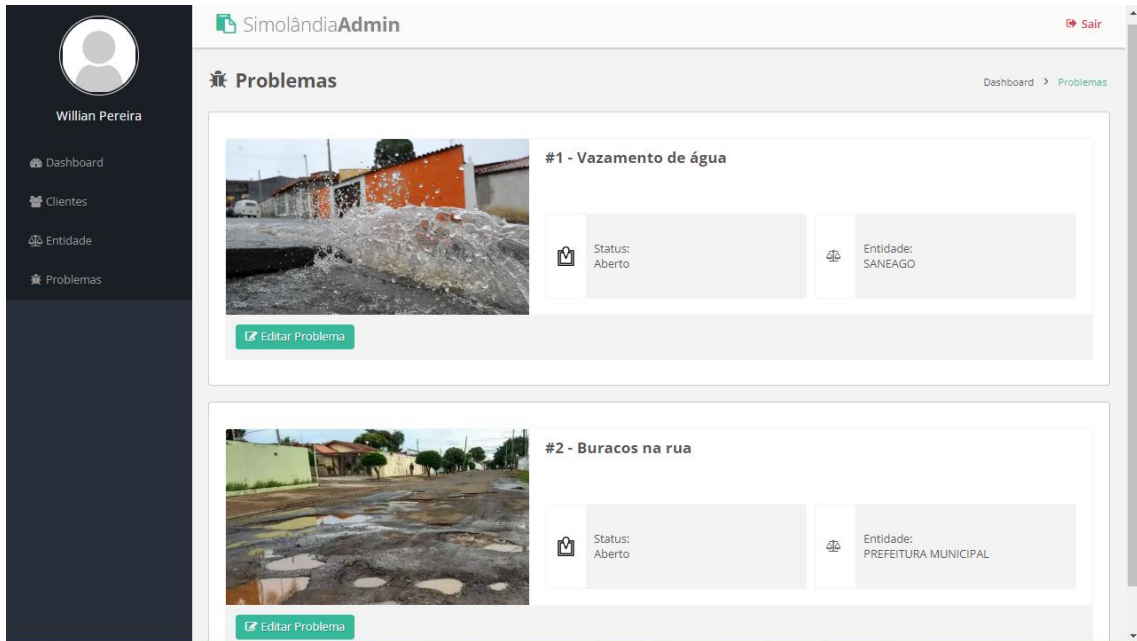
*E-mail: Melhor e-mail
Senha:

Salvar Alterações

Fonte: O Autor

A seguir temos a **figura 15**, que representa a listagem de todos os problemas. Tela essa que pode ser acessada por todos logados no sistema.

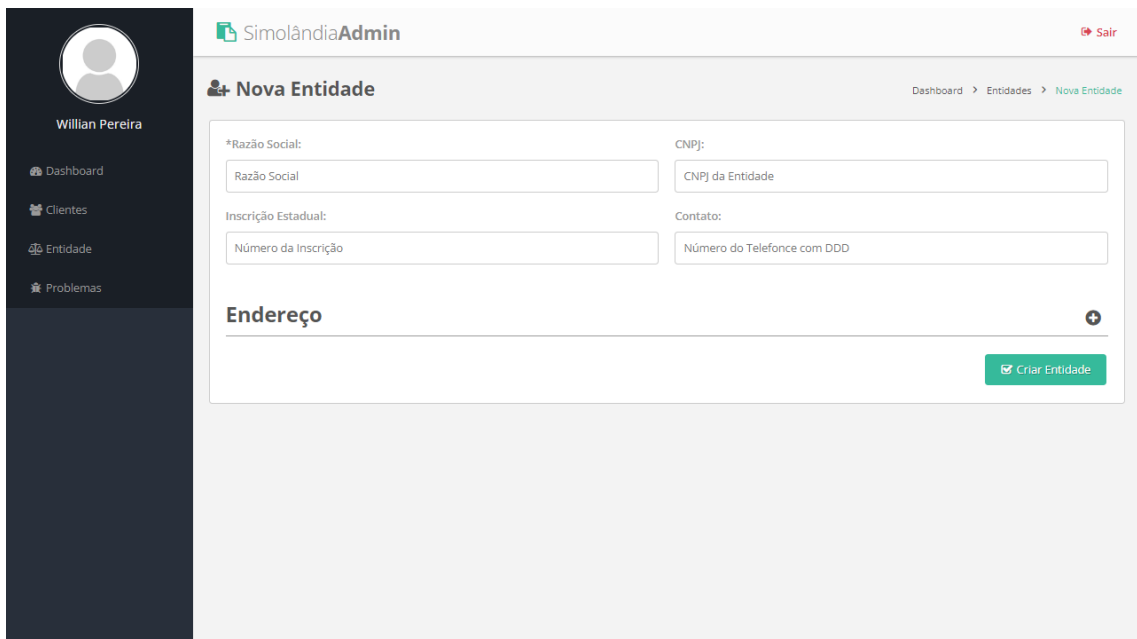
Figura 15. Tela de listagem de problemas



Fonte: O Autor

A seguir temos a **figura 16**, que representa o cadastro das entidades e também de edição. Tela essa que pode ser acessada por todos usuários autenticados no sistema.

Figura 16. Tela de cadastro de entidades



Fonte: O Autor

Continuação da tela anterior, **figura 17**.

Figura 17. Continuação da tela anterior

The screenshot shows a web form for creating an entity. On the left is a dark sidebar with navigation items: Dashboard, Clientes, Entidade, and Problemas. The main content area has the following fields:

- *Razão Social: Razão Social
- CNPJ: CNPJ da Entidade
- Inscrição Estadual: Número da Inscrição
- Contato: Número do Telefone com DDD
- Endereço** (Section Header)
- *CEP: Digite o CEP
- *Endereço: Endereço Completo
- *Número: Número do Endereço
- Complemento: Complemento
- *Bairro: Bairro
- *Estado: Estado
- *Cidade: Cidade
- Button: Criar Entidade

Fonte: O Autor

A seguir temos a **figura 18**, que representa a tela de listagem de todas as entidades cadastradas na plataforma. Tela essa que pode ser acessada somente por usuários autenticados no sistema.

Figura 18. Tela de listagem das entidades

The screenshot shows the 'Entidades' list page in the SimolândiaAdmin system. The page header includes the user profile 'Willian Pereira' and the system name 'SimolândiaAdmin'. The main content area shows a table of entities with the following data:

Razão Social	CNPJ	Endereço	IE
SANEAGO	12.345.678/0001-00	Rua central, lote Central	656555555

Additional details from the screenshot include: '25 resultados por página', 'Pesquisar' field, 'Mostrando de 1 até 1 de 1 registros', and navigation buttons 'Anterior', '1', and 'Próximo'.

Fonte: O Autor

A seguir temos a **figura 19**, que representa a tela edição do problema. Tela essa que pode ser acessada somente por usuários autenticados no sistema.

Figura 19. Tela de edição do problema

The screenshot shows a web application interface for editing a problem report. On the left is a dark sidebar with the user's name 'Willian Pereira de Jesus' and navigation options: 'Dashboard', 'Usuários', 'Entidade', 'Problemas', and 'Ver Todos'. The main content area has two tabs: 'Problemas' (active) and 'Imagens do problema'. The form contains the following fields:

- Título do Problema:** Lixo
- Descrição do Problema:** Lixo na frente da minha casa
- *Entidade Competente:** ADM
- *Status do Problema:** Resolvido
- *Solução do problema:** Problema solucionado

Below the form is a map section titled 'Localização no Mapa'. It features a Google Maps interface with a red pin marking the location. The map shows a residential street grid with various landmarks labeled, including 'MATERIAS P. CONSTRUÇÃO', 'LIDER GÁS ZEZÃO', 'Panificadora KI DELICIA', 'Mercado Primavera', and 'Gramado'. The map includes standard navigation controls like zoom in (+) and zoom out (-) buttons. At the bottom right of the map area is a green button labeled 'Editar Problema'.

Fonte: O Autor

6 CONCLUSÃO

Na tentativa de investigar e discutir acerca de como são gerenciados os problemas urbanos enfrentados, foi realizada uma pesquisa-ação aplicando-se um questionário afim aferir os problemas enfrentados na prefeitura municipal de Simolândia-GO, bem como avaliar a sua eficácia com os métodos e ferramentas que já são utilizadas pela gestão pública do local.

O resultado da pesquisa-ação possibilitou realizar uma análise aprofundada em relação a gestão de problemas urbanos, em conjunto com o aprimoramento das tecnologias para este fim. E com isso, o objetivo de desenvolver uma plataforma *Web* destinada para gerenciar os problemas urbanos, teve um melhor aprimoramento da organização e centralização de informações inerentes aos problemas urbanos, para uma boa relação entre gestor e cidadão.

Outrossim, foi observado que o sujeito da pesquisa demonstrou um notável interesse em utiliza-la e reconhecendo que a mesma seria de grande valia e auxílio para o município, acreditando em seu valor e na sua eficácia no tocante da centralização das informações que a plataforma proporciona através de suas funções.

Conclui-se que a plataforma foi desenvolvida com base nas tecnologias propostas, utilizando os princípios apresentados na justificativa e nos objetivos para a elaboração do mesmo foram alcançadas. Nela é possível ver a integração entre os cidadãos, relatando os problemas da cidade; o gestor da cidade, recebendo e redirecionado aos órgãos competentes, para manter a agenda do seu itinerário atualizada, realizar a manutenção da cidade e atender as solicitações.

7 REFERÊNCIAS

AGUNE, R. M. **Impacto do governo eletrônico da gestão governamental: observatório de Tecnologias de Informação**. In: 1er Congresso Íbero Americano de e-Government, out. 2006, Santiago-Chile. Anais... Santiago: Cátedra Software AG – Alianza Sumaq de e-Government - INCAE, 2006.

ALMEIDA, K. COSTA, Rostand. **Uma Proposta de Barramento de Dados para Integração de Serviços Públicos Digitais**, 2021. Trabalho de Conclusão de Curso, Centro de Informática – Universidade Federal de Paraíba. Paraíba, João Pessoa. Disponível em: <https://sol.sbc.org.br/index.php/wcge/article/download/15974/15815>. Acesso em 30 de junho de 2022.

ALENCAR, João Marcelo U. de. et. al. **Comparação de Desempenho entre Soluções de Interoperabilidade**, 2022. Disponível em: <https://sol.sbc.org.br/index.php/wcge/article/view/20708>. Acesso em 02 de agosto de 2022.

ALURA. **“PHP - UMA INTRODUÇÃO À LINGUAGEM”**, 2022, <https://www.alura.com.br/artigos/php-uma-introducao-linguagem>. Acesso em 05 de dezembro de 2022.

BALBE, Ronald da Silva. **Uso de tecnologias de informação e comunicação na gestão pública: exemplos no Governo Federal**. Revista do Serviço Público, Brasília, v. 61, n. 2, p. 189-209, 2010.

BERÇA, Douglas Diniz. **INTEROPERABILIDADE DE APLICAÇÕES COM SOAP (SIMPLE OBJECT ACCESS PROTOCOL)**, 2021. Relatório Técnico Científico, Universidade Estadual de Goiás, Campus Posse. Disponível em <http://179.255.33.190:8081/jspui/handle/123456789/272>. Acesso em 10 de agosto de 2022.

COMPUTERWEEKLY. **Interface de programação de aplicação (API)**, 2021. Disponível em: <https://www.computerweekly.com/br/definicoe/Interface-de-programacao-de-aplicacao-API>. Acesso em 31 de julho de 2022.

CONTEÚDO EDITORIAL. **“Estudo aponta crescimento de 13% no Setor de Distribuição de TIC em 2020”**, 2021. Disponível em: <https://www.decisionreport.com.br/mercado/estudo-aponta-crescimento-de-13-no-setor-de-distribuicao-de-tic-em-2020/#.Y5pzHHaZPb1>. Acesso em 14 de dezembro de 2022.

DHEIN, Jonas Alberto. **Software de Gerenciamento de Serviços Urbanos, p. 15**, 2009, Disponível em: <https://univates.com.br/bdu/handle/10737/90>. Acesso em 26 de maio de 2022.

DOBZHANSKY, Theodosius. “**A EVOLUÇÃO HUMANA.**” **Revista de Antropologia**, vol. 4, nº. 2, pp. 98, dez. 1956. JSTOR. Disponível em: <http://www.jstor.org/stable/44114035>. Acesso em 09 agosto de 2022.

DURÃES, Geison. **REST X SOAP, Qual o melhor tipo de integração**, 2020. Disponível em: <https://blog.tecnospeed.com.br/rest-x-soap>. Acesso em: 10 de agosto de 2021.

EULALIO, Athos Denis.; CORDEIRO, Diego.; SOUZA, Rodrigo de. **WEB SERVICES: Integração De Sistemas Orientado a Serviços com uma Proposta de Aplicação na EAD**, Revista Informática Aplicada, vol. 12, nº 2, p. 57 - 63, abril. 2017. Disponível em: https://seer.uscs.edu.br/index.php/revista_informatica_aplicada/article/view/6918. Acesso em 15 de agosto de 2022.

FACIVI. “**INTRODUÇÃO AO PHP**”, 2001, https://saulo.arisa.com.br/aulas/bd1/progweb/php_intro.html. Acesso em 05 de dezembro de 2022.

GOV.BR. **DECRETO Nº 10.332**, de 28 de abril de 2020. Dispõe sobre a instrução estratégica de governo digital para o período de 2020 a 2022. **Diário Oficial da União**: Ed. 81 de 29/04/2020, seção 1, p. 6. Disponível em: <https://www.in.gov.br/web/dou/-/decreto-n-10.332-de-28-de-abril-de-2020-254430358>. Acesso em 30 de junho de 2022.

GUEDES, G. T. A. (2018). “**UML 2 UMA ABORDAGEM PRÁTICA**”, (3ª ed.) São Paulo: Novatec.

GEEKHUNTER. “**Mapeamento objeto-relacional: como funciona e técnicas**”, 2019. Disponível em: <https://blog.geekhunter.com.br/mapeamento-objeto-relacional/>. Acesso em 14 de agosto de 2022.

IBM. “**Diagramas de Classes**”, 2022. Disponível em: <https://www.ibm.com/docs/pt-br/rsas/7.5.0?topic=structure-class-diagrams>. Acesso em 13 de dezembro de 2022.

MARTINS, K. **Expansão urbana desordenada e aumento dos riscos ambientais à saúde humana: o caso brasileiro**, 2012. Trabalho de Conclusão de Curso, Universidade de Brasília, Planaltina. Disponível em: http://bdm.unb.br/bitstream/10483/4135/1/2012_KarlaGoncalvesMartins.pdf. Acesso em 27 de junho de 2022.

MDN. **HTTP**, 2022. Disponível em: <https://developer.mozilla.org/pt-BR/docs/Web/HTTP>. Acesso em 13 de agosto de 2022.

MONTEIRO, Marcus Vinícios Brito. VASQUES, Letícia Veiga. **APLICAÇÕES MÓVEIS E O SEU AUXÍLIO AO HOMEM: como a tecnologia mobile pode aumentar a produtividade das pessoas**, 2016. Disponível em <http://192.100.247.84/handle/prefix/468>. Acesso em 09 de agosto de 2022.

MORAIS, Emilie Trindade de, BATISTA, Ítalo Paiva. **Implementação de uma interface de programação e uma arquitetura de plugins para a plataforma de participação social “Empurrando Juntos”**, 2016. Disponível em <https://bdm.unb.br/handle/10483/19893>. Acesso em 09 de agosto de 2022.

MOVIDESK. **“REGRAS DE NEGÓCIO EM TI: DESCUBRA ESSE IMPORTANTE CONCEITO E ENTENDA COMO APLICÁ-LO NO SEU NEGÓCIO”**, 11 de dez. de 2020. Disponível em: <https://conteudo.movidesk.com/regras-de-negocio-ti/#o-que-sao-regras-de-negocio-em-ti>. Acesso em 28 de novembro de 2022.

OLIVEIRA M.; Eric C. **Web Services: Java e XML juntos pela interoperabilidade, 2020**. Disponível em: <http://www.linhadecodigo.com.br/artigo/378/web-services-java-e-xml-juntos-pela-interoperabilidade.aspx>. Acesso em: 13 de agosto de 2021.

PIRES, Jackson. **O que é API? REST e RESTful? Conheça as definições e diferenças**, 2018. Disponível em: <https://becode.com.br/o-que-e-api-rest-e-restful/>. Acesso em: 10 de agosto de 2021.

PLANALTO. **“Institui o portal único “gov.br” e dispõe sobre as regras de unificação dos canais digitais do Governo federal”**, 2019. Disponível em: http://www.planalto.gov.br/ccivil_03/_ato2019-2022/2019/decreto/D9756.htm. Acesso em 14 de dezembro de 2022.

Red Hat. **O que é uma API?**, 2017. Disponível em: <https://www.redhat.com/pt-br/topics/api/what-are-application-programming-interfaces>. Acesso em 16 de julho de 2022.

RODRIGUES, Ricardo B. et al. **“A cloud-based recommendation model”**. In: EURO AMERICAN CONFERENCE ON TELEMATICS AND INFORMATION SYSTEMS, 7., 2014. Proceedings... 2014.

SUZART, Janilson Antonio da Silva. **“Siafi: o nível de satisfação dos usuários e os investimentos em tecnologia no setor público, uma análise à luz dos pressupostos da racionalidade limitada”**, 2010. Disponível em: <https://repositorio.ufba.br/handle/ri/8481>. Acesso em 14 de dezembro de 2022.

VISURE. **“O QUE SÃO REQUISITOS FUNCIONAIS: EXEMPLOS, DEFINIÇÃO, GUIA COMPLETO”**, 2022. Disponível em: <https://visuresolutions.com/pt/blog/functional-requirements>. Acesso em 28 de novembro de 2022.

VISURE. **“O QUE SÃO REQUISITOS NÃO FUNCIONAIS: EXEMPLOS, DEFINIÇÃO, GUIA COMPLETO”**, 2022. Disponível em: <https://visuresolutions.com/pt/blog/non-functional-requirements>. Acesso em 28 de novembro de 2022.

W3TECHS. **“SERVER-SIDE PROGRAMMING LANGUAGES”** 2022.
<https://w3techs.com>. Acesso em 05 de dezembro de 2022.

8 ANEXOS

Anexo 1. Questionário para analisar e definir requisitos

1) Vocês utilizam algum sistema de automação ou planilhas para mapear os problemas?

Sim

Não

2) Se sim qual sistema:

3) O que gostaria que o sistema fornecesse:

4) O cidadão tem algum feedback com relação as solicitações feitas:

Sim

Não

5) Como vocês relatam para o cidadão se o problema foi direcionado para o departamento correto?

6) Existe algum mecanismo pra fazer o direcionamento desses problemas para cada órgão competente?

Sim

Não

7) Se sim qual seria esse mecanismo:
