

UNIVERSIDADE FEDERAL DO PARANÁ

GUSTAVO KOVALSKI SAPORITI

LUCAS RIBEIRO

SAMUEL TOLEDO DE ALMEIDA

**PORTAL CIDADÃO:**

**APROXIMANDO O CIDADÃO E O PODER PÚBLICO**

CURITIBA

2021

GUSTAVO KOVALSKI SAPORITI

LUCAS RIBEIRO

SAMUEL TOLEDO DE ALMEIDA

## **PORTAL CIDADÃO: APROXIMANDO O CIDADÃO E O PODER PÚBLICO**

Monografia apresentada como requisito parcial à obtenção do grau de Tecnólogo em Análise e Desenvolvimento de Sistemas, do Setor de Educação Profissional e Tecnológica da Universidade Federal do Paraná.

Orientador: Prof. Dr. Alessandro Brawerman

CURITIBA

2021



UNIVERSIDADE FEDERAL DO PARANÁ

ATA DE REUNIÃO

**TERMO DE APROVAÇÃO**

GUSTAVO KOVALSKI SAPORITI  
LUCAS RIBEIRO  
SAMUEL TOLEDO DE ALMEIDA

**PORTAL CIDADÃO: APROXIMANDO O CIDADÃO E O PODER PÚBLICO**

Monografia aprovada como requisito parcial à obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas, do Setor de Educação Profissional e Tecnológica da Universidade Federal do Paraná.

Prof. Dr. Alessandro Brawerman  
Orientadora – SEPT/UFPR

Prof. Dr. Pedro Rodrigues Torres Jr.  
SEPT/UFPR

Profa. Dra. Caroline Mazetto Mendes (Externa)  
SEPT/UFPR

**Curitiba, 21 de Dezembro de 2021.**



Documento assinado eletronicamente por **ALESSANDRO BRAWERMAN, PROFESSOR DO MAGISTERIO SUPERIOR**, em 21/12/2021, às 20:23, conforme art. 1º, III, "b", da Lei 11.419/2006.



Documento assinado eletronicamente por **Caroline Mazetto Mendes, Usuário Externo**, em 21/12/2021, às 20:34, conforme art. 1º, III, "b", da Lei 11.419/2006.



Documento assinado eletronicamente por **PEDRO RODRIGUES TORRES JUNIOR, PROFESSOR DO MAGISTERIO SUPERIOR**, em 07/01/2022, às 08:08, conforme art. 1º, III, "b", da Lei 11.419/2006.



A autenticidade do documento pode ser conferida [aqui](#) informando o código verificador **4145513** e o código CRC **3FD70C9C**.

## RESUMO

Com o crescimento acelerado das cidades e de suas populações, aumenta-se naturalmente a quantidade de incidentes urbanos. Ao mesmo tempo, o poder público possui dificuldades para atender todas as demandas da sua população, o que cria um distanciamento entre a percepção dos indivíduos e a dos gestores das cidades sobre os pontos de melhoria a serem implementados. Assim, a população necessita de uma ferramenta ágil para comunicar aos órgãos públicos e aos outros cidadãos os relatos de acontecimentos nas cidades. Essa ferramenta também poderia facilmente auxiliar na administração urbana, fazendo com que a população tenha uma participação mais ativa nos problemas existentes, criando postagens relativas a reclamações, elogios ou sugestões em locais da cidade, gerando indicadores que serão visualizados pelos órgãos públicos para auxiliar na tomada de decisão. Os cidadãos também podem visualizar e interagir com outras postagens relatadas por usuários ao seu redor, colaborando para a melhoria da cidade. Este projeto propõe o desenvolvimento de um sistema em nuvem, moderno, multiplataforma e de simples utilização, que possibilita a aproximação entre os cidadãos e o poder público. Portanto, o projeto proposto visa facilitar essa relação, com o objetivo de melhorar a gestão dos órgãos públicos e a qualidade de vida dos cidadãos, utilizando-se da tecnologia para implementar medidas de cidades inteligentes na sociedade. O projeto apresentado viabiliza uma solução viável e eficaz para a melhoria das cidades, bem como uma ferramenta útil para beneficiar o cotidiano dos cidadãos que irão utilizá-la.

**Palavras-chave:** Cidades Inteligentes. Sistemas em nuvem. Tecnologia para cidades.

## **ABSTRACT**

With the rapid growth of cities and their populations, the number of urban incidents naturally increases. Meanwhile, the government has difficulties meeting all demands of its population, creating a gap between the perception of the individuals and that of city managers about the points of improvement to be implemented. Accordingly, people need an agile tool to communicate reports of events in cities to public bodies and other citizens. This tool would also help in urban management, making people more able to participate in existing problems, by publishing posts about complaints, suggestions, and compliments at any point of the city, generating indicators that will be viewed by public agencies to help decisions making. The citizens can also visualize and interact with other posts created by users around them, collaborating to the city improvement. This project proposes the construction of a modern, multi-platform and simple-to-use cloud system, which makes it possible to bring citizens and government closer together. The proposed project aims to facilitate this relationship, aiming to improve the management of public bodies and also the citizens' quality of life, using technology to implement measures of smart cities in society. The project to be presented demonstrates that the proposed tool is a viable and effective solution for the improvement of cities, as well as a useful tool to benefit the daily lives of citizens who will use it.

**Key-words:** Smart cities. Cloud systems. Technologies for cities.

## LISTA DE ILUSTRAÇÕES

FIGURA 1: REVOLUÇÃO DIGITAL ATRAVÉS DE APLICATIVOS

FIGURA 2: APLICATIVO DUBAINOW

FIGURA 3: O QUADRO SCRUM

FIGURA 4: RANKING DE FRAMEWORKS MULTIPLATAFORMA

FIGURA 5: APLICATIVO SAUDE JÁ

FIGURA 6: APLICATIVO SMARTSANTANDER

FIGURA 7: APLICATIVO DO PORTAL 156

FIGURA 8 ARQUITETURA DO SISTEMA

FIGURA 9: EXEMPLO DE PERFIL

FIGURA 10: PÁGINA DE LOGIN

FIGURA 11: PÁGINA DE NOVO USUÁRIO

FIGURA 12: PÁGINA PRINCIPAL

FIGURA 13: PÁGINA DE NOVA POSTAGEM

FIGURA 14: PÁGINA DE EXIBIÇÃO DE POSTAGEM

FIGURA 15: PÁGINA DE LOGIN PARA DISPOSITIVOS MÓVEIS

FIGURA 16: PÁGINA NOVO USUÁRIO EM DISPOSITIVOS MÓVEIS

FIGURA 17: NOVA POSTAGEM EM DISPOSITIVOS MÓVEIS

FIGURA 18: PÁGINA DE DETALHES PARA DISPOSITIVOS MÓVEIS

FIGURA 19: DIGRAMA CASO DE USO – GERAL

FIGURA 20: DV01 – TELA INICIAL APLICATIVO

FIGURA 21: DV02 – TELA CADASTRO USUÁRIO WEB

FIGURA 22: DV01 – TELA DE LOGIN DISPOSITIVO MÓVEL

FIGURA 23: DV02 – TELA DE LOGIN WEB

FIGURA 24: DV01 – TELA NOVA POSTAGEM

FIGURA 25: DV01 – TELA NOVA POSTAGEM WEB

FIGURA 26: DV01 – TELA COMENTAR POSTAGEM

FIGURA 27: DV01 – TELA CURTIR POSTAGEM

FIGURA 28: DV01 – TELA RESOLVER POSTAGEM

FIGURA 29: DV01 – TELA RESOLVER POSTAGEM

FIGURA 30: DV01 – TELA VISUALIZAR DASHBOARDS - ADMINISTRADOR

FIGURA 31: DV01 – TELA VISUALIZAR DASHBOARDS - ADMINISTRADOR

FIGURA 32: DIAGRAMA DE CLASSES

FIGURA 33: DIAGRAMA DE ENTIDADE RELACIONAMENTO

FIGURA 34: COMENTAR POSTAGEM

FIGURA 35: CURTIR POSTAGEM

FIGURA 36: FAZER CADASTRO

FIGURA 37: FAZER LOGIN

FIGURA 38: FAZER POSTAGEM

FIGURA 39: FILTRAR POR BAIRRO

FIGURA 40: RESOLVER POST

FIGURA 41: VISUALIZAR DASHBOARDS

FIGURA 42: VISUALIZAR POSTAGENS

FIGURA 43: DIAGRAMA GANTT 1

FIGURA 44: DIAGRAMA DE GANTT 2

FIGURA 45: DOCUMENTAÇÃO DE API VIA SWAGGER



## LISTA DE ABREVIATURAS E SIGLAS

API	- Application Programming Interface
CSS	- Cascading Style Sheets
W3C	- World Wide Web Consortium
REST	- Representational state transfer
UML	- Unified Modelling Language
MER	- Modelo Entidade Relacionamento
DER	- Diagrama Entidade Relacionamento
AWS	- Amazon Web Services
CRUD	- Create, Read, Update, Delete
URL	- Uniform Resource Locator
DV	- Data View
GPS	- Global Positioning System
HTTP	- Hypertext Transfer Protocol
HTML	- Hypertext Markup Language
IBGE	- Instituto Brasileiro de Geografia e Estatística
IOT	- Internet of Things
UFPR	- Universidade Federal do Paraná

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO.....</b>	<b>13</b>
<b>1.1</b>	<b>PROBLEMA.....</b>	<b>14</b>
<b>1.2</b>	<b>JUSTIFICATIVA.....</b>	<b>14</b>
<b>1.3</b>	<b>OBJETIVOS .....</b>	<b>15</b>
<b>1.3.1</b>	<b>OBJETIVO GERAL .....</b>	<b>15</b>
<b>1.3.2</b>	<b>OBJETIVOS ESPECÍFICOS .....</b>	<b>15</b>
<b>1.4</b>	<b>ESTRUTURA DO DOCUMENTO .....</b>	<b>16</b>
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA .....</b>	<b>18</b>
<b>2.1</b>	<b>CONCEITOS FUNDAMENTAIS .....</b>	<b>18</b>
<b>2.1.1</b>	<b>CIDADES INTELIGENTES .....</b>	<b>18</b>
<b>2.1.2</b>	<b>CONCEITOS DE ARQUITETURA DE SISTEMAS.....</b>	<b>21</b>
<b>2.1.3</b>	<b>MICROSOFT .NET.....</b>	<b>24</b>
<b>2.1.4</b>	<b>WEB SERVICES E SERVIÇOS RESTFUL .....</b>	<b>25</b>
<b>2.2</b>	<b>TRABALHOS RELACIONADOS .....</b>	<b>25</b>
<b>2.2.1</b>	<b>PORTAL CIDADÃO 2013 .....</b>	<b>26</b>
<b>2.2.2</b>	<b>APLICATIVO SAÚDE JÁ.....</b>	<b>27</b>

2.2.3	APLICATIVO SMARTSANTANDER .....	27
2.2.4	CENTRAL E PORTAL 156 .....	29
2.2.5	COLAB.RE .....	29
3	MATERIAIS E MÉTODOS .....	31
3.1	MÉTODOS DE DESENVOLVIMENTO DE SOFTWARE E SCRUM .....	31
3.2	FERRAMENTAS DE MODELAGEM .....	33
3.2.1	UML .....	33
3.2.2	ASTAH .....	33
3.3	FERRAMENTAS DE DESENVOLVIMENTO .....	34
3.3.1	VISUAL STUDIO CODE .....	34
3.3.2	VISUAL STUDIO .....	34
3.3.3	GIT .....	35
3.3.4	TRELLO .....	35
3.4	PLANO DE ATIVIDADES .....	35
3.5	RESPONSABILIDADES .....	37
3.6	TESTES DE SOFTWARE .....	37
4	APRESENTAÇÃO DO PORTAL CIDADÃO .....	39

4.1	ARQUITETURA DO SISTEMA .....	39
4.2	FUNCIONALIDADES .....	40
4.3	PERFIS DO SISTEMA.....	42
4.4	INDICADORES DE DADOS.....	43
4.5	VISÃO GERAL - SISTEMA WEB .....	44
4.6	VISÃO GERAL – APLICATIVO PARA DISPOSITIVOS MÓVEIS .....	50
4.7	SOBRE A DOCUMENTAÇÃO .....	55
4.7.1	REQUISITOS FUNCIONAIS .....	55
4.7.2	DIAGRAMA DE CASO DE USO.....	55
4.7.3	ESPECIFICAÇÃO DOS CASOS DE USO .....	56
4.7.4	DIAGRAMA DE CLASSES.....	56
4.7.5	DIAGRAMA ENTIDADE-RELACIONAMENTO (DER) .....	57
4.7.6	DIAGRAMA DE SEQUÊNCIA.....	57
4.7.7	DIAGRAMA DE GANTT .....	57
4.7.8	DOCUMENTAÇÃO DA API .....	58
5	CONSIDERAÇÕES FINAIS .....	59
	REFERÊNCIAS.....	61

<b>APÊNDICE A – LISTA DE REQUISITOS .....</b>	<b>67</b>
<b>APÊNDICE B – DIAGRAMA DE CASO DE USO GERAL .....</b>	<b>69</b>
<b>APÊNDICE C – ESPECIFICAÇÃO DOS CASOS DE USO.....</b>	<b>70</b>
<b>APÊNDICE D - DIAGRAMA DE CLASSES .....</b>	<b>87</b>
<b>APÊNDICE E - DIAGRAMA ENTIDADE RELACIONAMENTO .....</b>	<b>88</b>
<b>APÊNDICE F – DIAGRAMAS DE SEQUÊNCIA .....</b>	<b>89</b>
<b>APÊNDICE G – DIAGRAMA DE GANTT 1 .....</b>	<b>94</b>
<b>APÊNDICE H – DIAGRAMA DE GANTT 2 .....</b>	<b>95</b>
<b>APÊNDICE I – DOCUMENTAÇÃO DA API DO SISTEMA .....</b>	<b>96</b>

## 1 INTRODUÇÃO

A tecnologia está constantemente evoluindo no mundo moderno, sendo que um dos principais objetivos e benefícios desta evolução é a melhoria da qualidade de vida das pessoas. Nas cidades, questões como infraestrutura, energia, mobilidade e segurança são grandes focos de novas implementações tecnológicas. Assim surgiu o conceito das cidades inteligentes, que visa melhorar as condições das cidades utilizando a tecnologia a favor do cidadão e das entidades governamentais (CUNHA, 2016). Diversas cidades ao redor do mundo estão utilizando conceitos e implantando recursos de cidades inteligentes para acelerar seu crescimento econômico, assim como países desenvolvidos buscam aprimorar sua infraestrutura e oferecer melhores condições para a sociedade nas cidades (CUNHA, 2016).

O uso da tecnologia para melhorar a infraestrutura urbana é um dos conceitos das cidades inteligentes, que usam ações inovadoras para a melhoria de vários setores urbanos. Curitiba está na lista das grandes cidades inteligentes do Brasil, sendo classificada em terceiro lugar no ranking *Connected Smart Cities*, ficando apenas atrás de Florianópolis (2º) e São Paulo (1º) (REVISTA USE, 2020).

Atualmente com mais de 1,9 milhão de habitantes (IBGE, 2021), Curitiba, apesar da excelente colocação na lista de cidades inteligentes do Brasil, não é isenta de problemas - pelo contrário; em uma matéria do *Le Monde* (BADIA, 2014), a capital paranaense foi alvo de críticas sobre seu transporte público, segurança e saúde. Segundo o jornal, a capital, que já foi modelo mundial em vários setores, precisa “se reinventar”, pois a cidade foi descrita como congestionada, poluída e socialmente desequilibrada. Críticas foram apontadas para o sistema de transporte da cidade, que, uma vez admirado, agora está saturado e inadequado para o número crescente de passageiros; também para a área de segurança, particularmente nos bairros Pinheirinho e Cidade Industrial, com ruas mal iluminadas e vazias após o anoitecer; por fim, na área de saneamento público foi criticado o despejo de toneladas de resíduos em algumas áreas do sul da cidade, particularmente no rio Belém (BADIA, 2014).

## 1.1 PROBLEMA

Para muitas situações, o meio de comunicação entre o cidadão e os poderes públicos administrativos é escasso ou de difícil acesso pelo cidadão. Em algumas cidades o funcionamento de centros de ouvidoria é precário, o que cria uma barreira para o cidadão se comunicar, reclamar e avaliar itens relacionados à gestão da cidade. Tendo em vista o crescente avanço da tecnologia nas metrópoles, surgiram inovações com o objetivo de sanar essa distância de comunicação.

Em Curitiba, poucas plataformas para reclamação de problemas urbanos estão sendo utilizadas. Em 2014, a prefeitura da cidade aderiu ao uso do aplicativo Colab.re (HENRIQUE, 2014), uma rede que permite aos usuários fiscalizar problemas, sugerirem propostas e avaliarem espaços públicos. No entanto, algumas avaliações da *App Store* sugerem que o funcionamento do aplicativo possui diversos problemas, como a impossibilidade de editar um post, sendo necessário criar um, caso um erro seja cometido.

Sabendo dos inúmeros problemas que a cidade enfrenta em várias áreas, e contando com o bom uso da tecnologia atual, é possível que a adesão dos cidadãos a uma plataforma sólida e unificada, disponível tanto para dispositivos móveis quanto para *Web*, seja benéfica no auxílio a administração da cidade. Este tipo de solução ajudaria a sanar o problema do distanciamento entre os cidadãos e os poderes públicos administrativos, de forma a criar soluções pertinentes para o levantamento de problemas na cidade e para a eventual resolução deles.

## 1.2 JUSTIFICATIVA

Em 2017, foi realizado um estudo pelo *The World Population Prospects* para estimar a população mundial em 2030. Segundo o estudo, o aumento estimado foi de 2,6 bilhões de pessoas. Conforme aumenta a população, a presença de problemas urbanos tende também a subir; sendo assim, estratégias eficazes na solução dos mesmos são necessárias.

Existem problemas em maior escala, como o efeito estufa, provocado pelo aumento do uso de veículos a combustão e de fábricas que poluem o meio-ambiente, e problemas em menor escala, como buracos nas calçadas e ruas sem iluminação. O tratamento desses pelos órgãos responsáveis vai ficando cada vez mais difícil, visto

que reclamações são feitas constantemente através de inúmeros veículos diferentes; por exemplo, são comuns reclamações de problemas locais, como ruas esburacadas, para programas de televisão e de rádio.

Essa pauta de avaliar como as cidades enfrentam seus problemas, como elas se posicionam em relação ao que está acontecendo e como os cidadãos lidam com isso, é algo sempre presente no espaço urbano. Prova disso é a existência do Fórum Urbano Mundial, que desde 2001 se encontra anualmente para discutir questões como essas (VIZIA, 2010). Como grande parte da população da cidade possui acesso a celulares, uma plataforma digital, em nuvem, integrada a um aplicativo para dispositivos móveis, de fácil acesso para a postagem de problemas encontrados na cidade, em especial, quando alguém está fora de casa, torna-se uma ponte que possibilitaria a aproximação entre o cidadão e os órgãos responsáveis pela resolução de problemas urbanos.

### **1.3 OBJETIVOS**

Este projeto visa aproximar, a partir do uso de tecnologia, cidadãos e poder público das cidades, de forma que possa existir uma possível melhora na administração das mesmas, desenvolvendo uma plataforma denominada Portal Cidadão.

#### **1.3.1 OBJETIVO GERAL**

Desenvolver o Portal Cidadão - uma plataforma em nuvem, com sistema Web e aplicativo para dispositivos móveis, que torne possível a aproximação do cidadão e de órgãos públicos administrativos, que por um lado dê poder ao usuário para colaborar com a administração da cidade, a partir de relatos de eventos, levantamento de problemas e cobrança de soluções; e por outro, forneça indicadores para auxiliar aos órgãos públicos na tomada de decisão.

#### **1.3.2 OBJETIVOS ESPECÍFICOS**

Para que este trabalho alcance o objetivo geral, foram traçados alguns passos que servem como objetivos mais específicos, sendo eles:



- a) Desenvolver uma *API* em nuvem capaz de prover serviços para outros sistemas, que poderão realizar integração independentemente de sua plataforma;
- b) Gerenciar e armazenar dados provenientes dos usuários em um sistema de banco dados, de forma confiável e segura;
- c) Desenvolver uma interface front-end para a *Web*, disponibilizando-a para acesso dos usuários a partir de navegadores, de forma a prover a solução proposta na sua forma completa e com todas as suas funcionalidades;
- d) Desenvolver e integrar o sistema com um aplicativo de dispositivos móveis, a fim de disponibilizar a ferramenta construída em diferentes tipos de sistemas operacionais e dispositivos, bem como de fácil acesso, atingindo o maior número possível de usuários que se beneficiem da mesma;
- e) Propor e utilizar indicadores que auxiliem o poder público na tomada decisão estratégica para a melhoria contínua da qualidade de vida dos cidadãos;

#### **1.4 ESTRUTURA DO DOCUMENTO**

Todas as informações relacionadas ao desenvolvimento do Portal Cidadão estão contidas neste documento, que foi dividido conforme apresentação abaixo.

No Capítulo 2 é apresentada a fundamentação teórica do trabalho, são analisados conceitos chave de cidades inteligentes e projetos relacionados utilizados em algumas dessas cidades. Seguindo neste capítulo são expostos conceitos de arquitetura de sistema, junto com tecnologias utilizadas no projeto.

No Capítulo 3 são apresentados os materiais e métodos utilizados para o desenvolvimento do projeto, incluindo a metodologia *Scrum*, as ferramentas de modelagem utilizadas, as ferramentas de desenvolvimento, e as responsabilidades de cada membro da equipe.

No Capítulo 4 é apresentada a arquitetura de infraestrutura e informação utilizada no projeto, bem como é detalhado o sistema a ser desenvolvido, incluindo prototipação de telas.

No Capítulo 5 são apresentadas as considerações finais sobre o desenvolvimento do Portal Cidadão, incluindo uma análise dos objetivos iniciais do projeto, além de expor objetivos futuros para o projeto.

Nos apêndices estão incluídos os diagramas técnicos e outros documentos utilizados no projeto que especificam detalhadamente o sistema a ser desenvolvido.

## **2 FUNDAMENTAÇÃO TEÓRICA**

Este capítulo apresenta a base teórica para fundamentar os conhecimentos necessários para uma melhor compreensão deste projeto. O capítulo é dividido em duas partes, a primeira com conceitos teóricos sobre os assuntos envolvidos na pesquisa e a segunda tratando de questões sobre a arquitetura e tecnologias utilizadas no trabalho.

### **2.1 CONCEITOS FUNDAMENTAIS**

São abordadas aqui as questões pertinentes aos conceitos teóricos relacionados ao trabalho.

#### **2.1.1 CIDADES INTELIGENTES**

Uma cidade inteligente pode ser definida como uma cidade que utiliza a tecnologia para gerar serviços urbanos mais eficientes e melhorar a qualidade de vida dos cidadãos (CEAPG, 2016). Seu surgimento pode ser visto como uma resposta a fatores como o grande crescimento populacional das últimas décadas, a revolução digital que causou um aumento do acesso à Internet, e a dispositivos móveis, entre outros.

Essas alterações drásticas demandam também mudanças na gestão das cidades. Entre as mudanças, podemos citar a criação e utilização de ferramentas para dispositivos móveis, que possibilitem ao cidadão o rápido acesso a informações relevantes da cidade ao usuário, a fim de melhorar sua qualidade de vida na cidade.

A Figura 1 exemplifica a utilização de serviços nos dispositivos móveis por parte dos cidadãos, com fácil acesso a comércios e serviços à sua volta. Dessa forma, as cidades podem se aproveitar deste uso crescente de tais dispositivos, para também prover ao cidadão o acesso a ferramentas que auxiliem na gestão e melhoria da cidade.

FIGURA 1: REVOLUÇÃO DIGITAL ATRAVÉS DE APLICATIVOS



FONTE: PINHEIRO E CRIVELARO, 2020

As medidas consideradas “inteligentes” podem ser vistas como soluções para um problema específico de determinado setor da cidade (CUNHA, 2016). Por exemplo, no âmbito do meio ambiente, podemos destacar medidas que envolvem sustentabilidade, como sistemas inteligentes para a distribuição de energia, ou o uso da tecnologia para um melhor gerenciamento do lixo urbano.

O setor de mobilidade urbana pode ser visto como um dos mais importantes e que mais carecem de novas soluções. Sistemas de integração de ônibus e metrô são bastante obsoletos em algumas cidades; a renovação desses sistemas para algo mais integrado, que utilize recursos sustentáveis e reduza a emissão de CO<sub>2</sub> são objetivos importantes no âmbito das cidades inteligentes. Nesse setor a tecnologia é fundamental, deixando de ser uma ferramenta de suporte para tornar-se protagonista do funcionamento dos sistemas de transportes (CCR, 2017).

Na área de segurança pública, a administração da violência urbana faz grande uso da tecnologia. Centros de emergência para denúncia, vigilância por câmeras e análise de imagens são alguns dos usos recorrentes nessa área. Uma sugestão de um pesquisador do Centro de Estudos em Segurança Pública e Direitos Humanos da UFPR seria a criação de uma central que centralize todos os serviços, evidenciando qual órgão melhor atende determinada situação (MACEDO, 2019).

O setor de saúde é outra área na qual o uso da tecnologia ganha grande destaque. Com inúmeras inovações nos últimos anos, o setor passou por uma melhora considerável, mas diante de sua demanda constante, há modos de aplicar

novas tecnologias para otimizar sistemas administrativos, por exemplo, reduzindo filas e agilizando os processos; aplicativos que permitem agendar consultas ou verificar resultados de exames são usados por vários planos de saúde.

A área de educação, um dos principais serviços públicos do país, se beneficiou bastante com a inclusão de novas tecnologias na esfera escolar, que incorporou dispositivos novos com inúmeras funcionalidades, facilitando o trabalho dos professores e o acompanhamento dos alunos. Um exemplo de aplicação da tecnologia nessa área é a cidade de Águas de São Pedro, em São Paulo. Com apoio da empresa Telefônica, a iniciativa incorpora dispositivos digitais para alunos e professores, conteúdo online através da nuvem e comunicação entre a escola e os pais (TELEFONICA, 2015)

Em Dubai, o aplicativo *DubaiNow* ganhou destaque por incluir mais de 120 serviços relacionados ao governo para os usuários. A Figura 2 exibe a interface do aplicativo. Entre os serviços, é possível pagar contas, renovar licença de veículos, pagamento de estacionamento, chamar serviços de emergência, agendar consultas médicas, entre muitos outros (WAM, 2021).

FIGURA 2: APLICATIVO DUBAINOW



FONTE: DUBAI PULSE, 2018b

É possível notar que aplicativos móveis vêm sendo utilizados de maneira gradual na administração de uma cidade, sendo disponibilizados pelos órgãos públicos e a prefeitura. Percebe-se que a funcionalidade das plataformas se adequa aos principais dilemas da cidade de uma forma a conectar o usuário com serviços importantes. A seguir são apresentados alguns conceitos importantes sobre a criação

de plataformas digitais em nuvem; um entendimento de suas camadas auxilia o entendimento das funções de cada um nas cidades inteligentes.

### 2.1.2 CONCEITOS DE ARQUITETURA DE SISTEMAS

A arquitetura de software visa a estruturação de um sistema antes de seu desenvolvimento; assim, pode-se dizer que ela inclui a estrutura dos componentes, as interações entre eles e objetivos de um sistema, além das propriedades físicas e lógicas do mesmo (MARTIN, 2019). A escolha dos princípios a serem utilizados em um projeto impacta diretamente em itens como escalabilidade, qualidade de software, segurança e facilidade de manutenção.

Um padrão bastante utilizado no mercado atualmente é o modelo de arquitetura baseado em API. Uma API é uma interface que permite a comunicação com serviços diferentes. Uma das vantagens desse tipo de arquitetura, é a criação de métodos altamente reutilizáveis. Existem possibilidades diferentes de implementação desse padrão, como o uso de *Web APIs*, que são acessados por métodos HTTP ou o uso de um design orientado a eventos, muito utilizado em jogos mobile, onde o servidor e o cliente respondem a novos eventos de acordo com as instruções da API.

Conforme aumentou a complexidade dos projetos de software nas últimas décadas, surgiram não apenas novos padrões de arquitetura, mas também tecnologias novas que permitem a criação de interfaces mais amigáveis ao usuário, por exemplo. Conforme alguns conceitos de acessibilidade que envolvem a interação humano computador foram ganhando espaço no meio tecnológico, surgiram ferramentas específicas para trabalhar com a interface que o usuário acessa. Esse é o objetivo do desenvolvimento *front-end*, que aplica tecnologias como HTML, CSS e Javascript para a criação de sites e interfaces para sistemas *Web* (LINDLEY, 2019). A maioria dos sites modernos usa Javascript, e todos os navegadores modernos incluem interpretadores Javascript, de forma que se pode dizer que o HTML especifica o conteúdo das páginas *Web*, o CSS especifica a apresentação das páginas, e o Javascript é utilizado para especificar o comportamento delas (FLANAGAN et al., 2013).

Além das tecnologias front-end, o aumento da complexidade nos projetos de software contribuiu também para a criação de novos frameworks de desenvolvimento, que seguem a ideia de que reutilizar software significa aumentar sua qualidade e reduzir o esforço de desenvolvimento (GIMENES et al., 2005). São estruturas com funcionalidades já prontas para o desenvolvimento, o que agiliza o trabalho.

Um dos principais frameworks na área de *front-end* é o Angular (GOOGLE, 2010) que teve sua primeira versão, conhecida como AngularJS, lançada em 2009, com o objetivo de tornar o HTML mais robusto para o desenvolvimento de aplicações. Essa versão era bem diferente do Angular 2, que foi lançado em 2014. Esse framework utiliza Javascript para construir SPAs, *Single Page Applications*, que é uma aplicação Web construída em uma página só. Ele opera com sincronização automática de modelos e visualizações, usando a chamada *two-way data binding*, ou seja, sempre que o usuário mudar alguma coisa em um Modelo, a visualização é atualizada automaticamente.

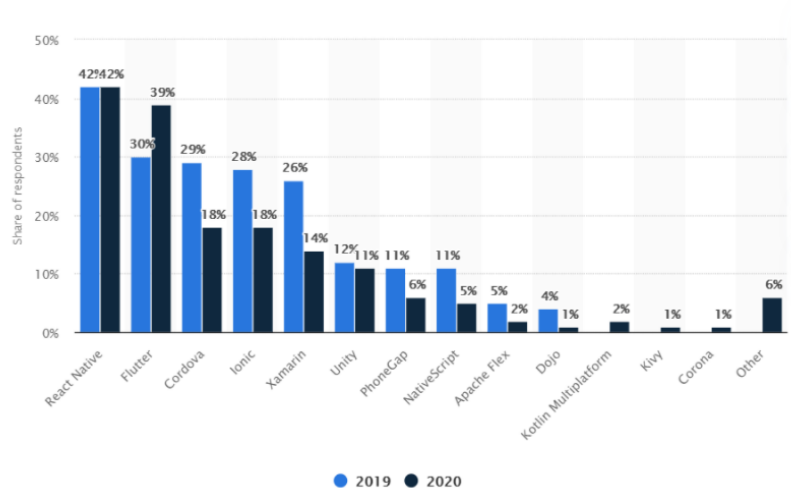
Uma característica interessante do Angular, e que teve impacto em seu uso neste projeto, é seu suporte *cross-plataform*, que são sistemas acessíveis em mais de uma plataforma, como dispositivos móveis e *Web*, a partir da mesma base de código; além das aplicações SPAs, é possível criar aplicações para dispositivos móveis com a mesma API.

Um outro framework bastante utilizado no mercado atualmente é o React Native (FACEBOOK, 2021). Ele é um framework Javascript para desenvolvimento de aplicações nativas de Android ou iOS. É baseado em uma biblioteca Javascript do Facebook, porém para plataformas para dispositivos móveis. O *framework* funciona com Javascript e uma ferramenta de marcação que se assemelha visualmente ao XML, conhecida como JSX. Também implementa uma espécie de ponte para as APIs do *framework*, que são diferentes no iOS e no Android.

Em relação a outros *frameworks* de desenvolvimento para dispositivos móveis, o React Native se destaca por trazer elementos nativos da interface de *smartphones*, o que em outros frameworks geralmente é feito por *webviews*, que são uma tentativa de “imitar” os elementos nativos do host (EISENMAN, 2015). Como não são usadas *webviews*, mas componentes nativos utilizados em *smartphones*, a plataforma dá ampla possibilidade de uso do celular para criação da aplicação, podendo ser utilizados recursos como a câmera do usuário, ou a localização.

Na Figura 4 é exibido um ranking dos *frameworks* multiplataforma para desenvolvimento móvel, de 2019 a 2021. Nele, é possível observar o React Native se mantendo entre os *frameworks* mais populares para os desenvolvedores.

FIGURA 4: RANKING DE FRAMEWORKS MULTIPLATAFORMA



FONTE: STATISTA, 2021

No geral, um bom conhecimento de Javascript é o suficiente para usar o React Native, o que é um grande facilitador na curva de aprendizado do framework.

Os padrões de arquitetura de *software* definem também funcionalidades e comportamentos específicos para o *back-end* do projeto, que se refere à parte da aplicação que cuida do funcionamento interno de um *software*. No padrão MVC (*Model-View-Controller*), por exemplo, o código é separado em camadas, sendo a camada *Model* a única que se comunica com o *back-end* da aplicação.

O *back-end* de um projeto inclui as operações realizadas, o banco de dados e a manipulação de informações (NORTHWOOD, 2018). Os desenvolvedores de *back-end* costumam criar e dar manutenção ao código que liga o banco de dados ao aplicativo. Algumas linguagens populares no desenvolvimento *back-end* são Java, PHP e .NET. Um projeto de *software* pode conter uma ou diversas tecnologias diferentes no *back-end*, dependendo dos objetivos da aplicação, como performance, facilidade de desenvolvimento ou integração com outros sistemas já existentes.

Além de dominar linguagens específicas, desenvolvedores *back-end* lidam com conceitos importantes no funcionamento de um sistema, como a escalabilidade, que se refere a quão flexível é uma aplicação caso ocorram mudanças de negócio,



alteração ou inclusão de código, ou um fluxo de dados maior que o esperado, por exemplo. Outro conceito importante é o uso de APIs – *Application Programming Interface* – que funciona como uma interface usada para conectar dois sistemas *back-end*. Também é essencial o cuidado com a segurança do sistema, aplicando práticas de controle de autenticação sempre que possível.

A manipulação do banco de dados, parte central do *back-end*, inclui o gerenciamento das tabelas, dos tipos de dados e dos relacionamentos entre as tabelas. O conhecimento de diversos bancos, como MySQL, PostgreSQL e SQL Server, é um diferencial importante entre as habilidades de um desenvolvedor *back-end*.

### 2.1.3 MICROSOFT .NET

O .NET é um *framework* lançado em 2000 pela Microsoft, que incorporou diversos recursos desenvolvidos pela empresa na década de 1990. Foi uma plataforma com objetivos ambiciosos por parte da empresa, incluindo um suporte otimizado, integração entre linguagens, desenvolvimento e compilação simples, e boa segurança (THAI, 2002). Possibilita o uso de diversas linguagens, como C#, F# e Visual Basic. Incorpora várias utilidades para o desenvolvimento de *software*, sendo os principais elementos a *Common Language Runtime*, um mecanismo feito para execução, e a biblioteca de classes, que contém códigos prontos para os programadores reutilizarem em seus aplicativos.

O componente principal do framework, a *Common Language Runtime*, é responsável pela execução do código; ele é responsável por ativar objetos e gerenciá-los. Na prática, ele irá carregar as classes necessárias, compilar o código e realizar análises de segurança (LIMA, 2002).

Uma grande vantagem do *framework* é a existência dos códigos reutilizáveis, o que poupa horas de desenvolvimento. Nas bibliotecas é possível encontrar funcionalidades prontas de diversos artefatos comuns no desenvolvimento, por exemplo conectores HTTP para chamadas à serviços externos, validações de requisições e tratamentos de erro.

#### 2.1.4 WEB SERVICES E SERVIÇOS RESTFUL

Em muitos projetos de software é indispensável que o programa seja portátil para meios diferentes, como uma parte do software estar presente em um aplicativo de celular que se comunica com um aplicativo instalado em uma máquina com Windows, por exemplo. Uma das soluções para criar essa ponte entre as plataformas são os *web services*, aplicações que fazem comunicação entre *software* em plataformas diferentes (RICHARDSON, 2007).

Os *web services* são vistos como aplicações cliente/servidor; segundo a W3C, podem ser definidos como sistemas capazes de proporcionar a interação entre duas máquinas através de uma rede (MORAES, 2013).

Um padrão bastante popular de serviços é o padrão REST, *Representational State Transfer*, que surgiu no início nos anos 2000 e tinha como objetivo definir as melhores práticas para a modelagem de serviços web. Tais práticas incluem a segregação de responsabilidades de *front-end* e *back-end*; *stateless* – ou seja, garantir que uma requisição não tenha ligação com requisições anteriores ou futuras; cache – possibilitar que as respostas para uma requisição idêntica não sejam reprocessadas; interface uniforme – recebendo e retornando mensagens no mesmo padrão em diversos recursos; sistema em camadas – possibilitar que elementos sejam adicionados na aplicação sem que isso afete o funcionamento aos usuários finais; e código sob demanda – uma prática que possibilita o código evoluir sem de forma extensível, sem afetar funcionalidades existentes. Uma implementação que utilize o padrão REST corretamente pode ser chamada de *RESTful*. (DIAS, 2016).

O conjunto de operações conhecido que é usado no padrão *RESTful* possibilita ao usuário criar (*POST*), visualizar (*GET*), alterar (*PUT*) e apagar (*DELETE*), através de mensagens HTTP.

## 2.2 TRABALHOS RELACIONADOS

Esta seção apresenta o resultado de uma pesquisa de trabalhos e aplicativos relacionados com esse projeto. Os trabalhos mais relevantes e semelhantes são apresentados.

### 2.2.1 PORTAL CIDADÃO 2013

Em 2013, um grupo de estudantes da UFPR, composto por Daniel Mamoré, Dionei Muller, Jessika Medeiros e Pedro Trevisani, fez o desenvolvimento de um sistema com o objetivo de divulgar pontos de interesse ou facilitar a divulgação de problemas na cidade (MAMORÉ et al., 2013).

Na época, a equipe optou por desenvolver um aplicativo para iOS utilizando a linguagem Objective-C e um portal *Web* utilizando a linguagem Java e alguns frameworks. *Web services* foram usados para interligar os sistemas. Por tratar-se de um trabalho de alguns anos atrás, as tecnologias utilizadas eram mais rústicas do que as disponíveis hoje, e a equipe não utilizou um *framework cross-compiling* para disponibilizar o aplicativo para Android e iPhone.

De forma resumida, o sistema desenvolvido funcionava começando pelo celular do usuário, que abre o aplicativo para inserir um post relatando algum problema observado. A localização do usuário é captada via GPS e enviada para o portal *Web* através do *Web service* implementado, com os outros detalhes inseridos pelo usuário.

O portal *Web* foi desenvolvido com o framework JSF (*Java Server Faces*), e atribui marcadores no mapa com a tecnologia GMaps de acordo com os dados recebidos pelo *Web service*. Os usuários são tratados em três níveis de acesso diferentes, visitante, usuário logado e administrador, cada um com seus privilégios. É possível, além de criar postagens, navegar no mapa visitando os posts de outros usuários, abrir uma postagem e fazer comentários.

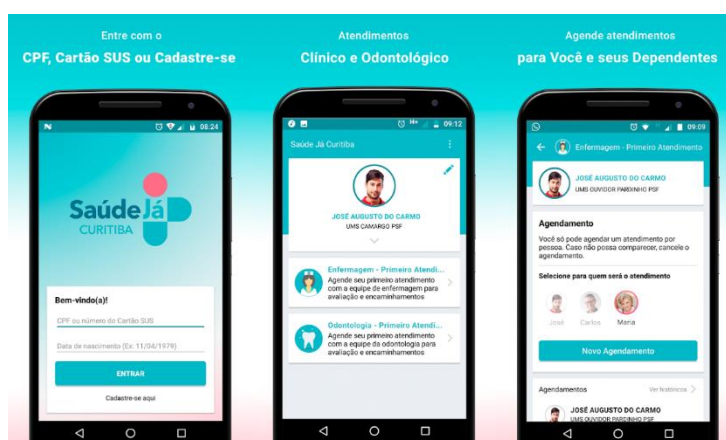
O trabalho atual, proposto neste documento, tem como seu objetivo principal incrementar drasticamente as funcionalidades antes propostas, utilizando as tecnologias mais atuais e relevantes proporcionando uma ótima experiência do usuário. Entretanto, nossa proposta inclui também a ideia de utilização de tecnologias capazes de apresentar indicadores de análise de dados, com o objetivo de obter uma visão detalhada de situações específicas na cidade, como quantidade de problemas de segurança em um determinado bairro. As análises de dados serão providas para perfis específicos no sistema, que serão pertencentes aos órgãos públicos. Com *dashboard* contendo gráficos e relatórios atualizados em tempo real, os gestores públicos poderão tomar decisões que afetarão diretamente a vida nas cidades.

### 2.2.2 APLICATIVO SAÚDE JÁ

Segundo o ICI, Instituto das Cidades Inteligentes, Curitiba é uma das cidades mais inteligentes do Brasil, destacando-se em suas ações o uso do aplicativo Saúde Já, que entre outras funcionalidades, possibilita o agendamento e visualização de resultados de exames feitos pelo SUS (ICI, 2021). O aplicativo recebeu um prêmio no *Latam City Awards 2018*, a premiação no México que avalia iniciativas para tornar as cidades mais inteligentes (ICI, 2021).

O aplicativo facilita diversos processos na área da saúde na capital. Nos últimos meses, com o aumento dos casos de coronavírus, o encaminhamento de exames pelo aplicativo facilitou bastante a vida dos moradores, que não precisavam se deslocar para o posto para ver o resultado do exame. Segundo a Secretaria Municipal de Saúde, 65% dos atendimentos são encaminhados e solucionados no mesmo dia. Na Figura 5 é possível visualizar algumas telas do aplicativo e suas funcionalidades.

FIGURA 5: APLICATIVO SAÚDE JÁ



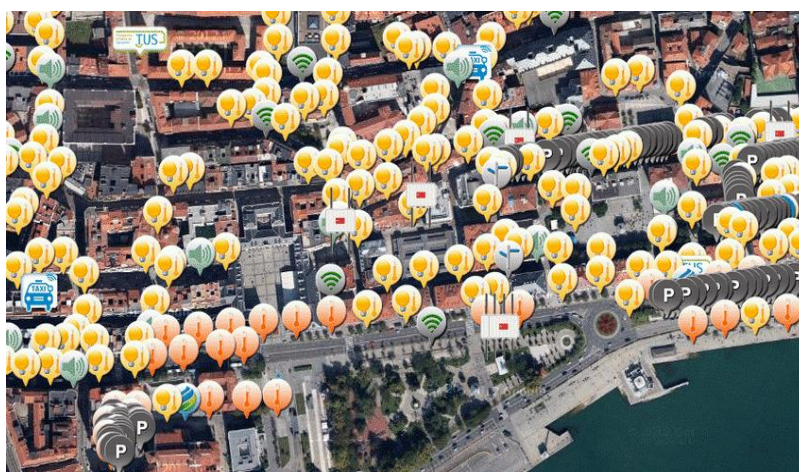
FONTE: INSTITUTO DAS CIDADES INTELIGENTES, 2020

### 2.2.3 APLICATIVO SMARTSANTANDER

Um outro exemplo de portal com plataforma digital para Cidades Inteligentes é o SmartSantander (SANTANA et al., 2021). Em 2010, em uma cidade no norte da Espanha, foi desenvolvido um sistema que coleta dados de sensores com o intuito de gerenciar o número de vagas disponíveis na cidade. O SmartSantander cresceu, e mais de 12 mil sensores foram implantados em outras áreas, como estacionamentos,

caminhões de lixo e táxis. A Figura 6 mostra uma das interfaces da plataforma, com marcadores espalhados pela cidade.

FIGURA 6: APLICATIVO SMARTSANTANDER



FONTE: SANTANA E KON, 2021

Os serviços ofertados por este aplicativo incluem gerenciamento de tráfego, monitoramento ambiental, irrigação de jardins e parques e realidade aumentada. Cada um desses serviços diferentes é integrado por *web services*. Nos sensores altamente acoplados a uma estrutura de IoT, são medidos parâmetros como temperatura, qualidade do ar, do som, iluminação e presença de veículos. Tais sensores conectam-se a repetidores via conexão *wireless*. Os repetidores, por sua vez, são localizados em semáforos e painéis informativos. Eles enviam as informações para os *gateways*, que são serviços Web. Através deles, as informações podem ser armazenadas ou enviadas para centrais de monitoramento (SMART SANTANDER, 2021).

O projeto do SmartSantander foi criado como um experimento inicial no continente europeu para pesquisas e experimentações de tecnologias de cidades inteligentes utilizando IoT. O objetivo principal é alavancar o uso das tecnologias que possam contribuir com a comunidade de pesquisa, testando a arquitetura de uma cidade inteligente em larga escala e em condições do mundo real. (SANCHEZ et al., 2014)

#### 2.2.4 CENTRAL E PORTAL 156

Em Curitiba, a Prefeitura Municipal da cidade, em conjunto com o Instituto das Cidades Inteligentes, mantém desde 1984 a Central 156, uma central telefônica que visa atender os cidadãos do município, prestando serviços, dando informações e servindo de ouvidoria para os mais diversos temas da cidade. (PORTAL 156)

Hoje em dia, a prefeitura disponibiliza também o Portal 156, plataforma *web* e aplicativo móvel que possibilita a realização destes serviços de forma digital e acessível. A plataforma possibilita que os cidadãos obtenham informações úteis sobre a cidade na palma da mão, bem como solicitar serviços e reportar problemas na cidade. (PORTAL 156) A Figura 7 exibe as telas iniciais do aplicativo móvel do Portal 156.

FIGURA 7: APLICATIVO DO PORTAL 156



FONTE: PORTAL 156, 2022

#### 2.2.5 COLAB.RE

Criado no Brasil, o aplicativo do Colab já foi premiado como o melhor aplicativo urbano do mundo pela Organização das Nações Unidas. O propósito do aplicativo é proporcionar a melhoria das cidades, com funcionalidades que permitem aos cidadãos fazer reclamações, sugestões e avaliações de espaços públicos da cidade. A

plataforma possui mais de 100 mil usuários cadastrados e já possui adesão por diversas prefeituras de grandes cidades do país, como Aracaju/SE, Campinas/SP e Recife/PE. (PORTAL BOA VONTADE, 2016).

Como pontos negativos, a plataforma possui reclamações no portal ReclameAqui, voltadas ao baixo índice de atendimento dos protocolos em algumas prefeituras, ponto este que não é diretamente relacionado ao aplicativo, mas que poderia ser sanado com medidas da plataforma que pudessem incentivar e cobrar as prefeituras a atenderem com mais rapidez as ocorrências. Outro ponto negativo é a não existência do sistema para uma versão web, sendo a plataforma apenas presente para aplicativos móveis. O Portal Cidadão, plataforma proposta neste projeto, é um sistema multiplataforma presente na web e também para aplicativos móveis, tendo vantagem neste aspecto.

### 3 MATERIAIS E MÉTODOS

O método de gerenciamento do projeto de *software* utilizado pela equipe no projeto foi o *Scrum*, um método de desenvolvimento ágil bastante popular, que separa as tarefas em ciclos denominados *Sprints*.

#### 3.1 MÉTODOS DE DESENVOLVIMENTO DE SOFTWARE E SCRUM

Nos meados de 1990, surgiram diversos paradigmas com o objetivo de diminuir o trabalho excessivo no desenvolvimento de *software*. Inicialmente conhecidos como modelos leves, passaram a ser chamados de métodos ágeis após a publicação do Manifesto Ágil em 2001. Além do *Scrum*, outros métodos populares que surgiram foram o *Extreme Programming (XP)*, *Feature Driven Development* e *Crystal Clear*.

Os métodos ágeis possuem alguns fatores em comum, como o foco em indivíduos e interações mais do que nos processos e ferramentas, adaptabilidade ao invés de um plano fixo de execução, e *software* ao invés de documentação (SOARES, 2021). Em relação aos métodos tradicionais de desenvolvimento, esses fatores estão mais alinhados com as expectativas dos clientes atuais em relação a entrega de *software*.

O *Scrum* é uma metodologia ágil de gerenciamento de projetos que consiste em uma abordagem dinâmica e objetiva para atingir melhores resultados nos projetos, utilizando menos recursos para isso. É um método inovador que busca simplificar o trabalho das pessoas, visando preservar o maior bem da humanidade moderna: o tempo (SUTHERLAND, 2019).

Essa metodologia se utiliza de diversas práticas para atingir seus objetivos, como por exemplo a divisão do projeto em ciclos de tarefas, chamados de *Sprints*, reuniões frequentes para alinhar o trabalho de cada membro da equipe e discutir formas de melhorar o que está sendo feito, sempre focando nos resultados e na produtividade. A Figura 7 apresenta um exemplo do quadro *Scrum*. A ideia é que todos os membros da equipe possam enxergar o estado atual do projeto, sabendo o que cada um está fazendo, quem está impedido ou com alguma dificuldade, assim aumentando a organização de toda a equipe, pois cada um sabe o que precisa entregar, bem como as entregas dos outros integrantes e suas dificuldades.



FIGURA 7: O QUADRO SCRUM

PROJETO/EQUIPE: EQUIPE SCRUM MARAVILHOSA					
	Backlog	A fazer	Fazendo	Em revisão/perguntas e respostas	Feito!
História do usuário 1					
História do usuário 2					
História do usuário 3					
História do usuário 4					

FONTE: SUTHERLAND (2019)

Pela familiaridade que os membros da equipe possuem com o *Scrum*, ficou definido que esta metodologia seria utilizada no projeto. Ao longo do semestre, os itens do *Scrum* foram utilizados pela equipe da seguinte forma:

- *Sprints* semanais: com uma definição de atividades a serem realizadas a cada sprint;
- Reunião de planejamento da *Sprint*: realizada com o professor orientador do projeto, o qual era também o *Scrum Master* do projeto. Nesta reunião era discutido o que foi feito durante o período anterior e planejado o que seria desenvolvido durante a próxima *Sprint*.
- Utilização da ferramenta Trello, um gerenciador virtual que possibilita a divisão de itens em colunas, atribuição de tarefas a membros da equipe, entre outros, como mostrado na Figura 8.

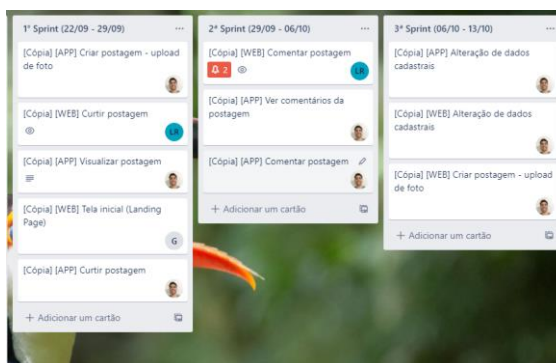


FIGURA 8: INTERFACE DO TRELLO

## 3.2 FERRAMENTAS DE MODELAGEM

Durante a fase de análise do projeto de *software*, é importante utilizar ferramentas que auxiliem na precisão da definição do escopo do projeto. Assim, se faz essencial explicitar os requisitos do sistema, com o objetivo de facilitar o entendimento da equipe a respeito do sistema. No projeto atual, a equipe fez uso da linguagem UML e da ferramenta de modelagem de diagramas ASTAH.

### 3.2.1 UML

A UML, do inglês *Unified Modeling Language*, fornece recursos e especificações para auxiliar no levantamento de requisitos de um projeto de *software* (GUEDES, 2004). Possui forte conexão com os conceitos de Orientação a objetos, fato que tornou a linguagem amplamente usada. Ela disponibiliza por meio de regras visuais, diagramas e conceitos, a possibilidade de documentar visualmente um sistema de forma altamente precisa.

Os diagramas UML auxiliam na criação de uma visão clara do sistema, e evitam erros na especificação de requisitos do mesmo. São geralmente usados em projetos onde é preciso uma documentação clara do funcionamento do *software*.

### 3.2.2 ASTAH

A equipe fez uso da ferramenta ASTAH para a modelagem de diagramas UML. Desenvolvida no Japão utilizando Java, a ferramenta possibilita a criação de diversos diagramas de UML. Ela possibilita inclusive a criação de diagramas a partir de códigos pré-existentes, caso forem classes Java.

Na versão gratuita da ferramenta, é possível criar diagramas de classe, diagramas de caso de uso, diagramas de máquina de estado, diagramas de atividades, diagramas de sequência, diagramas de comunicação, diagramas de *deploy* e diagramas de estrutura.

### 3.3 FERRAMENTAS DE DESENVOLVIMENTO

Durante o processo de desenvolvimento de um *software*, existem algumas ferramentas amplamente utilizadas para facilitar a codificação, integração, modelagem e organização da metodologia do projeto. Um exemplo é o uso de IDEs (do inglês, *Integrated Development Environments*), que combinam ferramentas relevantes para a manutenção do código do projeto em uma interface gráfica integrada. Neste projeto foram utilizadas as IDEs Visual Studio Code, para o aplicativo *Web* e para a API; e Visual Studio, para o aplicativo *mobile*. Também foram utilizados os frameworks .NET, React e Angular, já descritos anteriormente, além de ferramentas de apoio como Trello, para administração de *sprints*, e Git, para controle de versionamento de código.

#### 3.3.1 VISUAL STUDIO CODE

O Visual Studio Code é um editor de código desenvolvido pela Microsoft, com a capacidade de depurar aplicações web. Possui suporte a diversas extensões que acrescentam funcionalidades na ferramenta, com o objetivo de facilitar a codificação do projeto e integração com outros sistemas. Também possui versionamento Git nativo, além de ferramentas para complementação automática do código.

#### 3.3.2 VISUAL STUDIO

O Visual Studio é um ambiente integrado feito pela Microsoft, utilizado principalmente para a criação de software utilizando o framework .NET, e as linguagens C, C#, C++ e F#. Também é bastante utilizado para desenvolvimento web, com o uso da plataforma ASP.NET. Está disponível gratuitamente na versão *Community*, ou nas versões pagas *Professional* e *Enterprise*.

### 3.3.3 GIT

O Git é um sistema de controle de versão de código, que permite o desenvolvimento de um projeto por mais de uma pessoa em paralelo. Utiliza o conceito de “*branches*”, que funcionam como linhas separadas de códigos. Cada desenvolvedor pode trabalhar em uma “*branch*” diferente, por exemplo. O Git possibilita a união do código, além de uma comparação das diferenças entre as linhas de código.

### 3.3.4 TRELLO

O Trello é um aplicativo de gerenciamento de projetos, que oferece uma visão organizada da divisão dos requisitos do mesmo. Foi criado por *Fog Creek* em 2011. A aparência simples e fácil de usar da interface do aplicativo facilita o uso do mesmo, o que a tornou uma das ferramentas mais utilizadas na organização de projetos corporativos.

## 3.4 PLANO DE ATIVIDADES

Durante cada *Sprint* semanal, era apresentado o que tinha sido realizado na *Sprint* anterior e era definido um conjunto de atividades para a próxima semana, conforme detalhado no plano de atividades, que pode ser encontrado na Tabela 1.

TABELA 1 – PLANO DE ATIVIDADES

Sprint	Data de Entrega	Atividades da Sprint
1	10/05/2021	Revisão do TCC da equipe anterior; breve comparação de framework de desenvolvimento para dispositivos móveis; leitura da documentação do TCC 1; definição inicial das funcionalidades do Portal Cidadão.
2	17/05/2021	Evolução do documento de escopo do projeto.

3	24/05/2021	Descrição dos indicadores, gráficos e relatórios na parte de análise de dados.
4	31/05/2021	Início da modelagem de negócios, como diagramas de casos de uso.
5	07/06/2021	Escrita da introdução do documento, evolução da modelagem de negócios.
6	14/06/2021	Pesquisa de indicadores de cadeia de confiança de posts; pesquisa de instituições que irão acessar o Portal Cidadão; escrita de especificação informal de todo o sistema.
7	21/06/2021	Definição do template da página principal, mapa e outros; criação do MER; início do front-end.
8	28/06/2021	Escrita das especificações dos casos de uso, além do DER. Leitura de TCCs anteriores; avanço na documentação.
9	05/07/2021	Desenvolvimento das telas de login, esqueci minha senha e de cadastro do usuário (no front-end). Integração com Google Maps.
10	12/07/2021	Migração do sistema para o servidor AWS. Criado usuário no MySQL para uso no projeto. Leitura de materiais sobre cidades inteligentes e avanço na documentação do projeto.
11	29/09/2021	Continuação do desenvolvimento do projeto, com as seguintes funcionalidades: upload de foto (no aplicativo); curtir postagem (na web); visualizar postagem (no aplicativo); tela inicial (na web); curtir postagem (no aplicativo).
12	06/10/2021	Desenvolvimento do projeto, com as seguintes funcionalidades: comentar postagem (na web); visualizar comentários da postagem (no aplicativo); comentar postagem (no aplicativo).
13	13/10/2021	Desenvolvimento do projeto, com as seguintes funcionalidades: alteração de dados cadastrais (no aplicativo e na web); upload de foto (na web).
14	20/10/2021	Desenvolvimento do projeto, com as seguintes funcionalidades: filtro por categoria e subcategoria (na web); implementação de perfis (na web).
15	27/10/2021	Apresentação do sistema atualizado. Desenvolvimento das seguintes funcionalidades: implementação de perfis (no aplicativo); resolução de post pelo perfil especial (no aplicativo e na web); filtro por categoria e subcategoria (no aplicativo).
16	03/11/2021	Resolução de bugs.
17	10/11/2021	Desenvolvimento das seguintes funcionalidades: redefinição de senha (no aplicativo e na web);

		moderação de posts pelo administrador (no aplicativo e na web).
18	17/11/2021	Desenvolvimento das seguintes funcionalidades: criação de dashboards de categorias, incidentes em aberto (na web).
19	24/11/2021	Desenvolvimento das seguintes funcionalidades: criação de dashboards de segurança, postagens por bairro e incidentes em atraso (na web).
20	01/12/2021	Desenvolvimento das seguintes funcionalidades: implementação de regras de pontuação nos posts (na API); alterar tamanho do marcador no mapa de acordo com a pontuação do post (na web e no aplicativo).
21	08/12/2021	Desenvolvimento das seguintes funcionalidades: filtro por confiabilidade (na web e no app); mostrar confiabilidade na modal de visualizar postagem (na web).
22	15/12/2021	Testes e correções da plataforma; finalização da documentação.

### 3.5 RESPONSABILIDADES

No decorrer do projeto, as tarefas necessárias para o desenvolvimento do projeto foram divididas entre os membros da equipe conforme mostra a Tabela 2.

TABELA 2 – RESPONSABILIDADES DA EQUIPE

<b>Membro da equipe</b>	<b>Tarefas</b>
Gustavo Saporiti	Desenvolvimento do sistema web, desenvolvimento back-end e documentação teórica.
Lucas Ribeiro	Desenvolvimento do sistema web, desenvolvimento back-end, validações e testes.
Samuel Almeida	Desenvolvimento do aplicativo móvel, desenvolvimento back-end, publicação no servidor.
Alessandro Brawerman	Professor orientador e Scrum Master do projeto

### 3.6 TESTES DE SOFTWARE

No projeto do Portal Cidadão a equipe realizou testes de validação, de forma a testar todas as funcionalidades do sistema sem levar em consideração o seu código

fonte, técnica conhecida também como “caixa preta” Foram realizados testes simulando diferentes usuários e perfis acessando o sistema simultaneamente, de forma a validar o comportamento do sistema em uma situação real de uso. Todas as funcionalidades passaram por este processo após o seu desenvolvimento e sua publicação em ambiente de produção na nuvem.

## 4 APRESENTAÇÃO DO PORTAL CIDADÃO

Visando a aproximação da comunicação entre o cidadão e órgãos administrativos, o Portal Cidadão foi criado para suprir a demanda por canais de comunicação eficientes entre ambas as partes.

A plataforma proposta possui um portal *Web*, em que qualquer usuário pode visualizar em tempo real, no mapa da cidade, as reclamações, sugestões e elogios em qualquer ponto da cidade. Cidadãos cadastrados podem criar suas próprias postagens e interagir com outras postagens, contribuindo para uma rede colaborativa de melhorias na cidade. Usuários administradores podem gerenciar postagens para evitar abusos do sistema, enquanto usuários de órgãos públicos ganham um painel de controle que auxilia a visualização de ocorrências, para repassá-las aos departamentos responsáveis e resolvê-las no sistema, dando um rápido e eficiente retorno ao contribuinte. Já no aplicativo móvel, o cidadão possui ainda mais praticidade para efetuar as contribuições para a cidade, sendo utilizados os periféricos de seu telefone celular como câmera e GPS de alta precisão.

### 4.1 ARQUITETURA DO SISTEMA

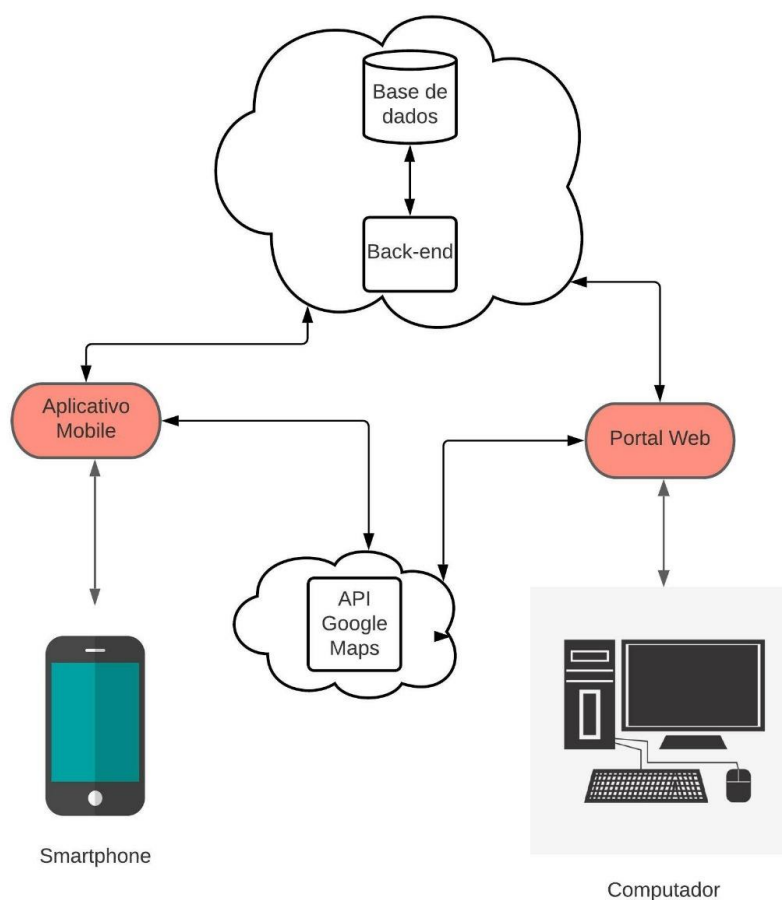
Em termos gerais, o Portal Cidadão funciona da seguinte forma: a partir de um aplicativo para smartphones, desenvolvido com o *framework* React Native, o usuário cria uma postagem (que inclui a descrição do problema, categoria, foto e localização, esta última capturada a partir do GPS do celular), que é enviada para o *back-end* do sistema - hospedado em algum servidor na nuvem, atualmente em servidor da UFPR - onde será tratada e armazenada no banco de dados MySQL. O usuário também pode criar a postagem a partir do sistema na *Web*, que foi desenvolvido utilizando o *framework* .NET 5 no *back-end*, e o *framework* Angular 12 no *front-end*. O *front-end* faz integração com a API do Google *Maps* para obter a geolocalização do usuário.

A Figura 9 apresenta o sistema como um todo, com suas tecnologias específicas visualmente representadas. Do lado inferior esquerdo está representado o dispositivo móvel, pelo qual o usuário acessa o aplicativo para *smartphone*. Do lado



direito está representado o computador, pelo qual o usuário acessa o site na *Web*. Tanto o aplicativo para smartphones quanto o site da *Web* integram-se ao *back-end* do projeto, representado na figura dentro da nuvem. O *back-end* conecta-se com o banco de dados. Os *front-ends* conectam-se e com a API do Google Maps, representado na Figura 9, também dentro da nuvem.

FIGURA 9: ARQUITETURA DO SISTEMA



FONTE: OS AUTORES, 2021

## 4.2 FUNCIONALIDADES

No Portal Cidadão foram propostas ferramentas que permitem ao usuário comunicar-se com órgãos administrativos da cidade por meio de posts, seja para fazer uma reclamação, um elogio, ou comentar a postagem de outro usuário. Abaixo, segue a listagem de funções de cada parte da plataforma.

- Portal Web
  - *Landing Page* (site institucional): nesta página a plataforma é apresentada, com descrição de suas funcionalidades e informações que atraíam o interesse do usuário para o acesso na ferramenta;
  - Login e redefinição de senha: os usuários podem efetuar *login* com seus dados, bem como redefinir a senha já existente em caso de perda ou esquecimento;
  - Cadastro de usuário: a plataforma proposta disponibiliza o acesso de forma anônima à ferramenta, porém para algumas funcionalidades é requerido o cadastro do usuário, que deverá inserir seus dados básicos para identificação na ferramenta;
  - Visualização de posts no mapa: as postagens são exibidas ao usuário de acordo com a sua geolocalização e a sua pontuação, de forma que o cidadão pode rapidamente visualizar as ocorrências próximas ao seu local;
  - Criação de posts / CRUD: o usuário pode criar sua postagem, informando os dados necessários para a inserção da mesma no mapa, que pode ser visualizada por outros cidadãos utilizando a ferramenta;
  - Dashboard (para usuário especial): para usuários com perfil especial, como funcionários de uma prefeitura, e para administradores, a plataforma disponibilizará um dashboard contendo informações importantes sobre o estado atual da cidade, auxiliando na tomada de decisão por parte do poder público responsável;
  - Moderação de posts (para administrador): os usuários com perfil administrador, como gerenciadores do sistema, possuem acesso a funcionalidade de editar ou apagar uma postagem, a fim de restringir o uso indevido da ferramenta, como postagens contendo *spam*, ofensas e conteúdos indevidos

- Sistema para Dispositivos Móveis
  - *Login* e redefinição de senha: da mesma forma presente no sistema web, os usuários podem se autenticar no sistema, bem como alterar sua senha para casos de perda ou esquecimento;
  - Cadastro de usuário: a plataforma exige o cadastro para que o usuário possa utilizar funcionalidades específicas, como criação de postagens e interações, como comentários e curtidas;
  - Visualização de posts no mapa: os cidadãos podem visualizar as postagens de outros cidadãos no mapa, de forma que o aplicativo móvel exibe as postagens mais próximas à localização obtida no aparelho do usuário;
  - Criação de posts / *CRUD*: o usuário pode criar sua postagem, informando os dados necessários para a inserção da mesma no mapa, que pode ser visualizada por outros cidadãos utilizando a ferramenta;

### 4.3 PERFIS DO SISTEMA

O sistema possui três perfis de acesso, cada um com suas respectivas restrições:

- Perfil de usuário comum: definido caso o usuário faça login com sucesso na tela exibida na Figura 8. Nesse caso, será permitido ao usuário a criação de novas postagens; o botão para inserção de nova postagem será exibido na tela principal. O usuário poderá abrir uma postagem exibida no mapa e fazer comentários, ou curtir a postagem.
- Perfil de administrador: ao realizar *login* pela página exibida na Figura 8, o sistema irá consultar através do banco de dados o tipo de usuário inserido. Caso seja do tipo administrador, o usuário não poderá realizar postagem no sistema; o botão para inserção de nova postagem não será exibido na tela principal. O administrador poderá visualizar e abrir as postagens exibidas no mapa, e conseguirá excluir postagens e comentários do sistema. Esse perfil também dá acesso a uma página com indicadores de dados, descritos na seção 4.4.

- Perfil especial: ao realizar login pela página exibida na Figura 8, o sistema irá consultar através do banco de dados o tipo de usuário inserido. Caso seja do tipo especial, o usuário não poderá realizar postagem no sistema; o botão para inserção de nova postagem não será exibido na tela principal. O usuário poderá visualizar e abrir as postagens exibidas no mapa, e conseguirá marcar a postagem como 'Aprovada'. Esse perfil também dá acesso a uma página com indicadores de dados, descritos na seção 4.4.
- Perfil para órgãos (futuro): como ideia futura do projeto, o sistema poderá ter um perfil específico para órgãos das prefeituras, como por exemplo subsecretarias ou companhias de setores específicos como água, energia, etc. Com isso, o perfil do órgão poderia ver apenas as postagens que estão relacionadas nas categorias do seu setor, e também, posteriormente, a prefeitura poderia fazer uso dos dados categorizados por órgãos para analisar os setores com melhor eficiência na resolução de incidentes, bem como aqueles com menor eficiência e que precisam de ações para melhorias.

#### 4.4 INDICADORES DE DADOS

Ao realizar login com um usuário do tipo 'Especial', o usuário possui acesso a uma página com indicadores sobre os dados do sistema. O objetivo dessa página é fornecer aos órgãos administrativos um painel de controle com visão panorâmica sobre itens considerados importantes dentro do sistema, o que facilitaria a tomada de decisões estratégicas e direcionaria o foco dos órgãos para uma resolução de problemas na cidade de maneira mais eficiente.

Estudos recentes mostram que o crescente número de dispositivos que fazem uso da Internet das Coisas (IoT) no ambiente urbano têm potencial para mudar vários processos que acontecem hoje nas cidades (WOODS, 2019). Desenvolvedores que trabalham em áreas como *Big Data*, *Machine Learning* e Inteligência Artificial estão

cada vez mais cooperando com líderes e administradores urbanos com ferramentas bastante eficazes para a resolução de problemas.


Através da análise de dados, é possível extrair dados de órgãos públicos e transformá-los em informações úteis para os líderes da cidade. Como consequência, podem surgir soluções eficientes para diversos problemas, o que geraria maior confiança nos gestores das cidades, e um maior engajamento dos cidadãos no meio urbano (OBESO, 2016).

Em um primeiro momento, os dados analisados no Portal Cidadão são os seguintes:

- Número de incidentes abertos
- Número de incidentes em atraso (com mais de 60 dias sem resolução)
- Bairros com mais incidentes abertos
- Maior número de horários de ocorrências do tipo “Segurança” por bairro
- Categorias com mais incidentes abertos
- Categorias com maior número de incidentes, desde o lançamento do sistema

#### 4.5 VISÃO GERAL - SISTEMA WEB

Após acessar a página inicial do portal (*landing page*), fica disponível para o usuário a página de *login*, mostrada na Figura 10.



PortalCidadão

Faça login para acessar a ferramenta

E-mail ou CPF

Senha

Lembrar de mim

Novo cadastro

Login

FIGURA 10: PÁGINA DE LOGIN

Caso pressione o botão “Novo Cadastro”, o sistema redireciona o usuário para a página de criação de usuário, conforme a Figura 11.



PortalCidadão

Digite os dados abaixo para se registrar no sistema

Nome Completo

CPF

Email

Senha

Salvar

FIGURA 11: PÁGINA DE NOVO USUÁRIO

Após o preenchimento dos dados, o sistema faz uma validação no banco de dados e, caso o usuário seja validado, faz a inserção do usuário no banco de dados, retornando automaticamente para a página de login.

Ao fazer *login* como usuário corretamente, é exibida a página principal, mostrada na Figura 12.

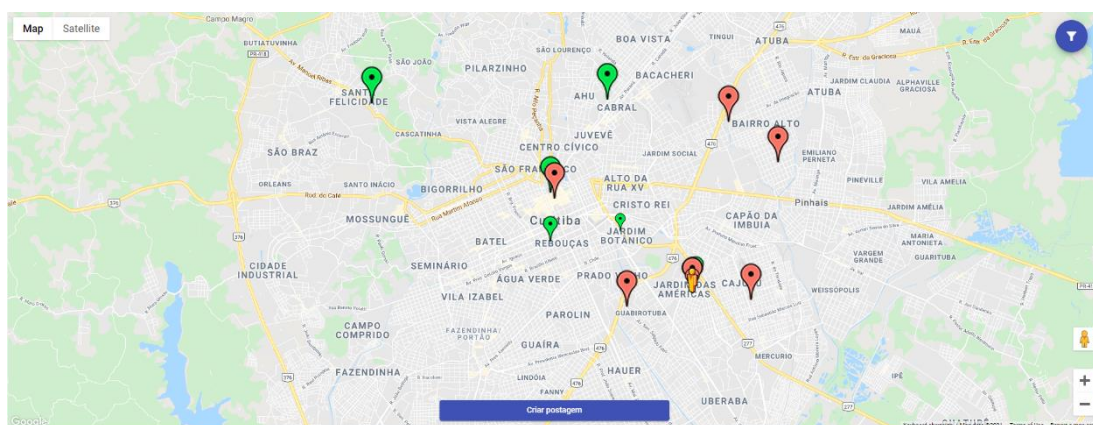


FIGURA 12: PÁGINA PRINCIPAL

Essa tela mostra os posts, exibidos pelos marcadores no mapa num raio que pode ser configurado pelo usuário. Os marcadores são coloridos de forma a representar o tipo (subcategoria) da postagem: vermelho para reclamações, verde

para elogios e amarelo para sugestões. É possível filtrar por bairro, para navegar até um bairro específico, ou filtrar por evento; ao pressionar o botão com ícone de filtro, será aberta uma janela para selecionar o tipo de filtro, conforme a Figura 13.



FIGURA 13 – PÁGINA DE FILTROS

Ao pressionar o botão “Criar Postagem” a partir da página principal, o usuário pode inserir um novo post, conforme a Figura 14.

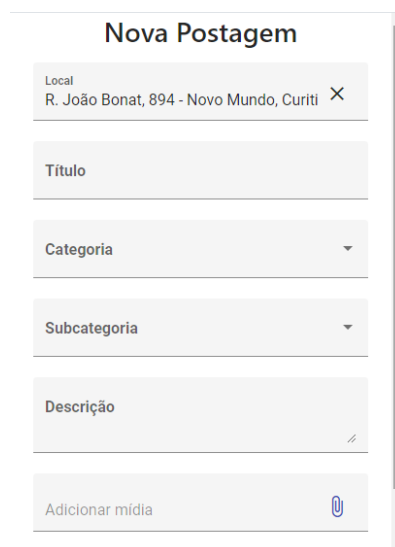


FIGURA 14 – PÁGINA DE NOVA POSTAGEM

No Portal Web, o usuário deve digitar a localização do post obrigatoriamente, visto que na Web não há como obter a localização de forma tão precisa como em um dispositivo móvel.

Depois de preenchido o campo “Localização”, o usuário deve especificar a categoria e subcategoria da postagem. Cada categoria possui uma descrição e um órgão responsável associado (ex: Iluminação – COPEL). Os órgãos não possuem acesso ao sistema, a prefeitura fica responsável de repassar a solicitação ao órgão e

acompanhar o caso até marcá-lo como resolvido. As categorias disponíveis são espaços públicos, iluminação, trânsito/pavimentação, calçadas, acessibilidade, transporte público, meio ambiente, eventos, vandalismo, pessoa em situação de rua, zoonoses e segurança. Em cada categoria, é obrigatório selecionar uma subcategoria: reclamação, elogio ou sugestão.

Após preencher os campos, o sistema faz a inserção do post no banco de dados, e o post fica disponível no mapa da tela principal.

Voltando para a tela principal, é possível ao usuário abrir a postagem de outros usuários, podendo deixar seu comentário ou curtida, conforme a Figura 15.

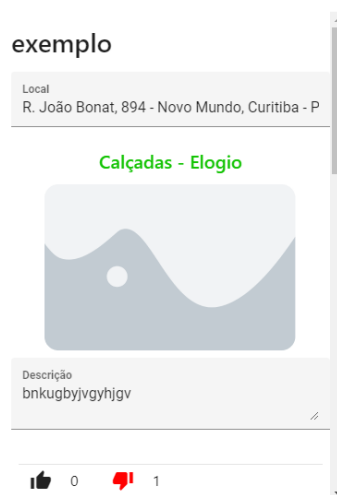


FIGURA 15: PÁGINA DE EXIBIÇÃO DE POSTAGEM

Caso o usuário seja perfil especial, é possível acessar a página “*Dashboard*”, onde dados relevantes sobre os problemas da cidade são exibidos em forma de gráfico. Esses dados podem ser usados pelos órgãos públicos como ferramentas poderosas de gestão urbana. A Figura 16 mostra os um gráfico que exhibe bairros com maior número de ocorrência.



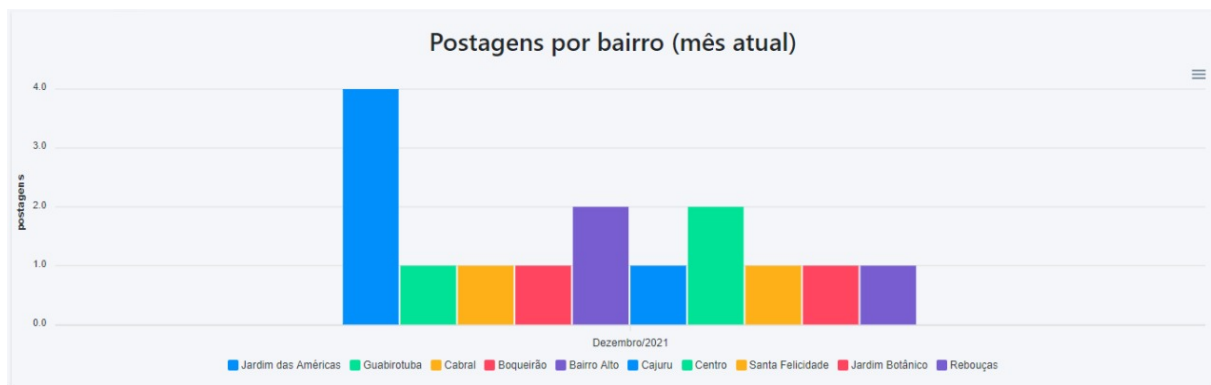


FIGURA 16: DASHBOARD 1

Na mesma página, é exibido o gráfico ilustrado na Figura 17, que mostra o número de incidentes abertos no sistema, por mês.



FIGURA 17: DASHBOARD 2

Também é mostrado um gráfico exibindo a quantidade de eventos por categoria, como mostra a Figura 18.



FIGURA 18 – DASHBOARD 3

Ainda na mesma página, outro gráfico disponível exibe as postagens com mais de 15 dias sem resolução, como mostra a Figura 19.



FIGURA 19 – DASHBOARD 4

A página também mostra os horários que o sistema mais registrou ocorrências, como exibe a Figura 20.



FIGURA 20 – DASHBOARD 5

A partir da página inicial, pressionando o botão “Minha Conta”, é possível acessar uma janela de alteração de dados do usuário, conforma a Figura 21.

091.614.849-11

Gustavo

Administrador

gustavo@gmail.com

Salvar dados

Redefinir senha

Voltar

FIGURA 21 – ALTERAÇÃO DE DADOS (WEB)

#### 4.6 VISÃO GERAL – APLICATIVO PARA DISPOSITIVOS MÓVEIS

Assim como no Portal *Web*, ao acessar o aplicativo para dispositivos móveis, o usuário é levado para a tela de login, apresentada pela Figura 22.



FIGURA 22: PÁGINA DE LOGIN PARA DISPOSITIVOS MÓVEIS

Caso pressione o botão “Entrar como visitante”, o sistema é exibido no modo de leitura, ignorando a etapa de login. O usuário pode então apenas visualizar os posts, sem comentar ou avaliar.

Caso pressione o botão “Novo Usuário”, o sistema leva o usuário para a página de criação de usuário, conforme a Figura 23.



FIGURA 23: PÁGINA NOVO USUÁRIO EM DISPOSITIVOS MÓVEIS





FIGURA 25: PÁGINA DE DETALHES PARA DISPOSITIVOS MÓVEIS

Pressionando o botão de filtro a partir da página principal, é exibida uma *modal* com possíveis opções para filtrar o mapa, conforme a Figura 26.

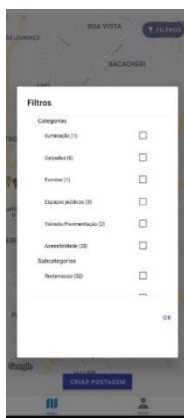


FIGURA 26: FILTROS EM DISPOSITIVOS MÓVEIS

Pressionando o botão “Criar Postagem” a partir da página principal, o usuário é levado a uma janela para criação de postagem, conforme Figura 27.



FIGURA 27: NOVA POSTAGEM EM DISPOSITIVOS MÓVEIS

Pressionando o botão “Inserir Mídia” a partir da janela de criação de postagem, o usuário é levado a uma nova janela para inserção de arquivos, conforme a Figura 28.

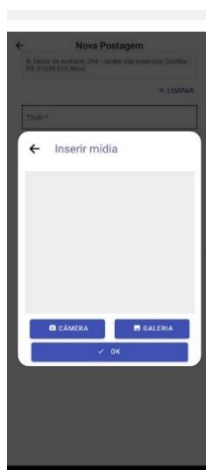


FIGURA 28: INSERIR MÍDIA EM DISPOSITIVOS MÓVEIS

É possível também alterar o cadastro de um usuário existente, como mostra a Figura 29.

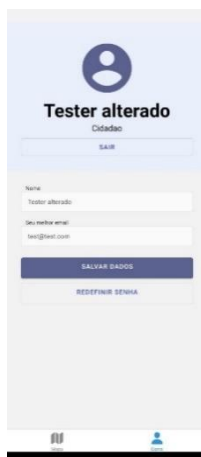


FIGURA 29: ALTERAR USUÁRIO EM DISPOSITIVOS MÓVEIS

## 4.7 SOBRE A DOCUMENTAÇÃO

Nos anexos do presente documento, estão disponíveis diagramas de negócio, cada um com um objetivo específico. São explicados neste capítulo cada um deles, assim como sua finalidade e importância para o entendimento do sistema.

### 4.7.1 REQUISITOS FUNCIONAIS

A seção de requisitos funcionais especifica todas as funcionalidades exigidas para que o *software* atenda às necessidades do sistema. Trata-se de uma descrição breve de cada uma das funcionalidades, que serão mais exploradas em outros diagramas (POHL et al., 2015). Os requisitos funcionais são apresentados no anexo A.

### 4.7.2 DIAGRAMA DE CASO DE USO

A Linguagem de Modelagem Unificada (UML) é a linguagem padrão para modelagem visual de software e de sistemas. A modelagem ajuda a ter um vislumbre panorâmico das funcionalidades do sistema, permitindo a comunicação, foco, documentação de aspectos importantes do mesmo (MILES, 2006).



Um dos principais diagramas usados na UML é o de caso de uso, que tem por objetivo fazer uma descrição visual do que o sistema deve fazer. Os atores, ou pessoas que acessam o sistema, são representadas no documento relacionadas às ações que serão realizadas pelas mesmas. As funcionalidades são descritas no documento visualmente de forma simplificada. O diagrama de caso de uso deste sistema é apresentado no Apêndice B.

#### **4.7.3 ESPECIFICAÇÃO DOS CASOS DE USO**

A especificação do caso de uso é uma descrição textual dos requisitos do sistema. Alguns conceitos envolvidos são diferentes do diagrama de caso de uso, que é uma representação visual dos requisitos do sistema. Aqui, também são abordados os atores envolvidos na ação específica descrita, mas além disso estão presentes itens como fluxo principal, que descreve o cenário mais frequente pelo qual o ator passará ao executar a ação; fluxo alternativo, que descreve um caminho diferente do principal pelo qual o ator irá passar; fluxo de exceção, que descreve um caminho pelo qual ocorre algum impedimento para o fluxo normal da ação no sistema; e regras de negócio, que descrevem requisitos para que a ação possa ocorrer no sistema (BEZERRA, 2015). Os casos de uso deste sistema são apresentados no Apêndice C.

#### **4.7.4 DIAGRAMA DE CLASSES**

O diagrama de classes representa a relação entre as classes do sistema, que servirão de modelo para a criação dos objetos no *software* real. Cada classe é representada por seu nome, seus atributos, e seus métodos. Na prática, a representação das classes em um diagrama auxilia também na criação do banco de dados, onde cada classe será uma tabela (GUDWIN, 2015). O diagrama de classes é apresentado no Apêndice D.

#### **4.7.5 DIAGRAMA ENTIDADE-RELACIONAMENTO (DER)**

O diagrama entidade-relacionamento representa visualmente as entidades envolvidas no sistema, juntamente com seus atributos e relacionamentos. Em geral, a UML segue um formato generalizado para esses diagramas, que especifica as entidades como retângulos, com seus respectivos atributos, e os relacionamentos como losangos, ligados às entidades por linhas (SILVA, 2001). O DER está representado no Apêndice E.

#### **4.7.6 DIAGRAMA DE SEQUÊNCIA**

Os diagramas de sequência modelam, em um único diagrama, interações entre entidades do sistema para a realização de uma ação, exibindo a ordem seguida para que o caso de uso específico seja executado – a sequência seguida para a ação ser executada (GUEDES, 2009).

O diagrama começa no topo, e a sequência de ações a seguir são mostradas abaixo de forma gradual. Na parte superior da página, os diversos objetos que interagem no cenário são organizados. Abaixo, as mensagens e interações entre os objetos são exibidas em linhas. Os diagramas de sequência são apresentados no Apêndice F.

#### **4.7.7 DIAGRAMA DE GANTT**

O diagrama de Gantt é uma ferramenta visual para o controle do cronograma de um projeto, contendo prazos de entrega, precedência de tarefas, duração da tarefa e responsáveis. Contém a lista de atividades a serem realizadas e o tempo de início de término de cada uma, de forma sequencial (FOWLER, 2005). O Diagrama de Gantt produzido foi dividido em duas partes, uma representando a primeira parte do projeto, no Apêndice G, e outra representando a segunda parte, no Apêndice H.

#### 4.7.8 DOCUMENTAÇÃO DA API

A documentação da API apresenta a maneira de realizar requisições (*requests*) a API implementada. Ela fornece os métodos a serem chamados, os parâmetros esperados e aceitos pela API, tanto na própria *URL* da requisição, quanto no corpo da mesma. Além disto, a documentação da API também informa quais os possíveis retornos dos métodos em respostas à uma requisição e quais os possíveis erros que podem acontecer. Por fim, mas não menos importante, é apresentada também a rota para cada requisição possível. A documentação da API do Portal Cidadão pode ser encontrada no Apêndice I.

## 5 CONSIDERAÇÕES FINAIS

O Portal Cidadão foi concebido com a ideia de facilitar a comunicação entre os cidadãos e os órgãos administrativos da cidade. A dificuldade de comunicar um problema para a prefeitura rapidamente foi uma das motivações para o desenvolvimento do sistema.

Outro ponto de destaque que o Portal Cidadão pode ser útil é a presença de indicadores de dados sobre os problemas postados no Portal. Tanto para os órgãos administrativos quanto para os usuários, essas informações podem servir para uma melhor administração da cidade. O material abundante sobre os conceitos das cidades inteligentes, além de outros aplicativos semelhantes que já foram usados em outras cidades facilitou a construção um sistema simples e eficiente, que pode auxiliar o problema da dificuldade de comunicação entre o cidadão e a administração da cidade, quando este encontra um problema urbano.

A primeira parte do desenvolvimento do projeto teve grande foco na documentação e na modelagem do escopo do projeto. Foram realizados o levantamento de requisitos do sistema, assim como as especificações do caso de uso, diagrama de classes, diagrama entidade-relacionamento, diagramas de sequência, diagrama de Gantt dessa parte do desenvolvimento e protótipos das telas do sistema. Também foi discutida a arquitetura do projeto e questões como linguagens e *frameworks* utilizados em cada parte do sistema permearam grande parte do desenvolvimento. Uma versão inicial do sistema foi desenvolvida tanto para dispositivos móveis quanto para a web.

Na segunda parte do projeto foi feita a implementação de todos os requisitos definidos anteriormente, bem como eventuais correções na arquitetura e modelagem do mesmo.

Toda a construção do projeto foi realizada com a tecnologias atuais do mercado, como o React Native, na parte do aplicativo para *smartphones*, ou a integração com a API do Google Maps, presente tanto no aplicativo para dispositivos móveis quanto no portal *web*.

Para trabalhos futuros, o sistema pode adquirir novas funcionalidades, como a implementação de novas situações para uma postagem, como por exemplo: não visualizada, em atendimento, aguardando validação do cidadão, etc. Também pode ser implementada a criação de um perfil voltado para órgãos das cidades, como por

exemplo secretarias da prefeitura ou companhias responsáveis por determinado departamento. Assim, cada postagem só poderá ser atendida pelo responsável específico do determinado departamento da cidade. Com isso, novos indicadores poderão ser implementados, mostrando a eficiência de cada um dos órgãos específicos dentro da cidade, possibilitando a melhoria contínua da prefeitura.

É sabido que pode haver dificuldades de respostas dos órgãos públicos, para motivar estes órgãos pretende-se implementar um ranking mostrando o percentual de problemas resolvidos para cada órgão cadastrado no sistema, uma listagem dos top 5 órgãos que mais resolveram problemas, ranking de interação mostrando os órgãos institucionais que mais responderam usuários e outros rankings que mostrem a eficiência de tais órgãos. Outra possibilidade que pode ser explorada é a mineração e análise de dados através do sistema para propiciar mais indicadores que possam auxiliar a tomada de decisão pelas prefeituras.

Comparando o sistema final com os objetivos esperados, pode-se dizer que o resultado desejado foi obtido, contribuindo imensamente para o conhecimento profissional e acadêmico dos desenvolvedores, bem como pode contribuir com a sociedade através das funcionalidades desenvolvidas no projeto.

## REFERÊNCIAS

**#TendênciasTelefônica: Smart Cities, as cidades do futuro.** 2015. Disponível em:

<https://www.telefonica.com.br/servlet/Satellite?c=Noticia&cid=1386094803043&pageName=InstitucionalVivo%2FNoticia%2FLayoutNoticia01>

Acesso em: 15 de agosto de 2021

ANGULAR. Google, 2010. Site disponível em:

<https://angular.io/>

Acesso em: 15 de agosto de 2021

Martin, C. Robert. *Arquitetura limpa: O guia do artesão para estrutura e design de software.* 2019. Editora Alta Books.

ISBN 8550804606

BADIA, Thomas Diego. Aux Bresil, Curitiba, lex ville modele damerique latine peine a se reinventer. **Blog Le Monde**, 2014. Disponível em:

<http://mondeacplanete.blog.lemonde.fr/2014/03/27/au-%E2%80%8Bbresil-curitiba-lex-ville-modele-damerique-latine-peine-a-se-reinventer/>

Acesso em: 31 de julho de 2021

BEZERRA, Eduardo. **Princípios de Análise e Projeto de Sistemas com UML.** Editora Campus.

ISBN 85-352-1032-6

CCR, **Como a tecnologia melhora a qualidade do sistema de transporte.** Disponível em:

<https://g1.globo.com/especial-publicitario/em-movimento/ccr/noticia/como-a-tecnologia-melhora-a-qualidade-do-sistema-de-transporte.ghtml>

Acesso em: 31 de julho de 2021

CUNHA, Maria Alexandra. **Smart cities: transformação digital de cidades**. São Paulo, 2016. Iniciativa FGV EAESP.

ISBN 978-85-87426-29-1

**Desmistificando REST com Java**; DIAS, Emílio. Disponível em:

<https://s3.amazonaws.com/algaworks-assets/ebooks/algaworks-livreto-desmistificando-rest-com-java-1a-edicao.pdf>

Acesso em: 16 de agosto de 2021.

Dubai Pulse, **Dubai Now – Smart Technology for Civil Engagement**. 2015. Disponível em:

<https://www.dubaipulse.gov.ae/casestudies/detail/dubai-now-smart-technology-for-civil-engagement>

Acesso em: 15 de agosto de 2021

ERIC, Woods. **Data Analytics: The Key to delivering smart cities**. Smart Energy International Magazine. 29 de outubro de 2019.

FOWLER, Martin. **UML Essencial: um breve guia para linguagem-padrão de modelagem de objetos**. Editora Bookman, 2005.

ISBN 9788536304540.

GUDWIN, Ricardo R. **Engenharia de Software: Uma Visão Prática**. 2015. DCA-FEEC-UNICAMP.

GUEDEA, Gilleanes T. A., 2009. **UML 2: Uma Abordagem Prática**. Editora Novatec. 2009.

ISBN 978-85-7522-281-2

HENRIQUE, Theo. Brasil, Curitiba usa aplicativo para resolver problemas da cidade. **Portal A Tarde** 2014. Disponível em:

<https://atarde.uol.com.br/digital/noticias/curitiba-usa-aplicativo-para-resolver-problemas-da-cidade-1601809/>

Acesso em: 31 de julho de 2021

ICI, **Tecnologia melhora gestão dos serviços da saúde municipal em Curitiba.** Disponível em:

<https://www.ici.curitiba.org.br/conteudo/saude-prefeitura-de-curitibapr/155>

Acesso em: 16 de julho de 2021

LIMA, Edwin. **C# e .Net para desenvolvedores.** 2002. Editora Campus Ltda.

ISBN 8535209549

LINDLEY, Cody. **Front-end Developer Handbook.** Disponível em:

<https://frontendmasters.com/books/front-end-handbook/2019/>

Acesso em: 15 de agosto de 2021

KLAUS, Pohl. **Fundamentos da Engenharia de Requisitos.** Um Guia de Estudo para o Exame CPRE-FL Certified Professional for Requirements Engineering – Foundation Level em conformidade com o padrão IREB. Editora T&M.

MAMORÉ, D. S. et al. **Portal Cidadão:** Informações da cidade na palma da mão. Orientador: Alessandro Brawerman. 2013.126f. TCC (Graduação) – Curso de Tecnologia em Análise e Desenvolvimento de Sistemas, SEPT – UFPR, Curitiba, 2013.

NORTHWOOD, Chris. **The Full Stack Developer.** Apress.

ISBN 978-1484241516

OBESO-ORENDAIN, A., Lopez-Neri, E., Donneaud-Bechelani, C. **The role of the Data Scientist within Smart Cities.** IEEE – GDL CCD Smart Cities.

População de Curitiba. Disponível em:

<https://www.ibge.gov.br/cidades-e-estados/pr/curitiba.html>

Acesso em: 31 de julho de 2021



REACT NATIVE. Facebook, Inc. 2021. Site disponível em:

<https://reactnative.dev/>

Acesso em: 15 de agosto de 2021. Revista Use, **Ranking Connected Smart Cities 2020**. Disponível em:

<http://www.revistause.com.br/ranking-connected-smart-cities-2020-classifica-curitiba-como-a-3a-cidade-mais-inteligente-do-pais-e-1a-em-urbanismo>

Acesso em: 31 de julho de 2021

RICHARDSON, Leonard; RUBY, Sam. **RESTful Web Services**. O'Reilly Media. 2007.

ISBN: 978-0596529260

RUSS, Miles. **Learning UML 2.0**. 2006. O'Reilly Media, Inc.

ISBN 9780596009823

SANTANA, Eduardo; KON, Fabio. **Cidades Inteligentes: Conceitos, plataformas e desafios**. Disponível em:

<https://sol.sbc.org.br/livros/index.php/sbc/catalog/download/6/6/17-1?inline=1>

Acesso em: 16 de agosto de 2021.

SILVA, Alberto Manuel Rodrigues da. **UML, Metodologias e Ferramentas CASE**. Edições Centro Atlântico, 2001.

ISBN 972-8426-36-4.

SMART SANTANDER. **SmartSantander Experimental Test Facilities**. Disponível em:

<https://www.smartsantander.eu/index.php/testbeds>.

Acesso em: 16 de agosto de 2021.

SANCHEZ, Luis et al. **SmartSantander: IoT experimentation over a smart city testbed**. Computer Networks, Volume 61. 2014. Disponível em:

<https://www.sciencedirect.com/science/article/pii/S1389128613004337>. Acesso em: 03 jan. 2022.

SOARES, Michel dos Santos. **Metodologias Ágeis Extreme Programming e Scrum para o Desenvolvimento de Software**. Disponível em:

<http://periodicosibepes.org.br/index.php/reinfo/article/viewFile/146/38>

STATISTA. **Cross-platform mobile frameworks used by software developers worldwide from 2019 to 2021**. Disponível em:

<https://www.statista.com/statistics/869224/worldwide-software-developer-working-hours/>.

Acesso em: 16 de agosto de 2021.

PORTAL BOA VONTADE. **Conheça o Colab.re, aplicativo que pode ajudá-lo a melhorar sua cidade**. Disponível em:

<https://www.boavontade.com/pt/tecnologia/conheca-o-colabre-aplicativo-que-pode-ajuda-lo-melhorar-sua-cidade-0>.

Acesso em: 02 de janeiro de 2022.

VIZIA, B. **Fórum Urbano Mundial 2010** - Pensando o urbano a partir do Rio de Janeiro. Disponível em:

[https://www.ipea.gov.br/desafios/index.php?option=com\\_content&view=article&id=1281:reportagens-materias&Itemid=39](https://www.ipea.gov.br/desafios/index.php?option=com_content&view=article&id=1281:reportagens-materias&Itemid=39)

Acesso em: 15 de agosto de 2021

WAM, 22 de julho de 2021. **Novos serviços DHA disponíveis no aplicativo DubaiNow**. Disponível em:

<https://wam.ae/pt/details/1395302946390>

Acesso em: 15 de agosto de 2021

**Web Services**. MORAES, Juliano; BREDÁ, Marcus; GIL, Paulo; MEDAGLIA, Rafael. Disponível em:

[https://www.inf.pucrs.br/~gustavo/disciplinas/sd/material/Artigo\\_WebServices\\_Conceitual.pdf](https://www.inf.pucrs.br/~gustavo/disciplinas/sd/material/Artigo_WebServices_Conceitual.pdf)

Acesso em: 01 de agosto de 2021

World Population Prospects, 2017. Disponível em:

[https://www.un.org/development/desa/pd/sites/www.un.org.development.desa.pd/files/documents/2020/Jan/un\\_2017\\_world\\_population\\_prospects-2017\\_revision\\_databooklet.pdf](https://www.un.org/development/desa/pd/sites/www.un.org.development.desa.pd/files/documents/2020/Jan/un_2017_world_population_prospects-2017_revision_databooklet.pdf)

Acesso em: 31 de julho de 2021

## APÊNDICE A – LISTA DE REQUISITOS

### Requisitos Não Funcionais

#### RNF001 – Multiplataforma

O sistema deverá ser acessível via portal web e também via dispositivos móveis, de sistema operacional Android ou iOS.

#### RNF002 – Segurança

O sistema deverá possuir autenticação utilizando o mecanismo JWT - Json Web Tokens, a fim de garantir a segurança das informações e dos usuários.

### Requisitos Funcionais

#### RF001 - Cadastrar postagem

O sistema deve permitir o cadastro, edição e exclusão de postagens. Uma postagem se refere a um relato em um determinado local, podendo ser categorizada entre: transporte, saúde, educação, lazer e segurança. Toda postagem deverá também ter uma subcategoria: reclamação, elogio ou sugestão. A postagem poderá ser relacionada ao local atual do usuário ou a outro local especificado pelo cidadão.

#### RF002 - Visualizar mapa

O sistema deve mostrar aos usuários o mapa da cidade, com as marcações nos respectivos locais em que as postagens foram realizadas.

#### RF003 - Cadastro de usuário

O sistema deve permitir o cadastro de novos usuários publicamente. Os campos para o cadastro serão: nome completo, CPF, e-mail e senha.

#### RF004 - Confirmar e-mail

O sistema deverá enviar um e-mail de confirmação para o usuário recém-cadastrado, assim evitando possíveis cadastros incorretos ou maliciosos. Ao clicar em um link recebido, o usuário estará apto a acessar as funções do sistema.

**RF005 - Redefinir senha**

O sistema deverá permitir que o usuário redefina sua senha em caso de esquecimento, enviando a ele um e-mail com o link da redefinição. Ao entrar no link o usuário deverá preencher os campos: nova senha e confirmar senha.

**RF006 – Login**

O sistema deverá permitir o login de usuários cadastrados no sistema, bem como o acesso como visitante. O acesso como visitante limita as funcionalidades do usuário no sistema, permitindo apenas a visualização de postagens no mapa.

**RF007 - Página institucional**

O sistema deverá possuir, em sua versão web, uma breve página explicativa do sistema contendo as principais funcionalidades da ferramenta.

**RF008 – Dashboard**

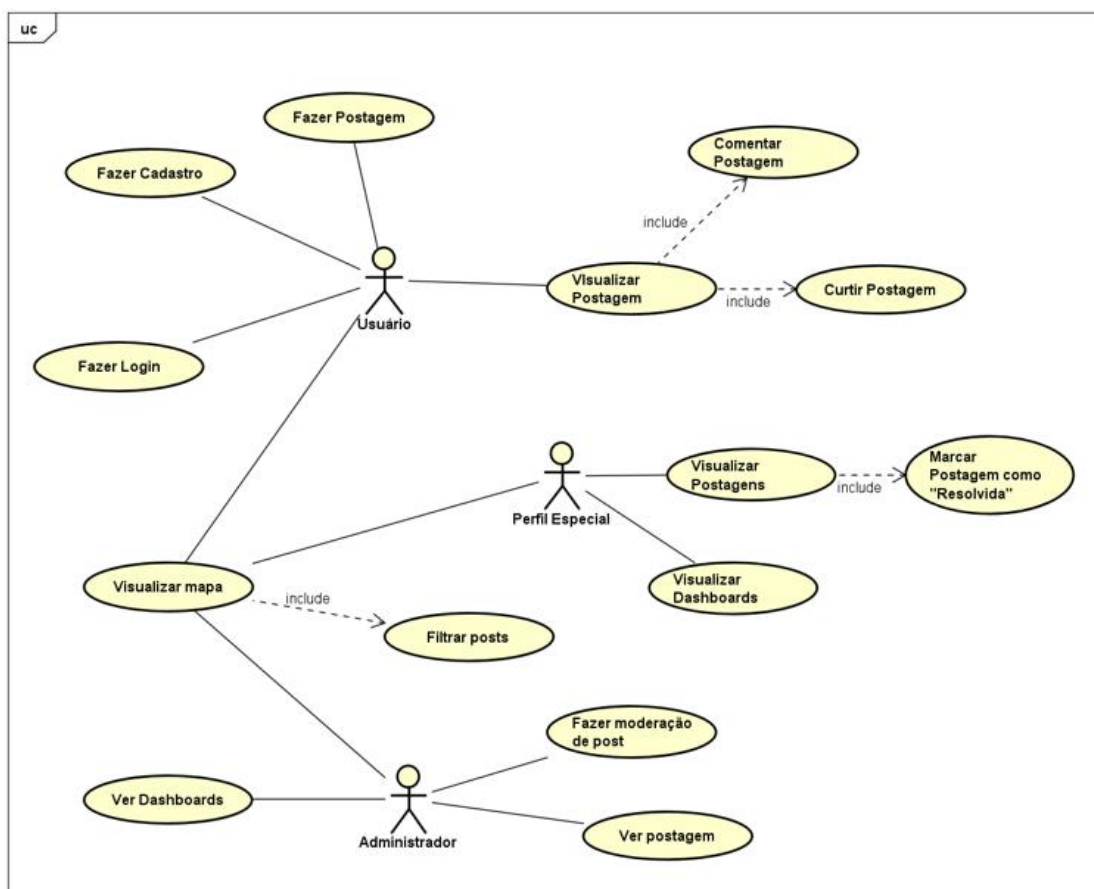
O sistema deverá mostrar, em sua versão web, para os usuários especiais do sistema e administradores, um *dashboard* com as informações mais relevantes do sistema, possibilitando uma visualização fácil e clara dos acontecimentos da cidade, auxiliando na tomada de decisão.

**RF009 - Moderação de postagens**

O sistema deverá permitir, em sua versão web, para usuários administradores, a moderação das postagens de todos os usuários. O administrador poderá excluir mensagens indesejadas, ofensivas e spam.

## APÊNDICE B – DIAGRAMA DE CASO DE USO GERAL

FIGURA 30: DIGRAMA CASO DE USO – GERAL



FONTE: OS AUTORES (2021)

## APÊNDICE C – ESPECIFICAÇÃO DOS CASOS DE USO

### UC001 - Efetuar cadastro

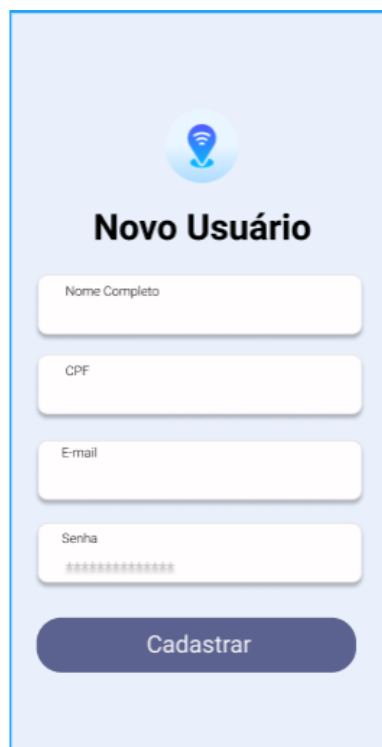
#### Descrição:

O caso de uso inicia-se quando o usuário deseja efetuar seu cadastro para ter acesso às funções do sistema.

#### Data View

#### DV01 - Tela de cadastro para dispositivos móveis

FIGURA 31: DV01 – TELA INICIAL APLICATIVO



A tela de cadastro para dispositivos móveis apresenta um fundo azul claro. No topo, há um ícone de uma lâmpada azul com um sinal de Wi-Fi. Abaixo do ícone, o título "Novo Usuário" é exibido em negrito. Seguem quatro campos de entrada de texto, cada um com um rótulo e um ícone de lupa para busca: "Nome Completo", "CPF", "E-mail" e "Senha". O campo "Senha" contém caracteres ocultos por pontos. Na base da tela, há um botão azul escuro com o texto "Cadastrar" em branco.

FONTE: OS AUTORES (2021)

**DV02 – Tela cadastro de usuário (web)**

FIGURA 32: DV02 – TELA CADASTRO USUÁRIO WEB

A imagem mostra uma interface web para o cadastro de usuário. No topo, há um ícone de lâmpada azul. Abaixo dele, o formulário é branco e contém os seguintes elementos: o rótulo "Nome Completo" seguido de um campo de texto; o rótulo "CPF" seguido de um campo de texto; o rótulo "Email" seguido de um campo de texto; o rótulo "Senha" seguido de um campo de texto; e, no final, um botão cinza com o texto "Cadastrar".

FONTE: OS AUTORES (2021)

**Atores:**

Usuário

**Pré-condições:**

Nenhuma.

**Pós-condições:**

Usuário cadastrado no sistema, e-mail de confirmação enviado ao e-mail informado no cadastro.

**Fluxo Principal:**

1. O usuário deseja se cadastrar no sistema, clicando no botão de Cadastro.
2. O sistema redireciona o usuário para a página específica de cadastro.
3. O usuário preenche os campos nome completo, CPF, e-mail, senha.
4. O sistema valida as informações digitadas e libera o clique ao botão Cadastrar. (E1) (E2) (E3) (E4) (R1) (R2)
5. O usuário clica no botão cadastrar, enviando as informações.
6. O sistema registra o usuário, enviando um e-mail de confirmação conforme e-mail informado no formulário.



7. O caso de uso é encerrado.

### **Fluxos de Exceção:**

#### **E1: Nome inválido:**

1. O sistema informa na tela a mensagem: “Por favor, preencha o campo Nome corretamente”
2. Retorna ao fluxo principal

#### **E2: CPF inválido:**

3. O sistema informa na tela a mensagem: “Por favor, preencha o campo CPF corretamente”
4. Retorna ao fluxo principal.

#### **E3: E-mail inválido:**

5. O sistema informa na tela a mensagem: “Por favor, preencha o campo E-mail corretamente”
6. Retorna ao fluxo principal.

#### **E4: Senha inválida:**

7. O sistema informa na tela a mensagem: “Por favor, preencha o campo Senha corretamente”
8. Retorna ao fluxo principal.

### **Regras de Negócio:**

**R1: Endereço de e-mail já cadastrado:** Se o endereço informado já estiver vinculado a uma conta no sistema, o sistema deve apresentar a mensagem: “E-mail já cadastrado no sistema”.

**R2: CPF já cadastrado:** Se o CPF informado já estiver vinculado a uma conta no sistema, o sistema deve apresentar a mensagem: “CPF já cadastrado no sistema”.

## UC002 - Efetuar Login

### Descrição:

O caso de uso inicia-se quando o usuário deseja efetuar seu login para acessar as funções do sistema.

### Data View

#### DV01 - Tela de login para dispositivos móveis

FIGURA 33: DV01 – TELA DE LOGIN DISPOSITIVO MÓVEL



FONTE: OS AUTORES (2021)

**DV02 - Tela de login (web)**

FIGURA 34: DV02 – TELA DE LOGIN WEB



FONTE: OS AUTORES (2021)

**Atores:**

Usuário

**Pré-condições:**

Nenhuma.

**Pós-condições:**

Usuário autenticado no sistema, podendo acessar as funcionalidades.

**Fluxo Principal:**

1. O usuário deseja logar-se no sistema, clicando no botão de Login.
2. O sistema apresenta a tela de login.
3. O usuário preenche os campos E-mail ou CPF e Senha.
4. O sistema busca os dados do usuário na base de dados. (E1) (E2) (R1)
5. O sistema autentica o usuário e permite o acesso.
6. O caso de uso é encerrado.

**Fluxos de Exceção:****E1: E-mail ou CPF não preenchido:**

1. O sistema informa na tela a mensagem: “Por favor, informe o E-mail ou CPF”
2. Retorna ao fluxo principal.

**E2: Senha não preenchida:**

1. O sistema informa na tela a mensagem: “Por favor, informe a senha”
2. Retorna ao fluxo principal.

**Regras de Negócio:**

**R1: Login não encontrado ou senha inválida:** Se o login não for encontrado na base de dados ou a senha não corresponder, o sistema deve apresentar a mensagem: “Usuário não cadastrado ou senha inválida”.

**UC003 – Fazer Postagem em Dispositivo Móvel****Descrição**

O UC permite o usuário fazer uma postagem pelo celular, preenchendo os dados necessários.

## Data View

### DV01 - Tela de criar postagem

FIGURA 35: DV01 – TELA NOVA POSTAGEM APLICATIVO

A imagem mostra a tela de criação de uma nova postagem em um aplicativo. No topo, há um ícone de lâmpada azul e o título "Nova Postagem". Abaixo, há um formulário com os seguintes campos:

- Título:** Um campo de texto contendo "Buraco na Rua Fulano da Silva".
- Local:** Um campo de texto contendo "Localização atual".
- Categoria:** Um menu suspenso com o valor "Transporte" selecionado.
- Subcategoria:** Um menu suspenso com o valor "Reclamação" selecionado.
- Descrição:** Um campo de texto grande e vazio.
- Inserir foto:** Um ícone de câmera e o texto "Inserir foto" ao lado do campo de descrição.

Na base do formulário, há um botão azul arredondado com o texto "Postar".

FONTE: OS AUTORES (2021)

## Ator Primário

Usuário

## Pré-requisitos

Estar logado com uma conta de Usuário.

## Fluxo de Eventos Principal

- 1) O sistema preenche o campo Local com a localização atual, caso o usuário permita sua localização seja compartilhada pelo celular. (A1)
- 2) O usuário preenche o campo 'Categoria'

- 3) O usuário preenche o campo 'Subcategoria'
- 4) O usuário preenche o campo 'Descrição'
- 5) O usuário faz upload de uma foto pelo botão 'Inserir Foto' (E1) (E2)
- 6) O usuário clica no botão 'Postar'
- 7) O sistema calcula latitude e longitude a partir do endereço selecionado.
- 8) O sistema registra o post na entidade 'Postagem' no banco de dados, colocando a data atual como data de cadastro.

### **Fluxo de Eventos Alternativo**

**A1** – Usuário não permite acesso a sua localização – campo não é preenchido automaticamente.

### **Fluxo de Exceções**

**E1** – Usuário tentou fazer upload, mas ocorreu erro na transmissão

- 1) Exibe mensagem "Erro ao fazer upload do arquivo X".
- 2) Volta para a tela de nova postagem

**E2** – Usuário tenta fazer upload de formato incorreto de arquivo

- 1) Exibe mensagem "Formato de arquivo incorreto. Selecione um arquivo JPEG."
- 2) Volta para a tela de nova postagem.

### **Regras de Negócio**

**R1** – Sistema deve verificar se os campos Descrição, Categoria e Subcategoria estão preenchidos, pois são obrigatórios. Caso não estejam preenchidos, o sistema exibe mensagem de alerta.

**R2** – Formatos admitidos para a foto são JPEG e PNG

## **UC004 – Fazer Postagem (Web)**

### **Descrição**

O UC permite o usuário fazer uma postagem pela Web, preenchendo os dados necessários.

## Data View

### DV01 - Tela de criar postagem

FIGURA 36: DV01 – TELA NOVA POSTAGEM WEB

The image shows a mobile web interface for creating a new post. On the left, a map displays a street grid with red location pins. The main form, titled 'Nova Postagem', contains the following elements: a text input for 'Titulo' with the value 'Buraco na Rua Fulano da Silva'; a text input for 'Local' with the value 'Localização atual'; a dropdown menu for 'Categoria' set to 'Transporte'; a dropdown menu for 'Subcategoria' set to 'Reclamação'; a large text area for 'Descrição'; a button with a camera icon labeled 'Inserir foto'; and a blue 'Postar' button at the bottom.

FONTE: OS AUTORES (2021)

### Ator Primário

Usuário

### Pré-requisitos

Estar logado com uma conta de Usuário.

### Fluxo de Eventos Principal

- 1) O usuário preenche o campo Local com a localização do incidente, e o sistema busca o endereço digitado.
- 2) O usuário preenche o campo 'Categoria'
- 3) O usuário preenche o campo 'Subcategoria'
- 4) O usuário preenche o campo 'Descrição'
- 5) O usuário faz upload de uma foto pelo botão 'Inserir Foto' (E1) (E2)
- 6) O usuário clica no botão 'Postar'
- 7) O sistema calcula latitude e longitude a partir do endereço selecionado.

8) O sistema registra o post na entidade 'Postagem' no banco de dados, colocando a data atual como data de cadastro. (E3)

### **Fluxo de Exceções**

**E1** – Usuário tentou fazer upload, mas ocorreu erro na transmissão

- 1) Exibe mensagem “Erro ao fazer upload do arquivo X”.
- 2) Volta para a tela de nova postagem

**E2** – Usuário tenta fazer upload de formato incorreto de arquivo

- 3) Exibe mensagem “Formato de arquivo incorreto. Selecione um arquivo JPEG.”.
- 4) Volta para a tela de nova postagem

**E3** – Endereço digitado não encontrado

- 1) Exibe mensagem “Endereço não encontrado.”.
- 2) Volta para a tela de nova postagem

### **Regras de Negócio**

**R1** – Os campos Descrição, Categoria e Subcategoria são obrigatórios.

**R2** – Formatos admitidos para a foto são JPEG e PNG

## **UC005 – Comentar Postagem na Web ou em Dispositivos Móveis**

### **Descrição**

O UC permite o usuário fazer um comentário em uma postagem



## Data View

### DV01 - Tela comentar postagem

FIGURA 37: DV01 – TELA COMENTAR POSTAGEM



FONTE: OS AUTORES (2021)

## Ator Primário

Usuário

## Pré-requisitos

Estar logado com uma conta de Usuário.

## Fluxo de Eventos Principal

- 1) O usuário abre uma postagem, clicando em um pin do mapa
- 2) O sistema abre uma modal com os detalhes do post
- 3) O usuário digita uma mensagem no campo 'Digite um comentário' e pressiona enviar (E1)
- 4) O sistema cadastra o comentário na entidade "Comentário" do DB, atribuindo o id do usuário, data atual e id da postagem automaticamente.
- 5) A modal do post é atualizada com o novo comentário do usuário.

## Fluxo de Exceções

**E1** – Usuário não digitou ao menos 10 caracteres

- 1) Exibe mensagem “Digite pelo menos 10 caracteres”.
- 2) Volta para a tela da postagem, sem adicionar o comentário

## UC004 – Curtir Postagem na Web ou em Dispositivos Móveis

### Descrição

O UC permite o usuário curtir uma postagem do mapa

### Data View

**DV01 - Tela de curtir postagem**

FIGURA 38: DV01 – TELA CURTIR POSTAGEM



FONTE: OS AUTORES (2021)

### Ator Primário

Usuário

### Pré-requisitos

Estar logado com uma conta de Usuário.

### Fluxo de Eventos Principal

- 1) O usuário abre uma postagem, clicando em um pin do mapa
- 2) O sistema abre uma modal com os detalhes do post
- 3) O usuário pressiona o botão de curtir, logo abaixo da foto da postagem. (E1)

- 4) O sistema cadastra a curtida na entidade “Curtida” no BD, passando o id da postagem e o id do usuário
- 5) A modal do post é atualizada com o novo número de curtidas.

### Fluxo de Exceções

**E1 – Erro na conexão ao tentar dar like**

- 1) Exibe mensagem “Erro. Tente novamente”.
- 2) Volta para a tela da postagem, sem adicionar a curtida.

## UC006 – Resolver Post na Web ou em Dispositivos Móveis

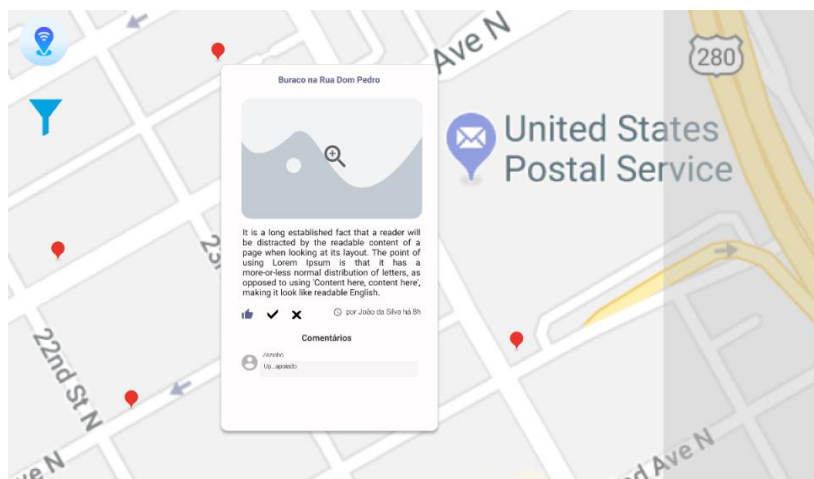
### Descrição

O UC permite o administrador marcar um post do mapa como resolvido

### Data View

**DV01 - Tela de resolver postagem**

FIGURA 39: DV01 – TELA RESOLVER POSTAGEM



FONTE: OS AUTORES (2021)

**Ator Primário**

Administrador

**Pré-requisitos**

Estar logado com uma conta de Administrador.

**Fluxo de Eventos Principal**

- 1) O administrador abre uma postagem, clicando em um pin do mapa
- 2) O sistema abre uma modal com os detalhes do post
- 3) O administrador pressiona o botão de resolver post
- 4) O sistema cadastra na entidade “Postagem” na linha “Resolvida”, e passará a ser exibido como resolvida para outros usuários.

**UC007 – Filtrar por Bairro na Web ou em Dispositivos Móveis****Descrição**

O UC permite que o usuário logado filtre os pins do mapa por bairro

**Ator Primário**

Usuário

**Pré-requisitos**

Estar logado com uma conta de Usuário

**Fluxo de Eventos Principal**

- 1) Na visualização principal, o usuário pressiona o ícone de filtro
- 2) O sistema abre uma modal com os tipos de filtro
- 3) O usuário escolhe um dos bairros disponíveis (A1)
- 4) O mapa principal é atualizado, filtrando apenas os pins com o bairro selecionado

**Fluxo de Eventos Alternativo**

**A1** – Usuário escolhe categoria ao invés de bairro

- 1) Mapa será atualizado de acordo com a categoria selecionada.

## UC008 – Visualizar Postagens na Web ou em Dispositivos Móveis

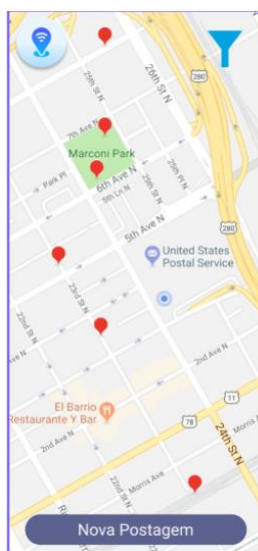
### Descrição

O UC permite ao usuário visualizar o mapa com as postagens de outros usuários.

### Data View

#### DV01 - Tela de visualizar postagens

FIGURA 40: DV01 – TELA VISUALIZAR POSTAGENS



FONTE: OS AUTORES (2021)

### Ator Primário

Usuário

### Pré-requisitos

Estar logado com uma conta de Usuário.

### Fluxo de Eventos Principal

- 1) Usuário faz login
- 2) O sistema carrega o mapa com base na localização atual do usuário, com pins em cada post (A1)

## Fluxo de Eventos Alternativo

**A1** – Usuário não permite acesso a sua localização – o mapa é carregado em um lugar padrão de Curitiba

## UC009 – Visualizar Dashboards (Web)

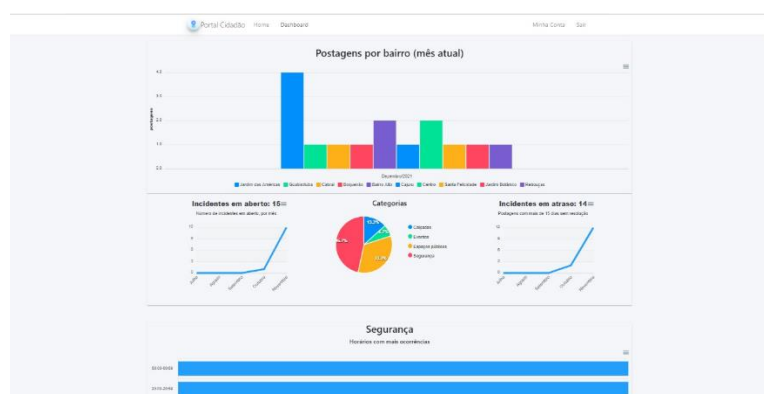
### Descrição

O UC permite o administrador visualizar a página com dashboards avançados.

### Data View

#### DV01 - Tela de visualizar dashboards

FIGURA 41: DV01 – TELA VISUALIZAR DASHBOARDS - ADMINISTRADOR



FONTE: OS AUTORES (2021)

### Ator Primário

Administrador

### Pré-requisitos

Estar logado com uma conta de Administrador

### Fluxo de Eventos Principal

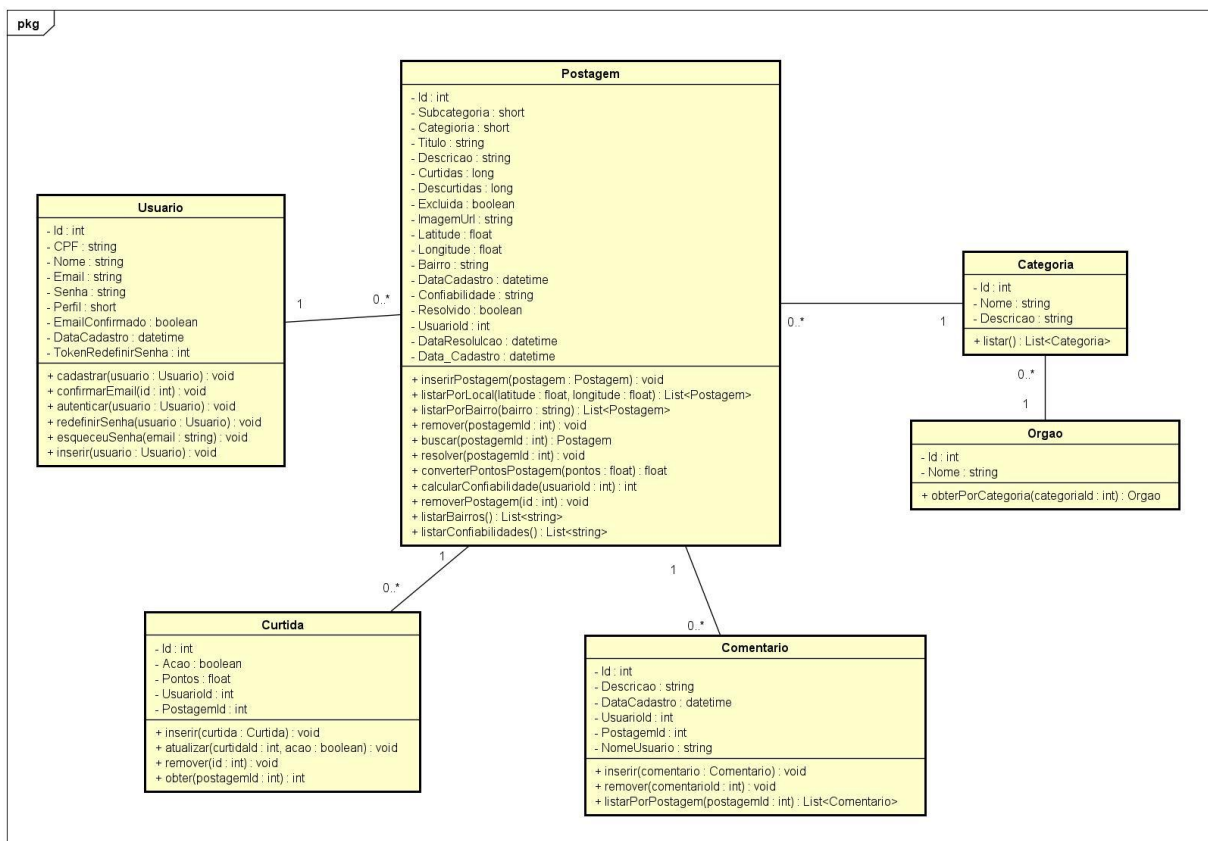
- 1) Da página principal, o administrador pressiona o botão de acesso aos dashboards
- 2) O sistema carrega os dashboards com os dados atuais do sistema (E1)

**Fluxos de Exceção****E1 – Erro ao Carregar Dashboard**

- 1) Sistema exibe uma imagem de erro na área onde o dashboard deveria ser exibido.

## APÊNDICE D - DIAGRAMA DE CLASSES

FIGURA 42: DIAGRAMA DE CLASSES



FONTE: OS AUTORES (2021)



## APÊNDICE E - DIAGRAMA ENTIDADE RELACIONAMENTO

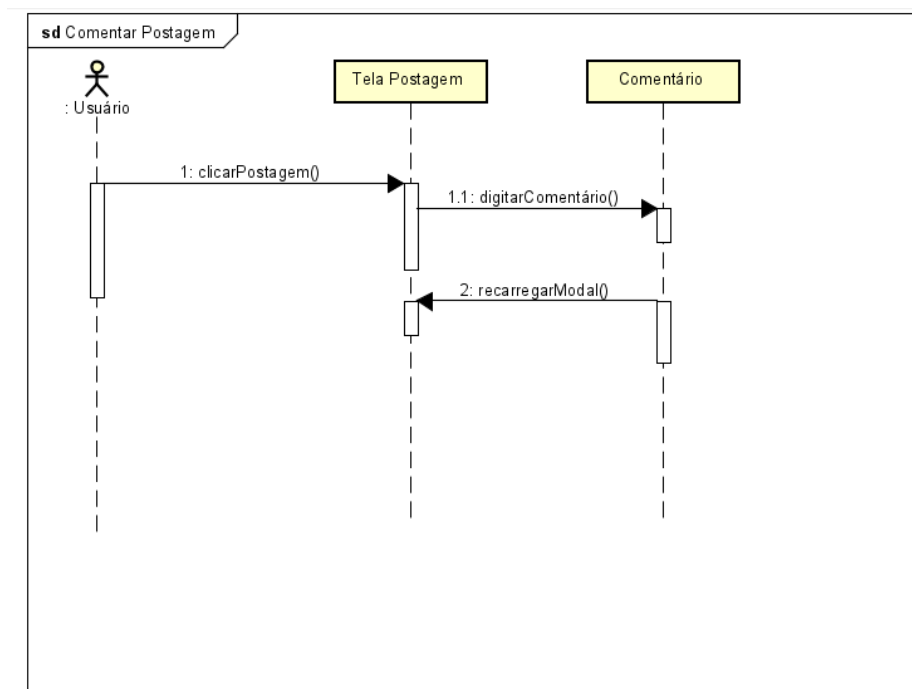
FIGURA 43: DIAGRAMA DE ENTIDADE RELACIONAMENTO



FONTE: OS AUTORES (2021)

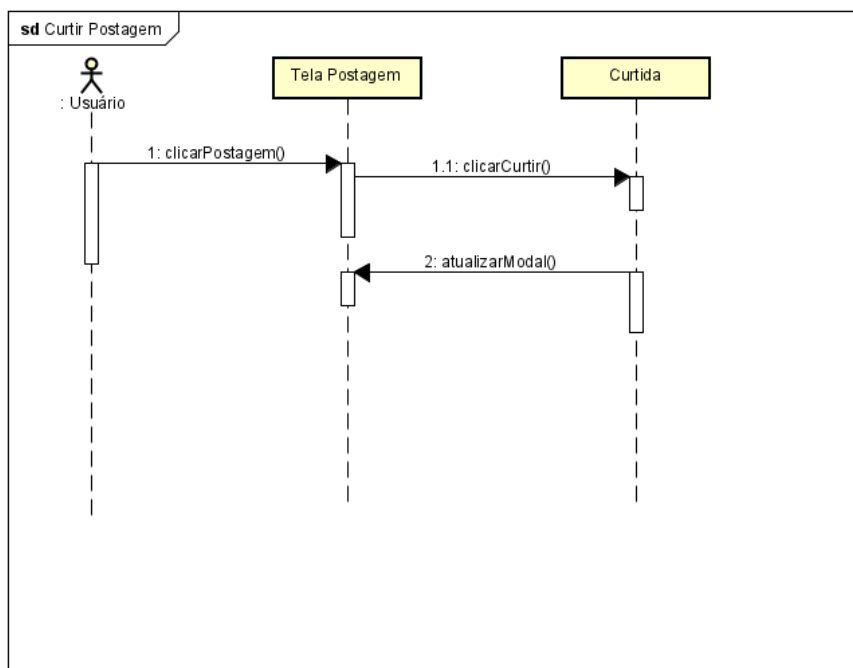
## APÊNDICE F – DIAGRAMAS DE SEQUÊNCIA

FIGURA 44: COMENTAR POSTAGEM



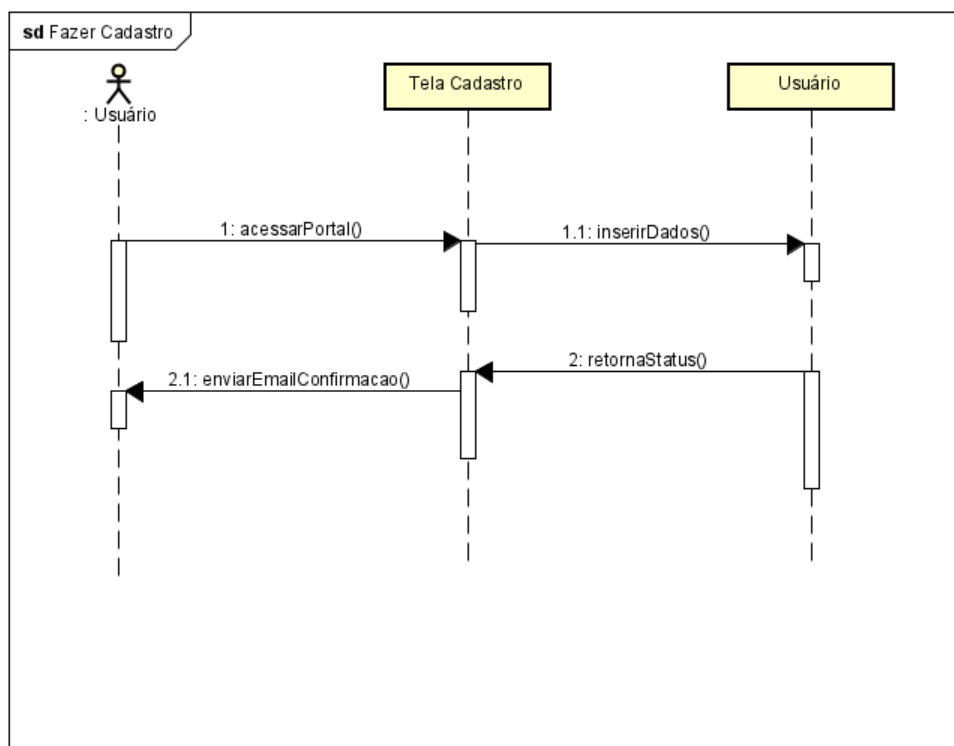
FONTE: OS AUTORES (2021)

FIGURA 45: CURTIR POSTAGEM



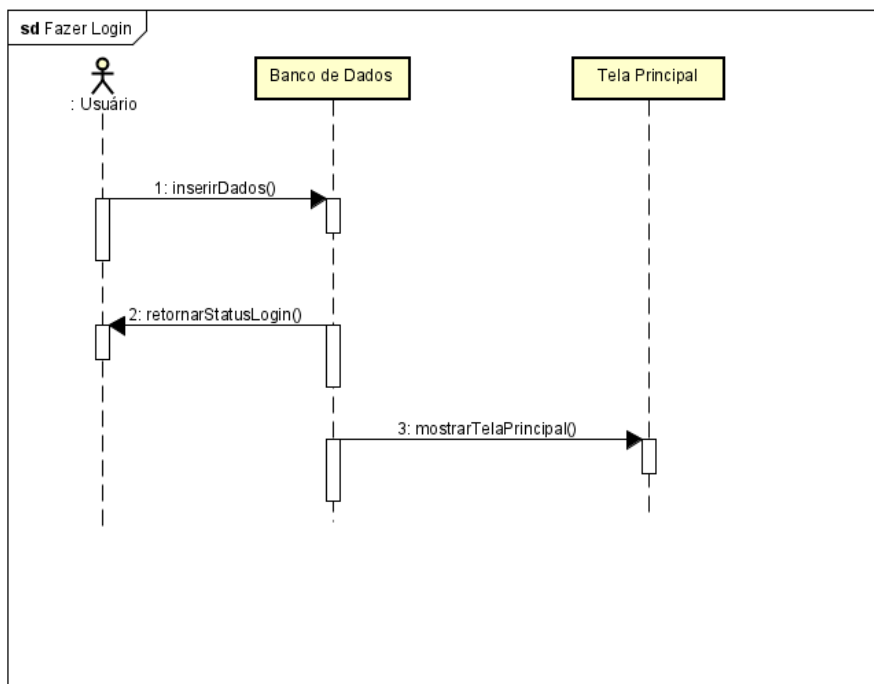
FONTE: OS AUTORES (2021)

FIGURA 46: FAZER CADASTRO



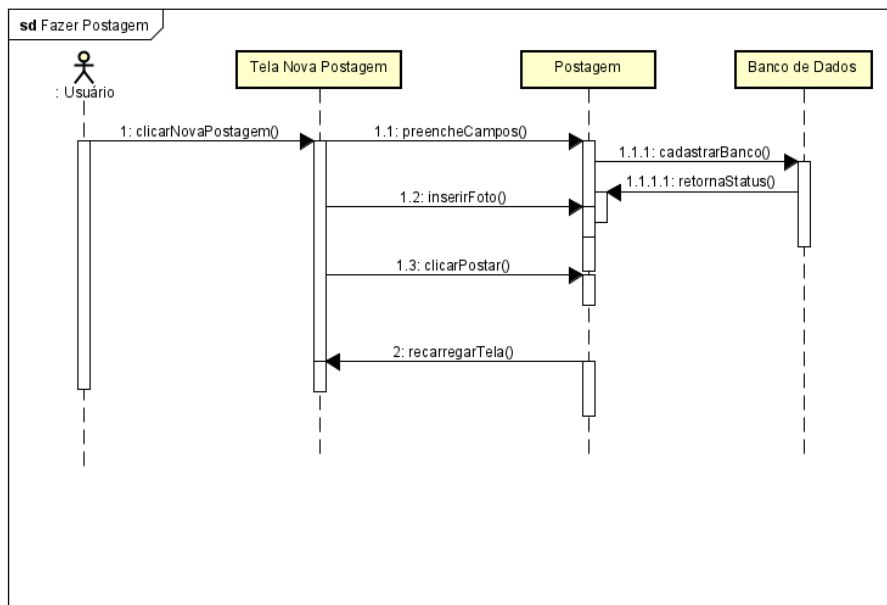
FONTE: OS AUTORES (2021)

FIGURA 47: FAZER LOGIN



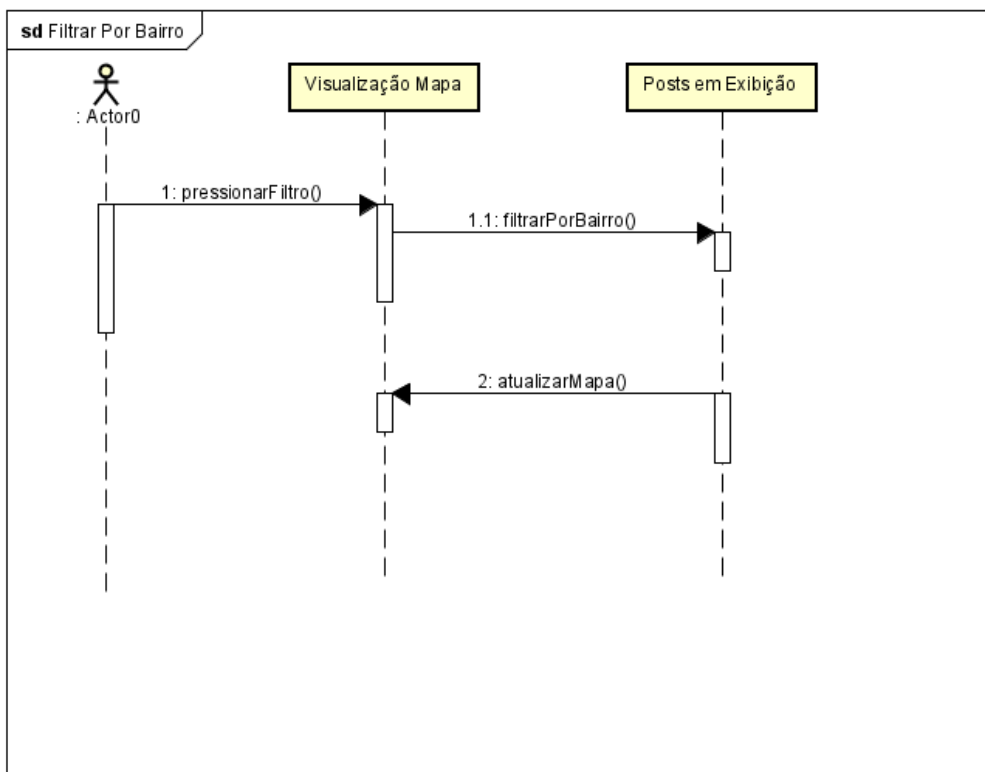
FONTE: OS AUTORES (2021)

FIGURA 48: FAZER POSTAGEM



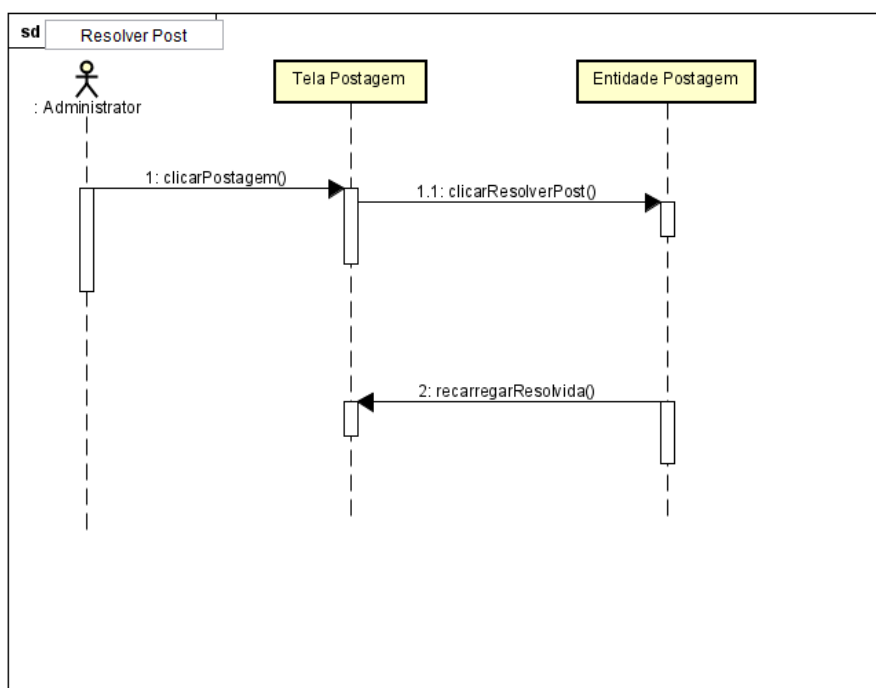
FONTE: OS AUTORES (2021)

FIGURA 49: FILTRAR POR BAIRRO



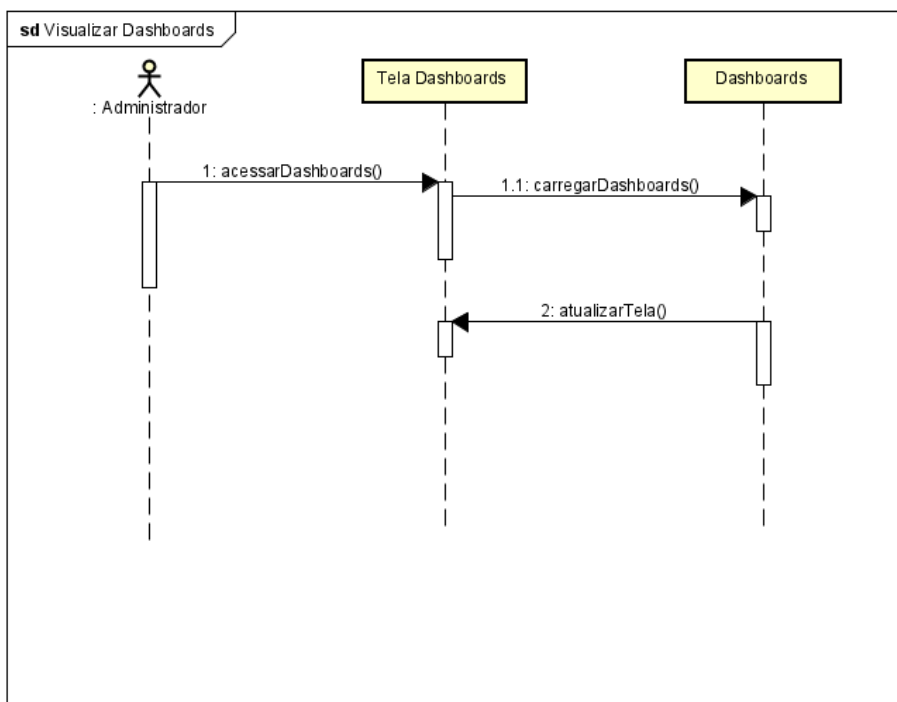
FONTE: OS AUTORES (2021)

FIGURA 50: RESOLVER POST



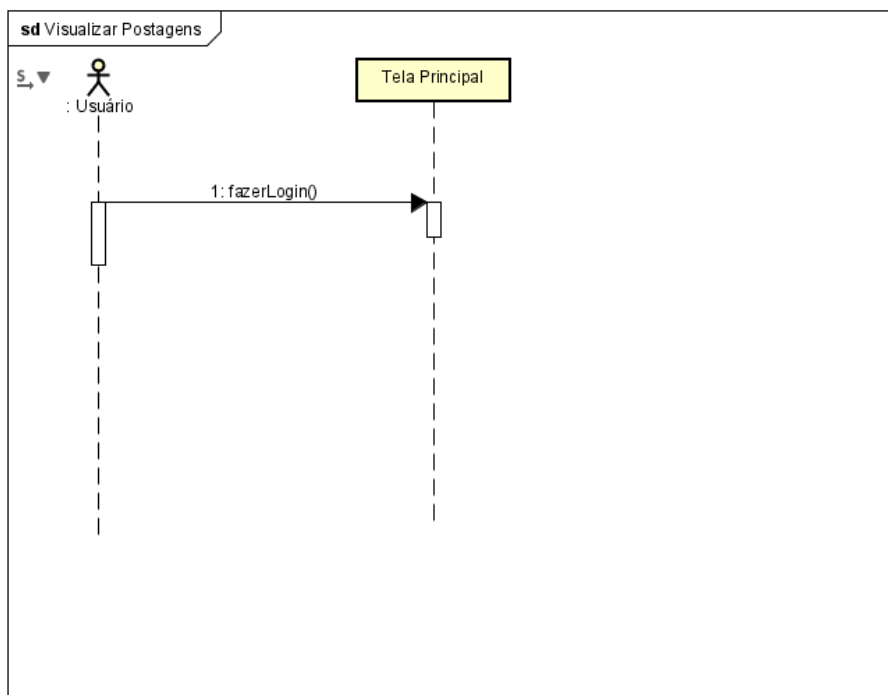
FONTE: OS AUTORES (2021)

FIGURA 51: VISUALIZAR DASHBOARDS



FONTE: OS AUTORES (2021)

FIGURA 52: VISUALIZAR POSTAGENS



FONTE: OS AUTORES (2021)

## APÊNDICE G – DIAGRAMA DE GANTT 1

FIGURA 53: DIAGRAMA GANTT (TCC-1)



FONTE: OS AUTORES (2021)

## APÊNDICE H – DIAGRAMA DE GANTT 2

FIGURA 54: DIAGRAMA GANTT (TCC-2)

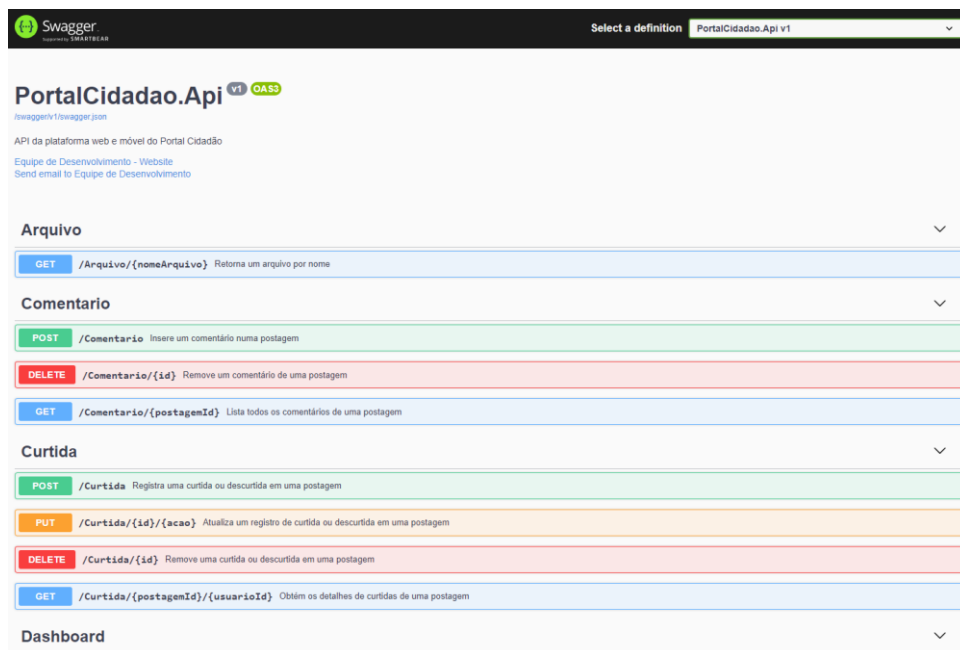
	Sprint 14	Sprint 15	Sprint 16	Sprint 17	Sprint 18	Sprint 19	Sprint 20	Sprint 21	Sprint 22
Filtro por categoria e subcategoria	Red								
Implementação de perfis	Yellow	Yellow	Grey	Grey	Grey	Grey	Grey	Grey	Grey
Resolução de posts		Black							
Resolução de bugs	Grey	Grey	Red	Grey	Grey	Grey	Grey	Grey	Grey
Redefinição de senha				Orange					
Moderação de posts	Grey	Grey	Grey	Black	Grey	Grey	Grey	Grey	Grey
Criação de dashboards					Red	Red			
Regras de pontuação	Grey	Grey	Grey	Grey	Grey	Blue	Grey	Grey	Grey
Alterar marcador						Yellow			
Filtro por confiabilidade	Grey	Grey	Grey	Grey	Grey	Grey	Red	Grey	Grey
Mostrar confiabilidade na modal								Yellow	
Testes e correções	Grey	Grey	Grey	Grey	Grey	Grey	Grey	Black	Black
Finalização da documentação							Blue	Blue	Blue

FONTE: OS AUTORES



## APÊNDICE I – DOCUMENTAÇÃO DA API DO SISTEMA

### FIGURA 55: DOCUMENTAÇÃO DE API VIA SWAGGER



The image shows a screenshot of the Swagger API documentation interface. At the top, the Swagger logo is visible on the left, and a dropdown menu on the right is set to "PortalCidadao.Api v1". The main heading is "PortalCidadao.Api" with a "v1" tag and an "OAS3" icon. Below the heading, there is a link to the Swagger JSON file and a brief description of the API as the platform for the web and mobile of Portal Cidadão, along with contact information for the development team.

The interface is organized into sections, each with a dropdown arrow on the right:

- Arquivo**: Contains one endpoint: `GET /Arquivo/{nomeArquivo}` with the description "Retorna um arquivo por nome".
- Comentario**: Contains three endpoints:
  - `POST /Comentario` with the description "Insere um comentário numa postagem".
  - `DELETE /Comentario/{id}` with the description "Remove um comentário de uma postagem".
  - `GET /Comentario/{postagemId}` with the description "Lista todos os comentários de uma postagem".
- Curtida**: Contains three endpoints:
  - `POST /Curtida` with the description "Registra uma curtida ou descurtida em uma postagem".
  - `PUT /Curtida/{id}/{acao}` with the description "Atualiza um registro de curtida ou descurtida em uma postagem".
  - `DELETE /Curtida/{id}` with the description "Remove uma curtida ou descurtida em uma postagem".
  - `GET /Curtida/{postagemId}/{usuarioId}` with the description "Obtém os detalhes de curtidas de uma postagem".
- Dashboard**: No endpoints are listed under this section.

FONTE: OS AUTORES