



UNIVERSIDADE FEDERAL DE SANTA CATARINA  
CENTRO TECNOLÓGICO  
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA

José Augusto da França

**Sistema para reportar problemas e sugestões na estrutura física de um campus  
universitário.**

Florianópolis  
2022

José Augusto da França

**Sistema para reportar problemas e sugestões na estrutura física de um campus universitário.**

Trabalho de Conclusão de Curso submetida ao Departamento de Informática e Estatística da Universidade Federal de Santa Catarina para a obtenção do título de Bacharel em Sistemas de Informação.  
Orientador: Prof. José Eduardo De Lucca, Dr.

Florianópolis  
2022

Dedico este trabalho ao filho, Miguel Cividini da França,  
e ao meu falecido pai Amauri Eugênio da França.

## **AGRADECIMENTOS**

Agradeço primeiramente a minha esposa, Amanda Brunetta Cividini, que sempre apoiou e me incentivou durante toda minha trajetória na universidade, que por diversas vezes me ouviu falar sobre tecnologia por horas com muita paciência e que sempre acreditou em mim.

Aos colegas e amigos que fiz durante essa jornada na Universidade, em especial ao Gabriel Donadel, Peter Krause e Samuel Landre, amigos que levarei para a vida, que por diversas vezes foram companheiros de equipes onde atravessamos noites estudando para resolução de trabalhos e estudo.

Agradeço aos professores e servidores contribuíram para a minha formação e em especial ao Prof. Dr. Jose Eduardo de Lucca, meu orientador, que me orientou e me deu suporte durante a realização deste trabalho. Também agradeço a Prof<sup>a</sup>. Dra. Lúcia Helena Martins Pacheco e ao Prof. Dr. Jose Francisco Danilo de Guadalupe Correa Fletes por compor minha banca e contribuírem para a conclusão deste trabalho.

Por fim, agradeço à todos que contribuíram direta ou indiretamente para meu desenvolvimento e formação.

*“A mudança não acontecerá se nós  
esperarmos por outra pessoa ou  
se esperarmos por algum outro momento.  
Nós somos as pessoas pelas quais esperávamos.  
Nós somos a mudança que buscamos.”  
(Obama, 2008)*

## RESUMO

As universidades estão em constante processo de manutenção e melhoria de sua estrutura física a fim de proporcionar a toda comunidade acadêmica um ambiente que apoie o ensino, a pesquisa e a extensão. Desta forma, visando fomentar um processo colaborativo entre os membros das comunidades com suas instituições, este trabalho implementa um sistema que ajude os membros da comunidade acadêmica a reportarem problemas de infra-estrutura e a fazerem sugestões de melhorias sobre estrutura física de um campus universitário. Para este trabalho utilizou-se o contexto do Campus Reitor João David Ferreira Lima da Universidade Federal de Santa Catarina, por meio de um sistema *web* para receber os pedidos de manutenções. O software em questão visa diminuir a distância na comunicação entre os membros da comunidade universitária e a instituição, possibilitando à universidade identificar as áreas que necessitam mais atenção através da interação dos usuários atestando a existência das demandas. E possibilitando à comunidade acadêmica ter conhecimento se a Universidade está trabalhando em uma solicitação ou se ela já foi resolvida.

**Palavras-chave:** Plataforma de participação colaborativa, Desenvolvimento Web, PWA.

## **ABSTRACT**

Universities are in a constant process of maintenance and improvement of their physical structure in order to provide the entire academic community with an environment that supports teaching, research and extension. Thus, aiming to foster a collaborative process between community members and their institutions, this work implements a system that helps members of the academic community to report infrastructure problems and make suggestions for improvements on the physical structure of a university campus. For this work, the context of the Campus Reitor João David Ferreira Lima of the Federal University of Santa Catarina was used, through a web system to receive maintenance requests. The software in question aims to reduce the distance in communication between the members of the university community and the institution, allowing the university to identify the areas that need more attention through the interaction of users, attesting to the existence of demands. And allowing the academic community to know if the University is working on a request or if it has already been resolved.

**Keywords:** Collaborative participation platform, Web Development, PWA.

## LISTA DE FIGURAS

Figura 1 – Estrutura Organizacional da Administração da Universidade . . . . .	16
Figura 2 – Estrutura Organizacional do Centro Tecnológico . . . . .	17
Figura 3 – Organograma Secretaria de Obras, Manutenção e Ambiente (SEOMA)	18
Figura 4 – Processos de trabalhos - Núcleo de Manutenção (NUMA) . . . . .	18
Figura 5 – Organograma UFS . . . . .	19
Figura 6 – Solicitação de melhoria no aplicativo Colab . . . . .	20
Figura 7 – Aba de conquistas no aplicativo Colab . . . . .	21
Figura 8 – Módulo Monitor . . . . .	22
Figura 9 – Módulo Panorama . . . . .	22
Figura 10 – Mapa com as solicitações no aplicativo ParticipAct. . . . .	23
Figura 11 – Tabela com as solicitações feitas no aplicativo ParticipAct nos últimos meses. . . . .	24
Figura 12 – Open311 da cidade de Boston . . . . .	26
Figura 13 – Reports FixMyStreet do Council Merton . . . . .	27
Figura 14 – Gráficos FixMyStreet de solicitações enviadas e resolvidas. . . . .	28
Figura 15 – Gráficos FixMyStreet de efetividade. . . . .	29
Figura 16 – Tabela comparativa das funcionalidades das aplicações pesquisadas.	30
Figura 17 – Diagrama de caso de uso: Universidade. . . . .	32
Figura 18 – Diagrama de caso de uso: Comunidade. . . . .	33
Figura 19 – Tela de relatórios . . . . .	34
Figura 20 – Tela de entradas (solicitações) . . . . .	34
Figura 21 – Tela de cadastro de centros . . . . .	35
Figura 22 – Arquitetura do sistema . . . . .	36
Figura 23 – Diagrama UML . . . . .	38
Figura 24 – Hasura Console - Criação de tabela . . . . .	40
Figura 25 – Hasura Console - Consulta GraphQL . . . . .	40
Figura 26 – Sistema - Acesso ao sistema versão <i>web</i> . . . . .	41
Figura 27 – Sistema - Acesso ao sistema versão <i>mobile</i> . . . . .	42
Figura 28 – Fluxo de informação - Acesso ao sistema . . . . .	43
Figura 29 – Sistema - Criação de Conta versão <i>web</i> . . . . .	44
Figura 30 – Sistema - Criação de Conta versão <i>mobile</i> . . . . .	44
Figura 31 – Sistema - Recuperação de senha versão <i>web</i> . . . . .	45
Figura 32 – Sistema - Recuperação de senha versão <i>mobile</i> . . . . .	45
Figura 33 – Sistema Comunidade - Minhas solicitações . . . . .	46
Figura 34 – Sistema Comunidade - Minhas solicitações versão <i>mobile</i> . . . . .	47
Figura 35 – Sistema Comunidade - Solicitações da comunidade . . . . .	47
Figura 36 – Sistema Comunidade - Detalhes de uma solicitação . . . . .	48



Figura 37 – Sistema Comunidade - Criação de solicitação versão <i>web</i> . . . . .	49
Figura 38 – Sistema Comunidade - Criação de solicitação versão <i>mobile</i> . . . . .	49
Figura 39 – Fluxo de informação - Criação de solicitação . . . . .	50
Figura 40 – Estrutura inicial do projeto em NextJS . . . . .	50
Figura 41 – Estrutura de arquivos/paginas do sistema . . . . .	51
Figura 42 – Comunidade - Botão carregar mais . . . . .	52
Figura 43 – Comunidade - Detalhes da solicitação . . . . .	54
Figura 44 – Universidade - Seção de dados gerais e gráficos de solicitações . . . . .	55
Figura 45 – Universidade - Gráfico de rosca . . . . .	55
Figura 46 – Universidade - Tabela das últimas dez solicitações . . . . .	56
Figura 47 – Universidade - Tabela das dez solicitações mais votadas . . . . .	56
Figura 48 – Solicitação - Opções rejeitar e em progresso . . . . .	57
Figura 49 – Solicitação - Opção marcar como resolvido . . . . .	58
Figura 50 – Universidade - Unidades . . . . .	59
Figura 51 – Universidade - Prédios/Áreas . . . . .	59
Figura 52 – Universidade - Categorias . . . . .	60
Figura 53 – Universidade - Criação de uma nova opção de prédio ou área . . . . .	61

## LISTA DE ABREVIATURAS E SIGLAS

API	Application Programming Interface
CGA	Coordenadoria de Gestão Ambiental
CSR	Client Side Rendering
CSS	Cascading Style Sheets
CTC	Centro Tecnológico
DEMAN	Departamento de Manutenção
DIMEQ	Divisão de Manutenção de Máquinas e Equipamentos
DMPI	Departamento de Manutenção Predial e Infraestrutura
HTML	HyperText Markup Language
INE	Departamento de Informática e Estatística
INFRAUFS	Superintendência de Serviços de Infraestrutura
NUMA	Núcleo de Manutenção
PU	Prefeitura Universitária
PWA	Progressive web apps
SEOMA	Secretaria de Obras, Manutenção e Ambiente
SQL	Standard Query Language
SSR	Server Side Rendering
UDESC	Universidade do Estado de Santa Catarina
UFS	Universidade Federal do Sergipe
UFSC	Universidade Federal de Santa Catarina
UML	Unified Modeling Language
UNIBO	Universidade de Bologna
URL	Uniform Resource Locator

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>12</b>
1.1	OBJETIVOS	13
<b>1.1.1</b>	<b>Objetivo Geral</b>	<b>13</b>
<b>1.1.2</b>	<b>Objetivos Específicos</b>	<b>13</b>
1.2	MÉTODO DE PESQUISA	13
1.3	JUSTIFICATIVA	14
<b>2</b>	<b>ANÁLISE DA ORGANIZAÇÃO DAS UNIVERSIDADES</b>	<b>16</b>
<b>3</b>	<b>ANÁLISE DOS SISTEMAS SIMILARES</b>	<b>20</b>
3.1	COLAB	20
3.2	PARTICIPACT	23
3.3	OPEN311	24
3.4	FIXMYSTREET	26
3.5	AVALIAÇÃO DAS FERRAMENTAS ESTUDADAS	29
<b>4</b>	<b>DESENVOLVIMENTO DA APLICAÇÃO</b>	<b>31</b>
4.1	PESQUISA E LEVANTAMENTO DE REQUISITOS	31
<b>4.1.1</b>	<b>Requisitos não funcionais</b>	<b>31</b>
<b>4.1.2</b>	<b>Requisitos funcionais</b>	<b>31</b>
4.2	CASO DE USO	32
4.3	PROTOTIPAÇÃO DO SISTEMA	33
4.4	TECNOLOGIAS UTILIZADAS NO SISTEMA	35
<b>4.4.1</b>	<b>Backend</b>	<b>36</b>
<b>4.4.2</b>	<b>Frontend</b>	<b>37</b>
4.5	SISTEMA	37
<b>4.5.1</b>	<b>API e Banco de dados</b>	<b>37</b>
<b>4.5.2</b>	<b>Acesso ao sistema</b>	<b>41</b>
<b>4.5.3</b>	<b>Comunidade</b>	<b>46</b>
4.5.3.1	Minhas solicitações	52
4.5.3.2	Solicitações da comunidade	53
4.5.3.3	Detalhes da solicitação	53
<b>4.5.4</b>	<b>Universidade</b>	<b>54</b>
4.5.4.1	Relatórios	54
4.5.4.2	Solicitações e Detalhe de Solicitação	57
4.5.4.3	Unidades, Prédios/Áreas e Categorias	58
<b>5</b>	<b>CONCLUSÃO</b>	<b>62</b>
5.1	SUGESTÕES PARA TRABALHOS FUTUROS	62
	<b>REFERÊNCIAS</b>	<b>64</b>

<b>APÊNDICE A – ARTIGO SISTEMA PARA REPORTAR PROBLEMAS E SUGESTÕES NA ESTRUTURA FÍSICA DE UM CAMPOS UNIVERSITÁRIO . . . . .</b>	<b>66</b>
<b>APÊNDICE B – CÓDIGO FONTE . . . . .</b>	<b>87</b>

## 1 INTRODUÇÃO

Devido a pandemia de COVID-19, as instituições de ensino superior federais no Brasil ficaram com os espaços ociosos nesses dois anos de restrições sanitárias. Desta forma, com o retorno das atividades presenciais nas universidades é esperado que o número de chamados de manutenção aumente. O presente trabalho implementou um sistema que permite à comunidade universitária comunicar à Universidade, de forma centralizada, problemas na estrutura física e sugestões de melhoria. Foi realizado o mapeamento do fluxo de informação para solução das demandas da manutenção das estruturas, uma vez que a Universidade tem diferentes órgãos e/ou setores que são responsáveis por manter e aprimorar todo ecossistema, tal mapeamento foi importante para esclarecer como está estruturada toda a cadeia de responsabilidades dentro da universidade.

O desenvolvimento se baseia em soluções já existentes no mercado voltadas para municípios, como a ParticipAct Brasil<sup>1</sup>, projeto desenvolvido pela Universidade do Estado de Santa Catarina (UDESC) em parceria com a Universidade Federal de Santa Catarina (UFSC) e a Universidade de Bologna (UNIBO) que tem como principal objetivo a criação de uma grande estrutura de banco de dados a partir das informações das tecnologias da informação e da comunicação, com o intuito de melhorar a gestão de cidades inteligentes. Outro exemplo é o sistema desenvolvido pela *startup* Colab<sup>2</sup> que tem como objetivo “dar à população o poder de colaborar com questões do setor público através de publicações de zeladoria urbana, participação em tomadas de decisões e em consultas de avaliação de serviços públicos”.

Para o desenvolvimento de um sistema que atenda a demanda dos Campi Universitários, além da interface web para a administração da Universidade e a interface responsiva *mobile* para interação da comunidade, foi elaborado um modo de interação da comunidade com as solicitações criadas. Cada usuário pode dar *like* para confirmar ou concordar com a solicitação ou *unlike* caso não concorde ou não confirme, desta forma as solicitações com mais *likes* serão mostradas primeiramente.

A realização do trabalho também prevê o desenvolvimento de uma *Application Programming Interface (API)* de comunicação com banco de dados que irá persistir esses dados, tornando assim possível a interação entre a comunidade universitária e a visualização dos pedidos de manutenção por parte da universidade.

---

<sup>1</sup><http://participact.tecnologia.ws/>

<sup>2</sup><https://www.colab.re/>

## 1.1 OBJETIVOS

### 1.1.1 Objetivo Geral

O objetivo do presente trabalho foi desenvolver um sistema que permita aos membros de um Campus Universitário reportarem problemas referente a estrutura física e possibilite à administração do Campus visualizar essas demandas de manutenção.

### 1.1.2 Objetivos Específicos

Para atender ao objetivo geral, foi necessário traçar alguns objetivos específicos:

- Avaliar as estruturas organizacionais da universidade para obter o conhecimento sobre o fluxo de informação;
- Avaliar soluções já existentes no mercado a fim de levar possíveis pontos de melhorias ou adaptações para o contexto universitário;
- Elaborar um sistema de *ranking* das solicitações, por meio dos botões *like* e *unlike*;
- Criar interfaces *web* para interação comunidade-universidade.
- Criar interfaces responsivas *mobile* que possibilitam melhor interação da comunidade com o sistema;
- Desenvolver uma *API* que irá suportar aplicação e efetuar a comunicação entre interface e banco de dados.

## 1.2 MÉTODO DE PESQUISA

Para a elaboração deste trabalho foram adotados os seguintes procedimentos que compõem o desenvolvimento da aplicação em questão:

- Realização de análise comparativa das principais aplicações do mesmo segmento para entender pontos positivos e negativos;
- Levantamentos dos requisitos visando compreender e identificar as necessidades da comunidade universitária;
- Prototipação das interfaces dos usuários (*web* e *mobile*);
- Modelagem do banco de dados para armazenar os dados dos usuários da aplicação.

A partir do levantamento das informações e documentos citados anteriormente, foi possível iniciar o desenvolvimento da *API* de comunicação e as interfaces para visualização das informações e interação dos usuários.

### 1.3 JUSTIFICATIVA

Atualmente as instituições públicas e privadas buscam um melhor aproveitamento dos recursos, cortando gastos e investindo no que é realmente necessário. Com a popularização da tecnologia e o seu uso para situações corriqueiras, notou-se uma forte tendência mundial na adoção de serviços com vertente cívica, visando incluir os cidadãos a participar ativamente da melhoria da sociedade. Esses serviços cívicos não são novidade, no entanto com a popularização dos dispositivos móveis eles estão sendo cada vez mais usados ao redor do mundo.

About 34 municipalities across the world have Open 311 server endpoints (27 of which are in the U.S.) (as per Open 311 Wiki, visited on August 18, 2014). However, there are many private sectors firms specializing in Gov 2.0 solutions that provide 311 services to different local government entities (cities, counties, and other special districts). See ClickFix, for example, server nearly 290 such units; CitySourced features over 100 government entities; PublicStuff claims presence in over 200 cities (Gailbraith, 2013). [...] (JULNES; GIBSON, 2015)

No relatório de gestão (RELATÓRIO... , 2019) emitido pela UFSC do ano de 2019 apresentou dados referente as solicitações atendidas pelo Departamento de Manutenção Predial e Infraestrutura (DMPI), o relatório informa que 93,66% das 8601 solicitações foram atendidas, os dados são expressivos e ressaltam a importância de um sistema para a gestão das solicitações e que possibilite a interação com a comunidade universitária. Uma ação criada pela Coordenadoria de Gestão Ambiental (CGA) em 2022, na mesma Universidade, visando a redução do custo nas contas de consumo de água, distribuiu adesivos colados pelas estruturas da universidade convidando a comunidade a reportar problemas relacionados vazamento de água através do aplicativo de mensagem *Whatsapp* o que reforça a importância da participação de todos para a construção de um ambiente melhor pois quanto menor o tempo para a resolução do problema, maior será a quantidade de recurso economizado pela instituição.

Considerando o cenário atual e a necessidade exposta anteriormente sobre a manutenção dos espaços que ficaram ociosos durante a pandemia, pensou-se em um sistema interno para universidades onde a comunidade poderá reportar problemas no campus e a Universidade terá essas informações de forma centralizadas, tornando mais fácil o processo de resolução.

A proposta busca também proporcionar uma maior participação da comunidade acadêmica, que ficou afastada das Universidades por dois anos devido à pandemia de COVID, utilizando a participação cívica como modo de aproximar a Comunidade e a Universidade,

Citizen feedback systems are also accountability instruments. They build on the concept of Social Accountability, describing to mechanisms that rely on civic engagement for keeping power-holders accountable.(CARMEN *et al.*, 2004)

Tendo como inspiração as aplicações existentes no mercado, analisadas no Capítulo 3 e políticas resultantes da padronização e estruturação dessas aplicativos, a aplicação permite a criação de relatórios de incidentes não urgentes em uma Universidade, tais como: luzes danificadas, equipamentos com defeito, móveis quebrados, lixo em local inadequado, carros estacionados em local proibido, entre outros. Além disso, na interface voltada para a Universidade é possível gerenciar as solicitações e marcá-las como em andamento ou já resolvidas.

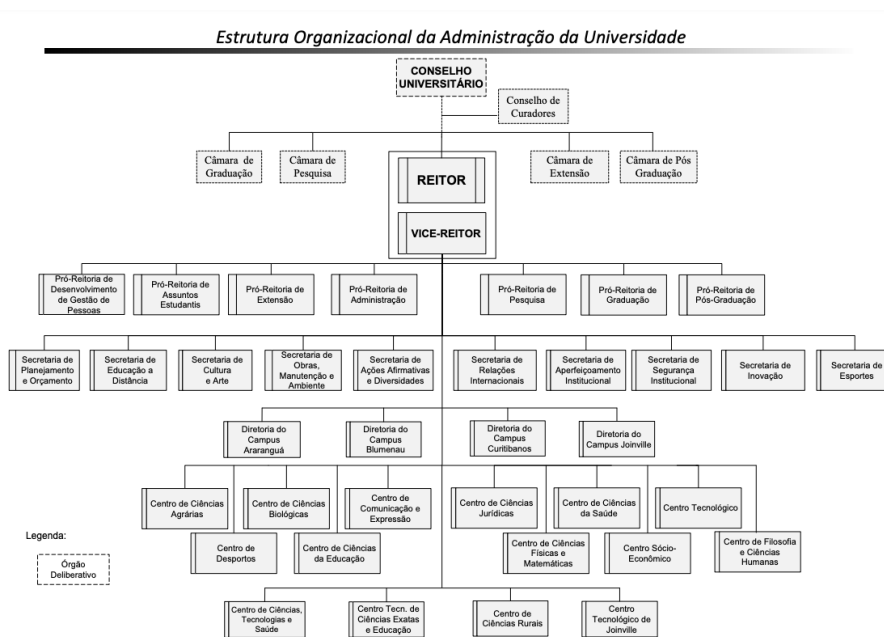
Acreditamos que a centralização das solicitações em um único sistema trará mais eficiência para as Universidades, que poderão determinar que setor ficará disponível para receber essas demandas e que setor será responsável por resolvê-las, tornando essa comunicação mais clara e direta além de proporcionar grande colaboração da comunidade acadêmica.



## 2 ANÁLISE DA ORGANIZAÇÃO DAS UNIVERSIDADES

Pensar a elaboração de um sistema que possibilite as universidades lidarem de forma simples, transparente e mais colaborativa com os problemas e sugestões de melhorias de suas estruturas físicas, equipamentos, entre outros, é um trabalho complexo uma vez que cada universidade tem seu próprio fluxo de informação e gestão de problemas. Desta forma, utilizou-se neste trabalho o Campus Reitor João David Ferreira Lima como referência. A Universidade Federal de Santa Catarina conta com cinco *campi* (Araranguá, Blumenau, Curitibanos, Florianópolis e Joinville), o que requer uma grande estrutura de gerenciamento para a garantia de um bom funcionamento e uma gestão que possibilite a execução e oferta de ensino, pesquisa e extensão. A Figura 1 exemplifica o tamanho e a complexidade da estrutura organizacional da universidade, neste estrutura destaca-se a divisão de centros que são pontos onde se concentram as solicitações de manutenção.

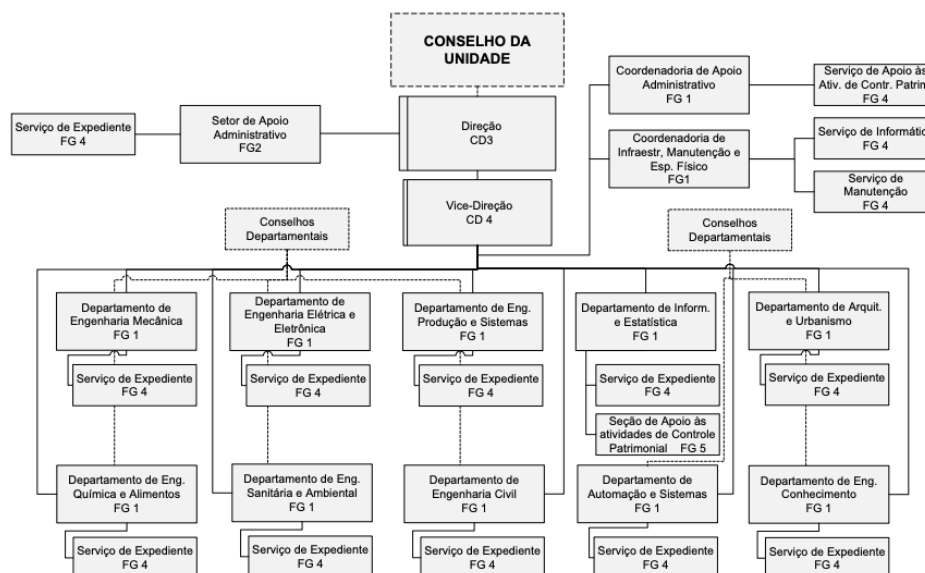
Figura 1 – Estrutura Organizacional da Administração da Universidade



Fonte: Printscreen da página <https://arquivos.ufsc.br/d/6560ff7645/files/?p=/Reitoria/Estrutura%20Geral.pdf>, feito em 05/02/2022.

Os centros por sua vez são compostos por departamentos, como por exemplo o Departamento de Informática e Estatística (INE) que pertence ao Centro Tecnológico (CTC), como apresentado na Figura 2.

Figura 2 – Estrutura Organizacional do Centro Tecnológico

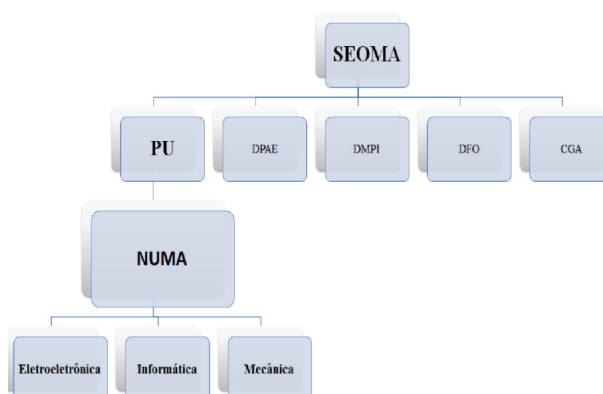
Estrutura Organizacional do Centro Tecnológico - Departamentos

Fonte: Printscreen da página <https://arquivos.ufsc.br/d/6560ff7645/files/?p=/Campus%20de%20%20Florian%C3%B3polis/CTC.pdf>, feito em 05/02/2022.

No caso da UFSC o gerenciamento da manutenção é de responsabilidade da SEOMA que tem como missão "contribuir no planejamento e aperfeiçoamento do espaço físico e ambiental da Universidade através de uma gestão de obras e manutenção sustentável, com transparência e eficiência em seus resultados"<sup>1</sup>, ou seja é ela quem atua quando os chamados de manutenção são solicitados pelos centros e/ou departamentos. A Figura 3 apresenta o organograma desta secretaria, onde são apresentadas algumas estruturas que exemplificam a complexidade do gerenciamento desta informação. As manutenções, dependendo de sua natureza ou origem, podem ser atendidas pela Prefeitura Universitária (PU), através do NUMA ou pelo DMPI.

<sup>1</sup><https://seoma.ufsc.br/>

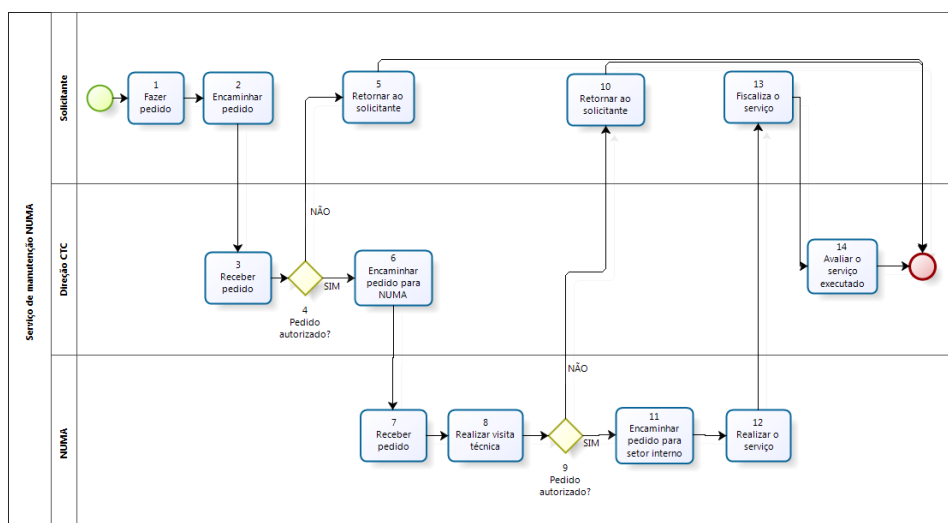
Figura 3 – Organograma SEOMA



Fonte: Printscreen da página <https://numa.paginas.ufsc.br/organograma/>, feito em 05/02/2022.

Para cada solicitação o solicitante deve enviar um requerimento junto ao respectivo centro (por exemplo CTC) e se aprovado pela direção do centro é encaminhado órgão responsável pela manutenção onde será dado prosseguimento ao processo de verificação do problema e se possível a resolução do problema como apresentado na Figura 4 que ilustra processo de trabalho para abertura de chamado de manutenção em Equipamentos Eletro-Eletrônico, Laboratório e Óticos através do NUMA.

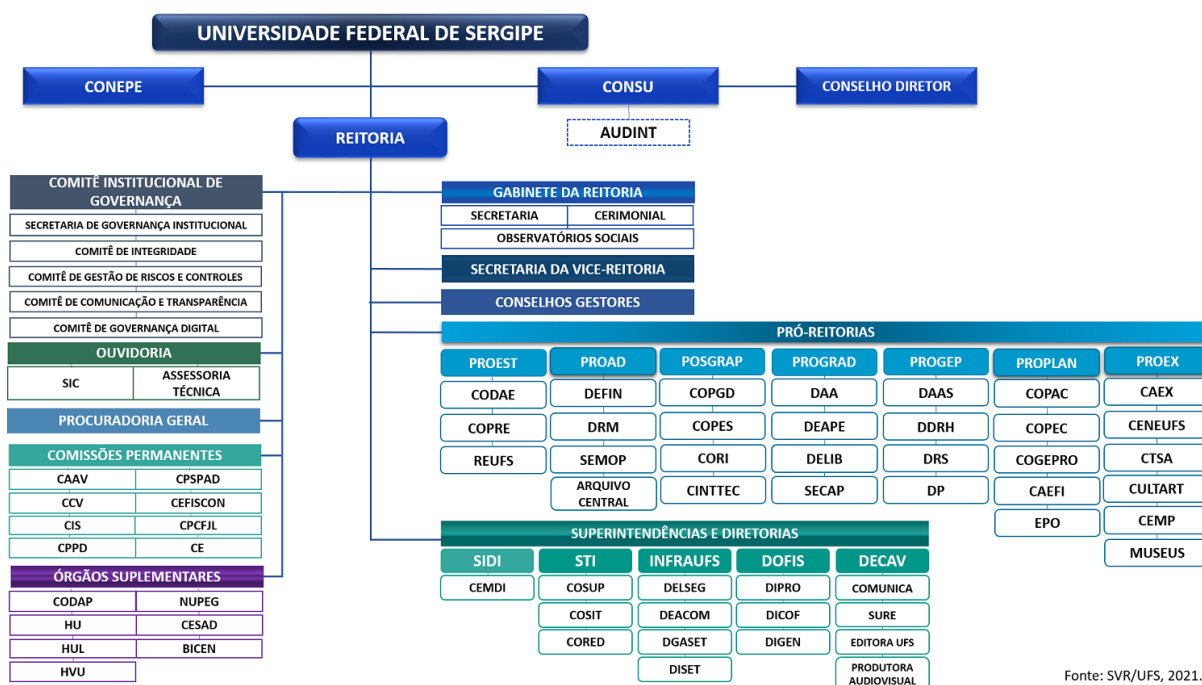
Figura 4 – Processos de trabalhos - NUMA



Fonte: Printscreen da página <https://portal.ctc.ufsc.br/equipamentos-eletro-eletronicos/>, feito em 05/02/2022.

Apesar de diferentes estruturas, outras universidades brasileiras também apresentam grandes estruturas organizacionais que formam grandes fluxos de informação entre quem solicita a resolução de um problema e quem de fato irá solucionar. A Figura 5 refere-se ao organograma da Universidade Federal do Sergipe (UFS), nesta universidade existe Superintendência de Serviços de Infraestrutura (INFRAUFS) que por sua vez tem subunidades como: o Departamento de Manutenção (DEMAN) que é "responsável pela execução dos serviços de manutenção de máquinas, aparelhos, equipamentos, utensílios, móveis e imóveis da UFS"<sup>2</sup> e a Divisão de Manutenção de Máquinas e Equipamentos (DIMEQ)<sup>3</sup>.

Figura 5 – Organograma UFS



Fonte: SVR/UFS, 2021.

Fonte: Printscreen da página <https://www.ufs.br/conteudo/56192-organograma-ufs>, feito em 05/02/2022.

A complexidade das organizações dificultam a interação entre a comunidade universitária e os setores responsáveis pelas manutenções, por outro lado se torna inviável a reestruturação das organizações a fim de suportarem um novo sistema, desenvolvido nesse trabalho. Desta forma o sistema elaborado visa a implementação de uma via secundária de comunicação, onde após o recebimento de de um pedido de manutenção ou sugestão de melhoria a Universidade tenha autonomia para tratar da melhor forma.

<sup>2</sup><https://infra.ufs.br/pagina/21508-departamento-de-manutencao>

<sup>3</sup><https://infra.ufs.br/pagina/21509-divisao-de-manutencao-de-maquinas-e-equipament>

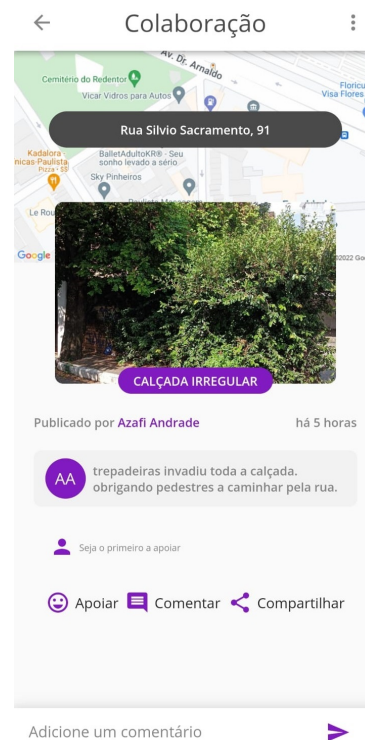
### 3 ANÁLISE DOS SISTEMAS SIMILARES

Durante essa etapa de pesquisa foram mapeadas quatro soluções existentes no mercado, duas brasileiras, uma americana e uma britânica. Abaixo estão as análises das principais funções de cada uma a fim de perceber o que têm em comum e o que diferem. São elas: *Colab*, *ParticipAct*, *Open311* e *FixMyStreet*.

#### 3.1 COLAB

O *Colab* (<https://www.colab.re/>) é uma plataforma que busca que as pessoas exerçam cidadania através das tecnologias. Eles oferecem um aplicativo gratuito, onde o usuário faz o *login*, compartilha sua localização e pode reportar problemas relacionados à sua cidade, podendo adicionar foto, descrição, endereço e título. Após a postagem, o usuário receberá um *email* informando se a Prefeitura está utilizando o sistema para receber a solicitação. Em caso positivo, a Prefeitura irá responder a solicitação e encaminhará para o setor responsável resolver. O aplicativo permite que outros cidadãos apoiem, comentem e compartilhem as publicações de outros usuários, funcionando com uma rede social.

Figura 6 – Solicitação de melhoria no aplicativo Colab



Fonte: Printscreen de uma solicitação feita no aplicativo Colab, feito em 06/02/2022.

Além disso, o aplicativo traz uma experiência gamificada, similar a um jogo em que o usuário vai evoluindo conforme vai avançando nas missões propostas pelo aplicativo. Os usuários ganham selos ao engajarem com a plataforma, tornando a experiência interessante e envolvente para o usuário.

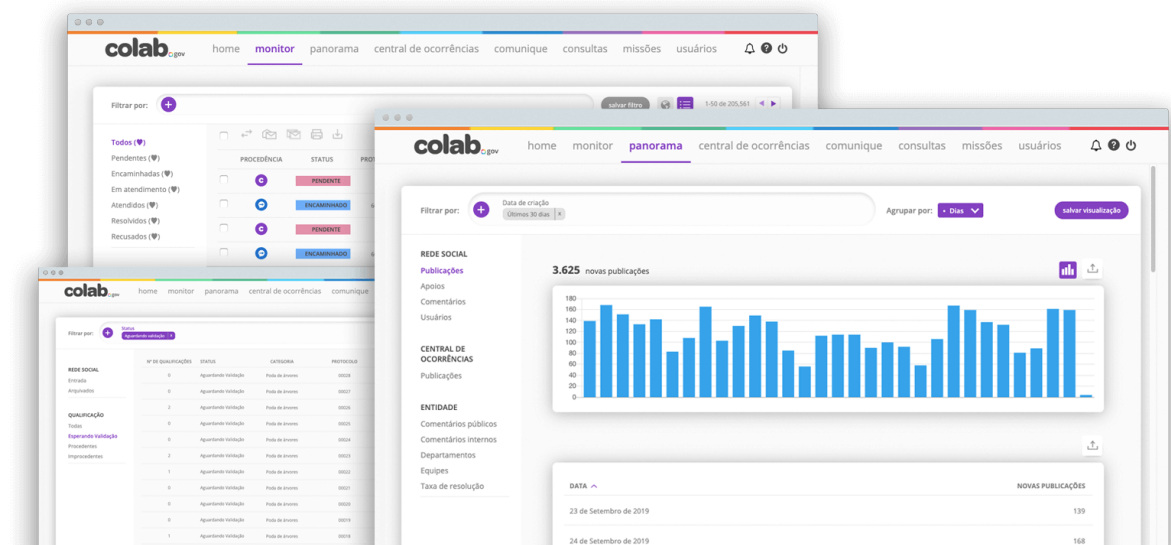
Figura 7 – Aba de conquistas no aplicativo Colab



Fonte: Printscreen da aba de conquistas feita no aplicativo Colab, feito em 06/02/2022.

Há também a aplicação destinada aos órgãos governamentais (<https://www.colab.re/gov>) que permite a gestão das demandas recebidas através do módulo "Monitor" e o acesso aos relatórios de dados e indicadores através de um módulo denominado "Panorama", o que possibilita uma melhor tomada decisão por parte das instituições uma vez que essas decisões podem ser baseadas em dados, conforme apresentado nas imagens a seguir:

Figura 8 – Módulo Monitor



Fonte: Printscreen retirada do site: <https://www.colab.re/gov>, feito em 06/02/2022.

Figura 9 – Módulo Panorama

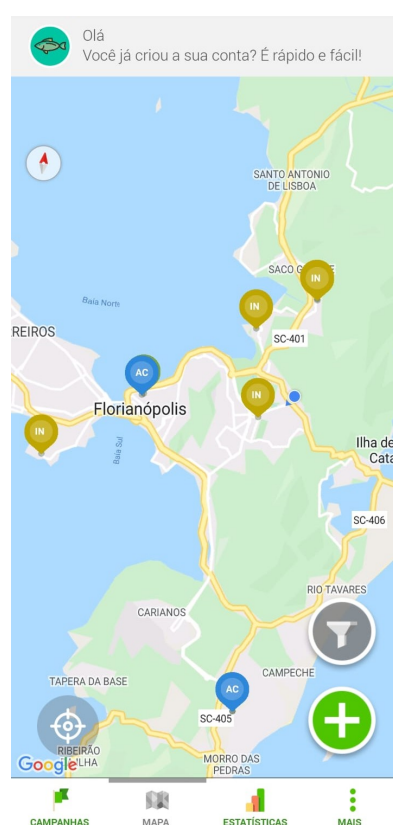


Fonte: Printscreen retirada do site: <https://www.colab.re/gov>, feito em 06/02/2022.

## 3.2 PARTICIPACT

O ParticipAct Brasil (<http://www.participact.com.br/>) é um projeto desenvolvido pela Universidade do Estado de Santa Catarina (UDESC) em parceria com a Universidade Federal de Santa Catarina (UFSC) e a Universidade de Bologna (UNIBO) que tem como principal objetivo a criação de uma grande estrutura de banco de dados a partir das informações das tecnologias da informação e da comunicação, com o intuito de melhorar a gestão de cidades inteligentes. O aplicativo permite que o usuário envie uma solicitação e adicione título, descrição, foto, áudio, vídeo e localização. Vemos a inclusão do envio de vídeo como uma ferramenta interessante para ilustrar melhor o problema quando for necessário. E o envio de áudio como algo positivo em relação aos outros sistemas, visto que esse recurso auxilia na acessibilidade. O usuário consegue visualizar um mapa da cidade e ícones onde há solicitações, bem como as iniciais do problemas referido.

Figura 10 – Mapa com as solicitações no aplicativo ParticipAct.

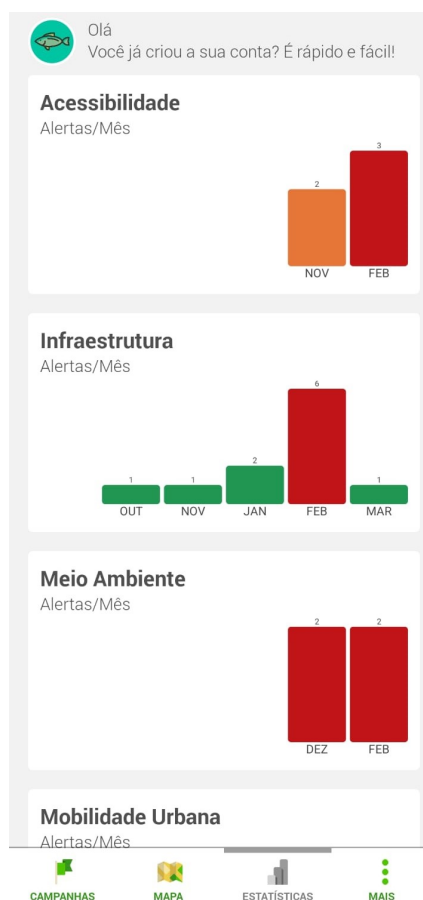


Fonte: Printscreen da aba de mapa feita no aplicativo ParticipAct, feito em 07/02/2022.

O aplicativo também disponibiliza gráficos classificando os tipos de solicitações feitas no sistema em relação aos meses que ocorreram esses reportes.



Figura 11 – Tabela com as solicitações feitas no aplicativo ParticipAct nos últimos meses.



Fonte: Printscreen da aba de estatísticas feita no aplicativo ParticipAct, feito em 07/02/2022.

### 3.3 OPEN311

Nos EUA, várias cidades têm linhas telefônicas governamentais não emergenciais, acessíveis pelo número de telefone 311. Como consequência, '311' passou a se referir a mais do que apenas uma linha telefônica - passou a significar todo o processo de manuseio de solicitações de serviço dos cidadãos em relação a questões não emergenciais, como lixo em local inadequado, buracos nas ruas e vizinhos barulhentos. Diversas municipalidades e empresas passaram a implementar soluções para gerenciar as demandas que chegam por esse serviço telefônico. Entretanto, com o desenvolvimento deste segmento de software, identificou-se a necessidade de um padrão para facilitar a adoção de peças de software de fornecedores distintos. Assim, surgiu a iniciativa Open311, que propõe um protocolo padronizado, colaborativo e baseado em localização para registro e acompanhamento de problemas.

As tecnologias desenvolvidas em torno do protocolo Open311 usam a internet para transformar as antigas comunicações individuais síncronas feitas por um cidadão

a um *call center* em um ambiente interativo, assíncrono e de muitos-para-muitos. Ou seja, diversas pessoas podem trocar informações abertamente sobre um problema local identificado, fornecendo muito mais informações para aqueles que precisam responder ao problema identificado<sup>1</sup>. Ao tornar a informação pública, o sistema oferece transparência e automaticamente atribui responsabilidades. A transparência também garante que a voz de todos é ouvida e isso encoraja a participação.

Durante a etapa de pesquisas sobre soluções similares à pensada neste trabalho, foram encontradas diversas propostas ao redor do mundo que implementam os protocolos Open311, tais quais os sistemas desenvolvidos pela Connected Bits<sup>2</sup> e a FixMyStreet<sup>3</sup>. A Connected Bits possui diversas cidades do mundo como clientes, como: Dallas, Boston, Calgary, Austin, Edmonton, entre outras. A iniciativa de reporte de situações não emergenciais em Boston já existia a muitas décadas por meio de uma linha telefônica, mas percebeu-se que muitas pessoas não usavam esse serviço. Com o avanço da tecnologia e a ampliação dos aplicativos para celular, em 2010 o reporte de questões não emergenciais passou a ser feito por um aplicativo desenvolvido pela Connected Bits, a iniciativa teve muito sucesso e logo foi adotada pelos cidadãos de Boston através do sistema Bos:311<sup>4</sup>. Hoje o aplicativo continua em funcionamento após muitas melhorias e tem muitas solicitações por dia, sendo possível ler sobre a equipe que resolveu o problema quando o reporte está concluído e em alguns casos é possível ver fotos do processo.

The Mayor's Office of New Urban Mechanics (MONUM) in Boston, MA was set up in 2010 to encourage greater civic engagement in the city through the use of innovation and experimentation, because Mayor Thomas Menino felt that government was too bureaucratic and lacked a human face. MONUM's main aim was to improve the quality of Bostonians' lives by involving them in participatory civic engagement. It has created many different projects to improve city services for citizens and workers, such as the app BOS:311 to report local maintenance issues to Boston City Hall. MONUM has been widely successful and remained a feature of the Mayor's Office under the leadership of Mayor Marty Walsh. It currently works on a variety of projects across the city's departments, such as housing and education, and has even prompted other cities across America to adopt its approach. (BOSTON'S... , 2019)

Os sistemas pensados pela empresa Connected Bits possuem versão *web* e *mobile*, em ambas o usuário pode visualizar as solicitações feitos por outros usuários, filtrando-os em solicitações abertas (a resolver) e fechados (já resolvidos), além de tipo de serviço, como: estacionamento em local ilegal, buracos, calçadas, carro abandonado, entre outros. Os usuários também podem reportar problemas, para isso devem selecionar a categoria do problema, adicionar uma breve descrição, a geolocalização e

---

<sup>1</sup><https://www.open311.org/learn/>

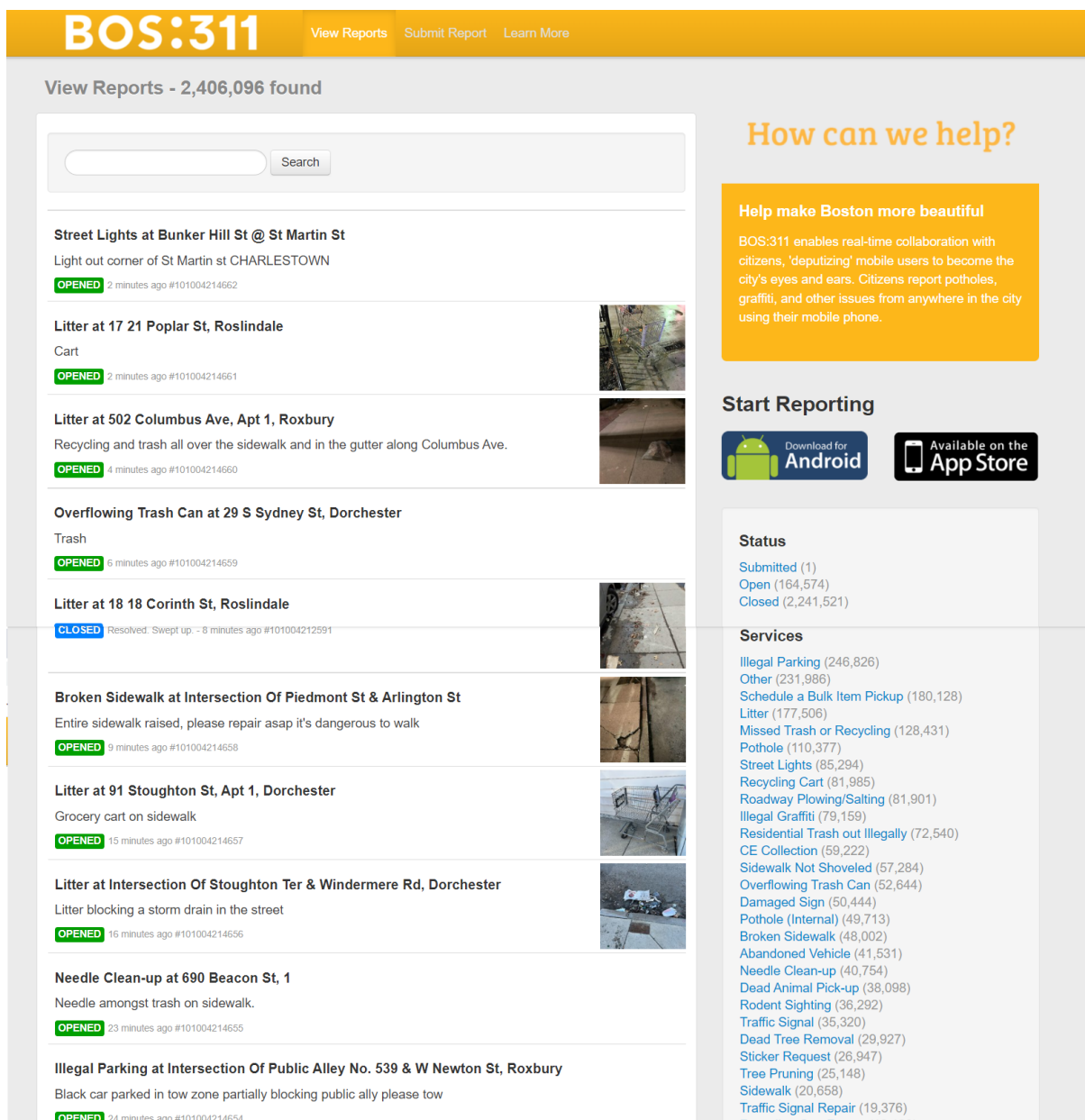
<sup>2</sup><http://grcity.spotreporters.com/#header>

<sup>3</sup><https://www.fixmystreet.com/>

<sup>4</sup><https://311.boston.gov/>

uma foto. Ao acessar cada reporte é possível ler o título, a descrição, a foto (se houver), a geolocalização do fato, o endereço e as coordenadas.

Figura 12 – Open311 da cidade de Boston



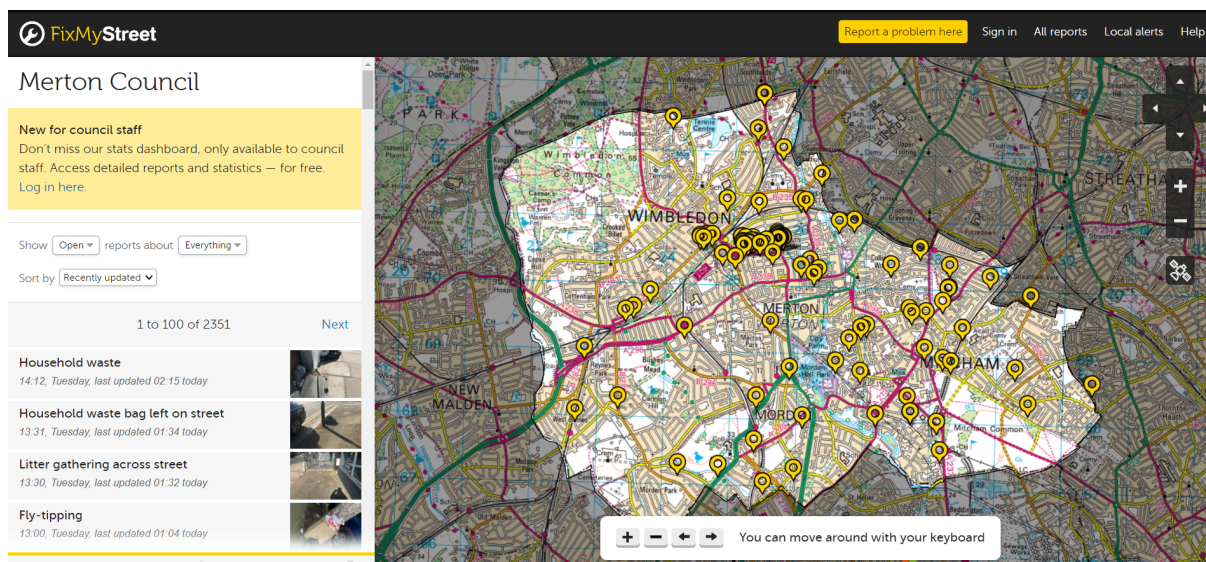
Fonte: Printscreen da página <https://311.boston.gov/> , feito em 08/02/2022.

### 3.4 FIXMYSTREET

A aplicação FixMyStreet é utilizada na Inglaterra, França, Austrália, e outros. As funcionalidades da aplicação são similares às do Bos:311, mas a interface é diferente. O site filtra as solicitações por "council", portanto ao digitar o council escolhido aparecerá um mapa do local e pequenas bandeiras indicando onde há reportes. no menu

lateral esquerdo pode-se ler do que se trata cada solicitação.

Figura 13 – Reports FixMyStreet do Council Merton

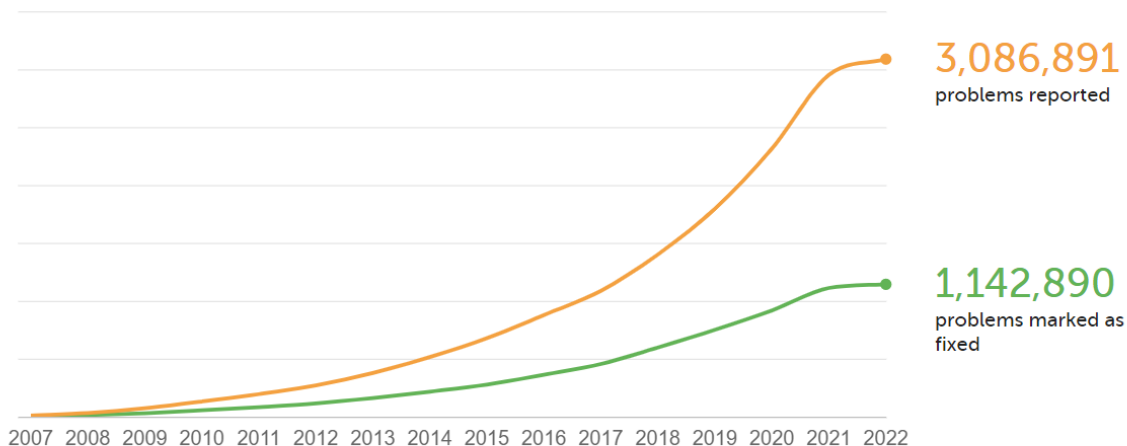


Fonte: Printscreen da página <https://www.fixmystreet.com/reports/Merton?zoom=13&lat=51.41083&lon=-0.18921> , feito em 08/02/2022.

No menu lateral é possível filtrar as solicitações por abertos, encerrados e resolvidos. Além disso, é possível filtrar por categoria de problema e por ordenação. Ao selecionar uma solicitação, a bandeira referente a ele no mapa fica maior e é possível ler o título, a descrição, data que foi postado, status e foto se houver. Também é possível postar uma atualização sobre aquela solicitação, mostrando como está a situação no momento em que você está passando por aquele local. O FixMyStreet disponibiliza gráficos da quantidade de solicitações feitas e quantas foram resolvidas desde que foi lançada e nos últimos sete dias, é possível filtrar essas ferramentas por council.

Figura 14 – Gráficos FixMyStreet de solicitações enviadas e resolvidas.

All time



Last 7 days



Show reports in your area

Pick your council

Fonte: Printscreen da página <https://www.fixmystreet.com/reports>, feito em 08/02/2022.

Além disso, ele categoriza os councils mais eficientes e as categorias de problemas mais reportadas pelos usuários. Esses dados ajudam cada council a entender os maiores problemas que possui e áreas que deve investir mais.

Figura 15 – Gráficos FixMyStreet de efetividade.

Top 5 responsive councils		Top 5 most used categories	
Average time between a problem being reported and being fixed, last 100 reports.		Number of problems reported in each category, in the last 7 days.	
Merton Council	1 day	Flytipping	1,241 reports
Hackney Council	2 days	Potholes	1,173 reports
Lincoln City Council	5 days	Pothole	1,101 reports
Peterborough City Council	8 days	Roads/highways	460 reports
Bromley Council	8 days	Pothole on road/cycleway	414 reports
Overall average	82 days	Other categories	10,741 reports

Fonte: Printscreen da página <https://www.fixmystreet.com/reports>, feito em 08/02/2022.

### 3.5 AVALIAÇÃO DAS FERRAMENTAS ESTUDADAS

Nesta análise foi possível identificar dois tipos de interface de design, a primeira é minimalista, concentra-se na comunicação entre o usuário e o sistema, proporcionando uma comunicação mais rápida na solicitação de problemas e melhorias. A gama de interações possíveis é, portanto, deliberadamente restringida; a escolha de categorias de serviço fixa. Como é o caso da Spot Reporters da Connected Bits <sup>5</sup>, da FixMyStreet e da ParticipAct.

O segundo tipo de interface se concentra em uma interação mais social, que proporciona mais interação entre os usuários e com a plataforma. apresentam muitos elementos familiares de outras plataformas de mídia social. Cidadãos e funcionários públicos são representados por meio de perfis de usuários e esses usuários têm a possibilidade de avaliar e comentar sobre problemas existentes em seus *feeds* de atividades.

Com essas informações pode-se perceber que as aplicações possuem muitas semelhanças, mas que cada uma contém suas particularidades. Utilizando esses modelos como inspiração para o presente trabalho foi acrescentada uma nova funcionalidade que não foi contemplada nas outras aplicações: o estado "em progresso". Esse estado serve para a Universidade notificar à comunidade acadêmica que aquela solicitação foi verificada e que está sendo tomada alguma providência para que seja resolvida o mais breve possível.

<sup>5</sup><https://connectedbits.com/reporters/>

Figura 16 – Tabela comparativa das funcionalidades das aplicações pesquisadas.

Nome da aplicação	Necessária autenticação?	Possível categorizar solicitações?	Possível filtrar solicitações?	Possível adicionar diversos tipos de mídia?	Possível interagir com solicitações de terceiros?	Fornecer dados estatísticos para o contratante do serviço?	Permite ao contratante marcar uma solicitação como "Em progresso"?
Colab	Sim, para enviar uma solicitação.	Sim, existem 68 categorias pré-definidas.	Não.	Apenas foto.	Sim, possível comentar, apoiar e compartilhar.	Sim.	Não.
ParticiAct	Sim, para enviar uma solicitação.	Sim, existem 5 categorias pré-definidas.	Sim, por data e localização.	Sim, fotos, vídeos e áudios.	Sim, possível compartilhar e denunciar.	Sim.	Não.
Open311	Sim, para enviar uma solicitação.	Sim, existem 8 categorias pré-definidas e 22 subcategorias.	Sim, por status e tipo de solicitação.	Apenas foto.	Não.	Sim.	Não.
FixMyStreet	Não.	Sim, existem 20 categorias pré-definidas.	Sim, por data, status e tipo de solicitação.	Apenas foto.	Sim, possível enviar uma atualização e marcar como corrigida.	Sim.	Não.

Fonte: Autor.

## 4 DESENVOLVIMENTO DA APLICAÇÃO

### 4.1 PESQUISA E LEVANTAMENTO DE REQUISITOS

A pesquisa e o levantamento de requisitos foram realizadas com base na análise das plataformas de participação colaborativa (Open311, FixMyStreet, Colab e ParticipApt) e na análise das estruturas organizacionais das Universidades. Desta forma foram levantados os requisitos com base nos sistemas já existentes e adaptar à realidade da Universidade. Os sistemas podem ser separados em duas partes: o sistema voltado para o uso da comunidade acadêmica e o sistema voltado para a gestão da Universidade que será utilizado pela instituição para gerenciar as solicitações.

#### 4.1.1 Requisitos não funcionais

Os requisitos não funcionais são:

- Os sistemas devem ser compatíveis com os navegadores Chrome e Mozilla Firefox;
- O sistema de uso da comunidade deve ser um Progressive Web App (PWA)<sup>1</sup>;
- O sistema de uso da comunidade deve ser responsivo;
- Ambos sistemas devem utilizar um sistema de autenticação;

#### 4.1.2 Requisitos funcionais

Os requisitos funcionais são:

- Deve ser possível ao usuário da comunidade cadastrar uma solicitação de manutenção;
- Deve ser possível ao usuário da comunidade cadastrar uma solicitação de sugestão;
- Deve ser possível ao usuário da comunidade visualizar a lista de solicitações criadas por outros usuários;
- Deve ser possível ao usuário da comunidade visualizar a lista de solicitações criadas pelo próprio usuário;
- Deve ser possível ao usuário da comunidade votar positivamente ou negativamente referente a uma solicitação;

---

<sup>1</sup> PWA: É uma metodologia de desenvolvimento de aplicações *web* que proporciona o usuário uma experiência de uso similar a aplicações nativas quando acessada de dispositivos móveis (*tablets* e celulares) porém sem a necessidade de instalação de um aplicativo pois ela utiliza o próprio *browse* já instalado no celular.

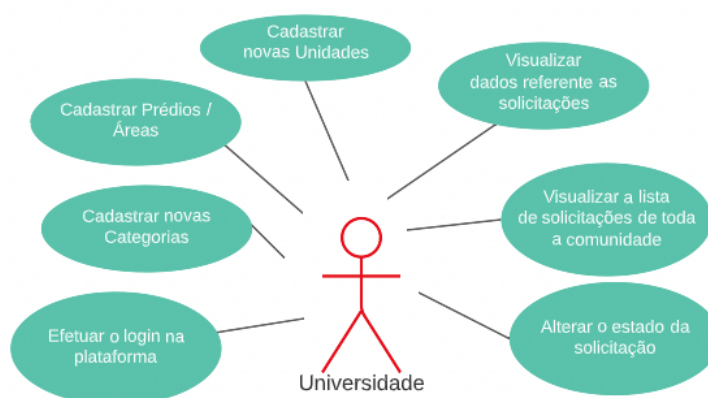


- Deve ser possível ao gestor da universidade visualizar a lista de solicitações dos usuários da comunidade;
- Deve ser possível ao gestor da universidade alterar o status de uma solicitação resolvida;
- Deve ser possível ao gestor da universidade cadastrar e excluir unidades;
- Deve ser possível ao gestor da universidade cadastrar e excluir prédio/área;
- Deve ser possível ao gestor da universidade cadastrar e excluir categorias;
- Deve ser possível ao gestor da universidade visualizar os dados referente a solicitações, unidades, prédios/áreas e categorias;
- Deve ser possível filtrar as solicitações por data, unidades e prédios/áreas;

## 4.2 CASO DE USO

Os diagramas de caso de uso descrevem e ilustram as principais funcionalidades do sistema do ponto de vista dos usuários (Universidade e comunidade). Desta forma entende-se por Universidade o usuário representante da Universidade que acessa o sistema com a finalidade de receber as demandas de solicitações de manutenção e/ou sugestões de melhoria, representadas pelo seguinte diagrama:

Figura 17 – Diagrama de caso de uso: Universidade.



Fonte: Autor.

De maneira análoga, temos o diagrama para o usuário comunidade, que pode ser representado por um aluno, por um servidor ou outro membro da comunidade universitária.

Figura 18 – Diagrama de caso de uso: Comunidade.



Fonte: Autor.

### 4.3 PROTOTIPAÇÃO DO SISTEMA

Inicialmente foi realizada a prototipação das interfaces do sistema através da plataforma Figma<sup>2</sup> a partir dos dados levantados anteriormente. A prototipação foi baseada no pacote de componentes Material UI<sup>3</sup>. Neste primeiro momento, priorizou-se as telas do sistema da Universidade a fim de ilustrar como ficariam a tela de solicitações, a de relatórios e a de cadastro de centros acadêmicos. Importante notar que esse foi um primeiro esboço e que foram acrescentadas algumas funcionalidades à versão final. Desta forma, a primeira elaboração das telas do sistema para a Universidades foram realizadas da seguinte forma:

---

<sup>2</sup><https://figma.com/>

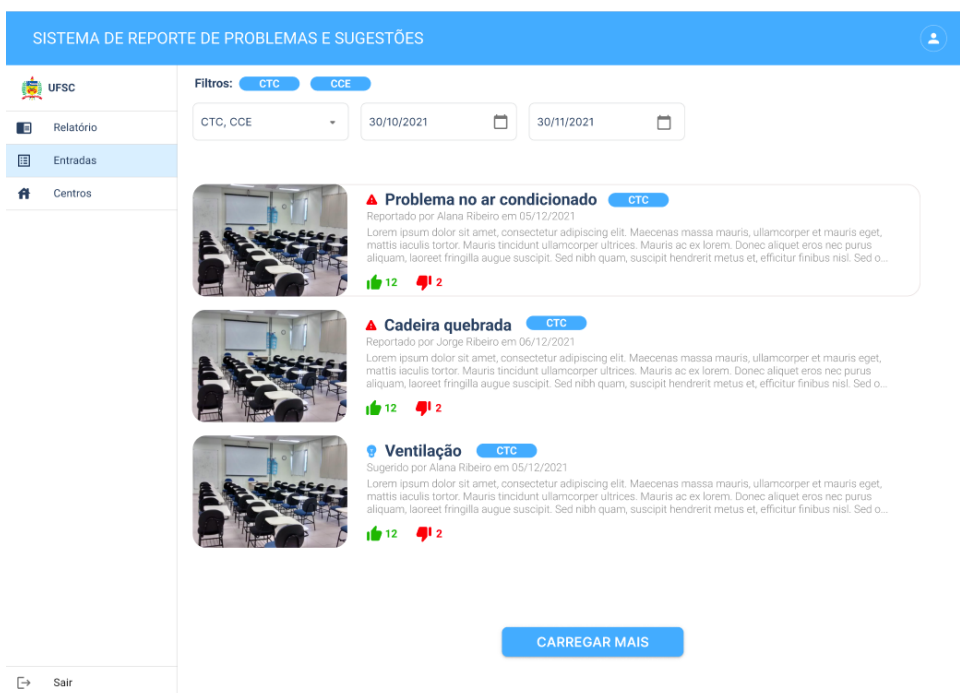
<sup>3</sup><https://mui.com/pt/>

Figura 19 – Tela de relatórios



Fonte: Autor.

Figura 20 – Tela de entradas (solicitações)



Fonte: Autor.

Figura 21 – Tela de cadastro de centros

UFSC	Centros	+ ADICIONAR CENTROS
Relatório	↓ Sigla	Nome
Entradas	CCE	Centro de Comunicação e Expressão
Centros	CTC	Centro Tecnológico
	CSE	Centro Socioeconômico
	CCE	Centro de Comunicação e Expressão
	CTC	Centro Tecnológico
	CSE	Centro Socioeconômico
	CCE	Centro de Comunicação e Expressão
	CTC	Centro Tecnológico
	CTC	Centro Tecnológico
	CSE	Centro Socioeconômico
	CSE	Centro Socioeconômico

Sair

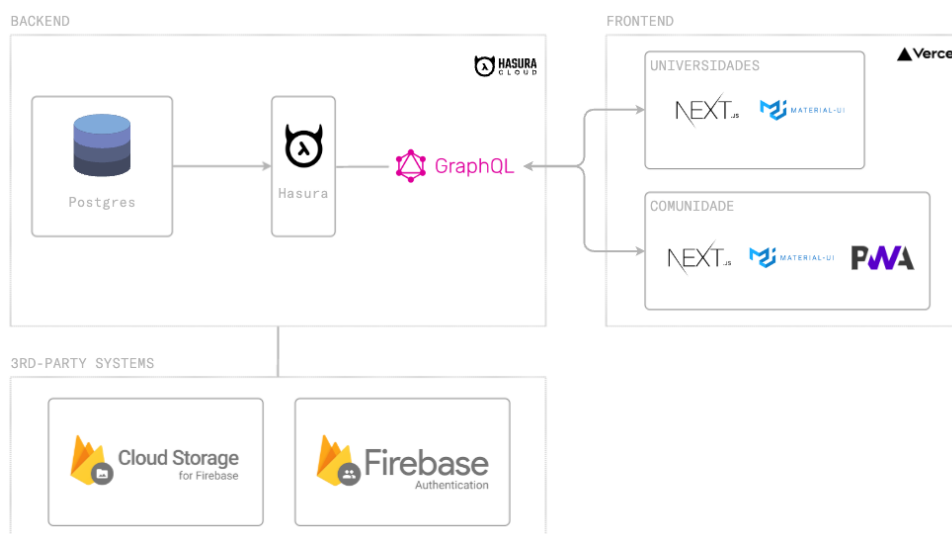
Linhas por página: 10 1-5 de 13

Fonte: Autor.

#### 4.4 TECNOLOGIAS UTILIZADAS NO SISTEMA

A arquitetura do sistema foi pensada para suportar os requerimentos apresentados nas seções anteriores e entregar ao usuário final uma estrutura confiável e que esteja de acordo com as melhores práticas do desenvolvimento *web*. Apesar de ambos os sistemas, Universidade e Comunidade, serem desenvolvidos para *web* é necessário que o sistema da Comunidade tenha uma boa usabilidade nos *smartphones* colocando a praticidade como chave para um maior engajamento. Desta forma, a Figura 22 apresenta a arquitetura pensada para o desenvolvimento do sistema:

Figura 22 – Arquitetura do sistema



Fonte: Autor.

#### 4.4.1 Backend

Atualmente grande partes das aplicações disponíveis na *World Wide Web* utilizam o modelo de *Restful* de comunicação entre banco de dados e interface do usuário.

REST defines a set of architectural principles by which you can design Web services that focus on a system's resources, including how resource states are addressed and transferred over HTTP by a wide range of clients written in different languages. (RODRIGUEZ, 2008, p. 1)

Essa estrutura é composta basicamente por um banco de dados e uma *API* de comunicação, neste trabalho devido ao grau de relacionamento entre as entidades mapeadas na modelagem do sistema, optou-se pela utilização de um banco de dados relacional PostgreSQL<sup>4</sup>. Outro fato que contribui para a escolha deste banco de dados é a facilidade de implementação de uma *API* de comunicação através do Hasura<sup>5</sup> que abstrai a necessidade de utilização de consultas *Standard Query Language (SQL)* para obtenção dos dados no banco. Hasura é uma tecnologia que permite a criação de interfaces de comunicação de maneira simples, através da utilização da linguagem de consulta GraphQL. Esta linguagem de consulta permite que o desenvolvedor defina no lado do cliente o que será retornando pela *API* em cada instância de comunicação.

<sup>4</sup><https://www.postgresql.org/>

<sup>5</sup><https://hasura.io/>

#### 4.4.2 Frontend

Ambas as aplicações, Comunidade e Universidade, serão desenvolvidas utilizando *Javascript* devido a sua grande popularidade nos últimos anos. Em 2021 a plataforma Statista<sup>6</sup> realizou uma pesquisa com aproximadamente 83.000 desenvolvedores a fim de identificar quais as linguagens mais utilizadas para o desenvolvimento *web*, esta pesquisa revelou que 64.96% dos entrevistados utilizavam *Javascript* e 56.07% utilizam *HyperText Markup Language (HTML)* e *Cascading Style Sheets (CSS)*, sendo essas respectivamente a primeira e segunda colocadas no *ranking*. Desta forma escolhemos a utilização do *framework Next.js* que foi desenvolvido para utilizar a mais popular biblioteca *Javascript* de desenvolvimento denominada *React*<sup>7</sup> criada pelo Facebook em 2013. Outra tecnologia importante para o desenvolvimento das interfaces foi a utilização de uma biblioteca de componentes (*Material UI*) que auxilia na padronização dos elementos de interface, resultando em uma aplicação padronizada.

O desenvolvimento do sistema da Comunidade foi executado utilizando um conceito denominado *Progressive web apps (PWA)*, que prevê um padrão de desenvolvimento que possibilita as plataformas *web* funcionarem de forma muito semelhante aos aplicativos de dispositivos móveis, ou seja, apesar da execução ser através dos navegadores a apresentação se dá através do modo *FullScreen*, com possibilidade de adição de uma tela de apresentação (tela com o logotipo da empresa) antes abrir o sistema de fato, ícone como atalho para abertura direta do sistema, funcionamento *offline*, entre outras características que tornam essa aplicação muito similar a um aplicativo nativo. Este conceito se baseia na definição de um arquivo chamado *manifest* que contém diversas informações sobre a aplicação, este arquivo define diversas informações importantes como nome, abreviação do nome da aplicação, cores, ícones e modo de abertura. Outro ponto que se destaca é utilização de *cache* através do componente *Service Worker* presente no *Javascript*, com ele é possível ter a aplicação disponível mesmo no modo *offline*, armazenando as informações no *cache* do navegador.

### 4.5 SISTEMA

#### 4.5.1 API e Banco de dados

Esta seção apresenta o desenvolvimento do sistema: como o desenvolvimento do banco de dados, a *API* de comunicação e a interface do usuário. Na fase de prototipação e de levantamento de requisitos elencou-se cada ponto que o sistema deveria possuir possibilitando o desenvolvimento da *API* e do banco de dados. Como citado anteriormente, a utilização do *Hasura* facilita esta implementação, uma vez que é pos-

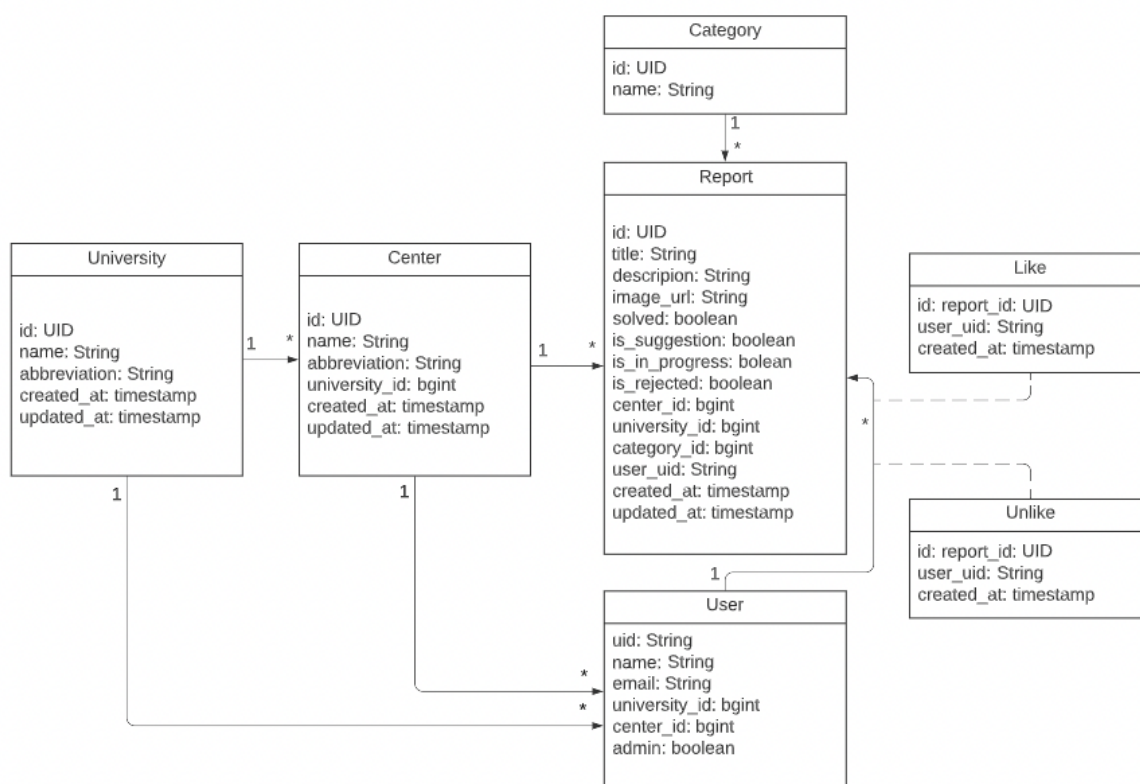
---

<sup>6</sup><https://www.statista.com/statistics/793628/worldwide-developer-survey-most-used-languages/>

<sup>7</sup><https://reactjs.org/>

sível realizar a configuração de todo o banco e inclusão das tabelas bem como suas relações através da interface do console que será abortado posteriormente. Desta forma a primeira etapa foi elaborar o diagrama *Unified Modeling Language (UML)* como representado na figura a seguir:

Figura 23 – Diagrama UML



Fonte: Autor.

O ponto central deste diagrama é a classe *Report* que se refere a uma solicitação, que pode ser de dois tipos: sugestão ou um pedido de manutenção, representado no diagrama pela propriedade *is\_suggestion*. Uma solicitação também contém a referência de um usuário (*user\_uid*), de um centro acadêmico (*center\_id*), da categoria (*category\_id*) e de um campus (*university\_id*). Além disso há dados que descrevem a solicitação como: título (*title*), descrição (*description*), a *url* da imagem (*image\_url*), data de criação (*created\_at*) e de atualização (*updated\_at*) e por fim o seu estado atual (solicitação criada ou resolvida) é representada pela propriedade *solved*.

Para a classe *User* que contém as informações referentes ao Usuário seja ele parte da Comunidade ou da Gestão da Universidade, nota-se a ausência de uma propriedade que faça referência à ligação do Usuário com o contexto real da Universidade, como por exemplo o número matrícula. Para a elaboração deste sistema idealizou-se

a integração com o Sistema de Autenticação da Universidade, porém para o desenvolvimento prático foi necessário utilizar outro sistema de autenticação a fim de simular a experiência de um usuário de forma a obter um sistema usual. Portanto, o sistema de autenticação escolhido para substituir o Sistema de Autenticação Centralizada da UFSC foi o *Firebase*<sup>8</sup> fornecido pela *Google* e utiliza o email como dado de identificação para acesso à plataforma juntamente com uma senha que deverá ser cadastrada. A escolha do *Firebase* deu-se pelo fato do serviço disponibilizar uma versão gratuita, a abundante quantidade de documentação disponível e também a possibilidade de utilização do serviço de hospedagem de arquivos (*Store*) que possibilita o *upload* de imagens para ilustrar a solicitação.

As classes *University* e *Center* são responsáveis por armazenar as informações referentes as Unidades e Prédios/Áreas respectivamente. A classe *Category* é responsável por armazenar as informações referentes às categorias de solicitações e por fim as classes *Like* e *Unlike* referem-se às interações do Usuário com as solicitações.

A partir das informações de cada classe foi possível iniciar a inclusão de tabelas no console Hasura, que automaticamente gera a *API* de comunicação para obtenção dos dados através da linguagem de consulta GraphQL, uma vez que ao iniciar o console pela primeira vez é realizada a criação e configuração do banco de dados PostgreSQL hospedado na plataforma de nuvem chamada Heroku<sup>9</sup>. Abaixo segue uma imagem que ilustra a inserção de uma tabela no console Hasura:

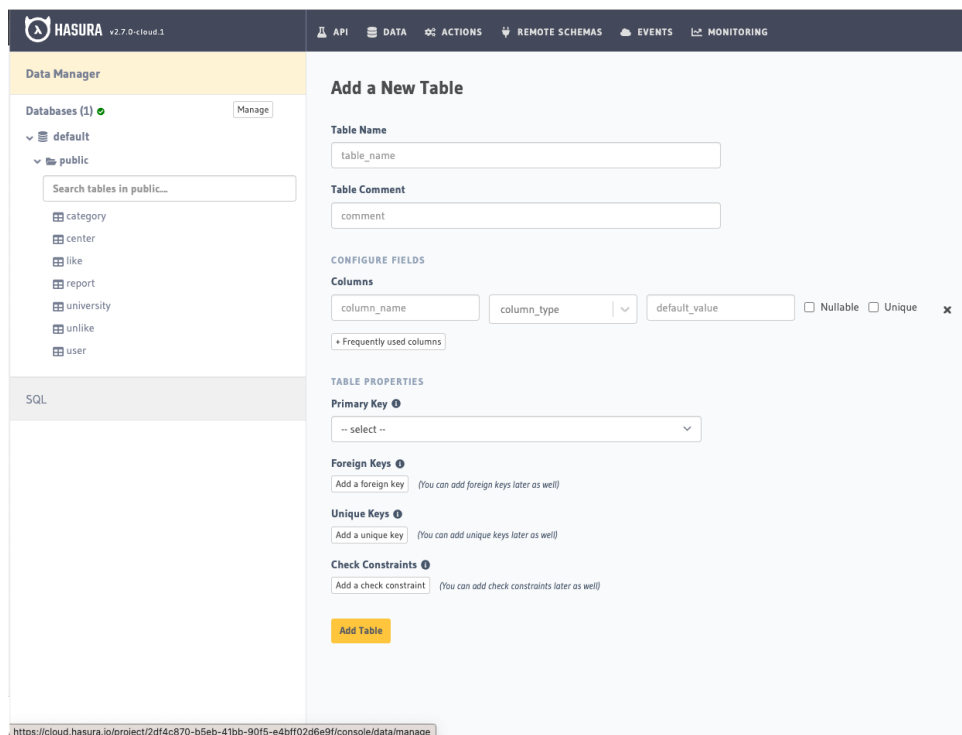
---

<sup>8</sup><https://firebase.google.com/>

<sup>9</sup><https://www.heroku.com/>



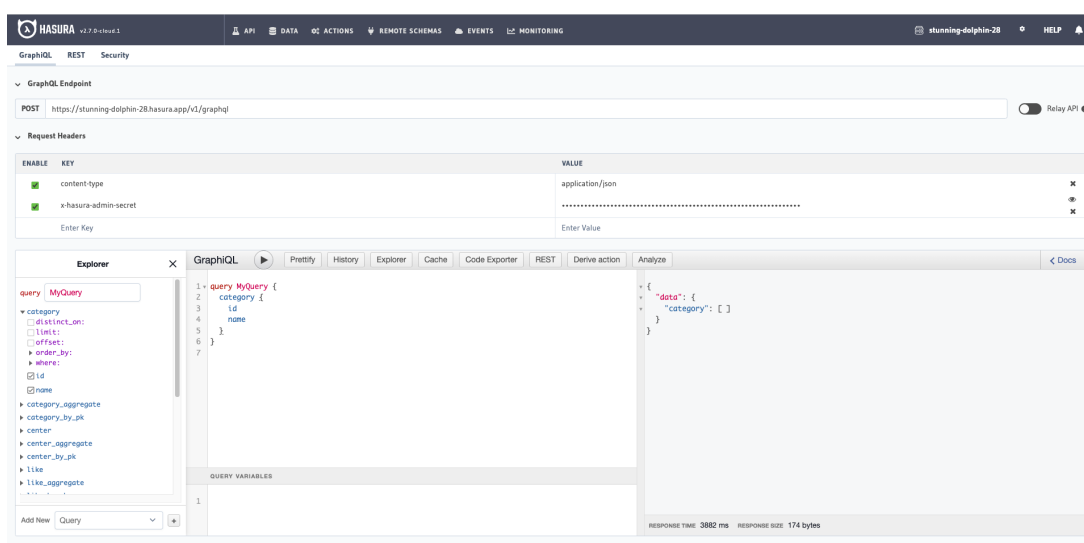
Figura 24 – Hasura Console - Criação de tabela



Fonte: Autor.

Após a inserção das tabelas pode-se utilizar o *Explorer* no console para testar o funcionamento da API através da linguagem de consulta GraphQL, como ilustrado na imagem a seguir:

Figura 25 – Hasura Console - Consulta GraphQL



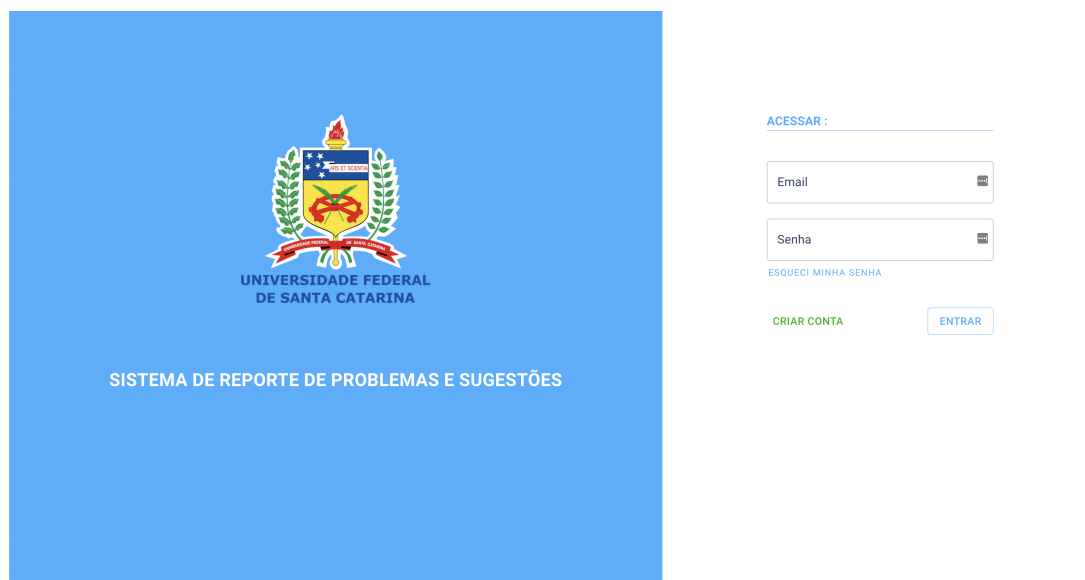
Fonte: Autor.

A linguagem de consulta se baseia em três diferentes tipos: *Query*, *Mutation* e *Subscription*. A *Query* é utilizado para obter dados do Banco de Dados, o *Mutation* é utilizado para alterar dados no Banco de Dados, como: adição, edição e remoção. O *Subscription* é utilizado para receber notificações de alterações no Banco de Dados através de um *WebSockets*, por exemplo. No exemplo anterior, definiu-se uma *Query* para obter todas as categorias cadastradas no Banco. Também é possível adicionar condições para filtrar o dados requeridos, bem como a ordenação dos dados.

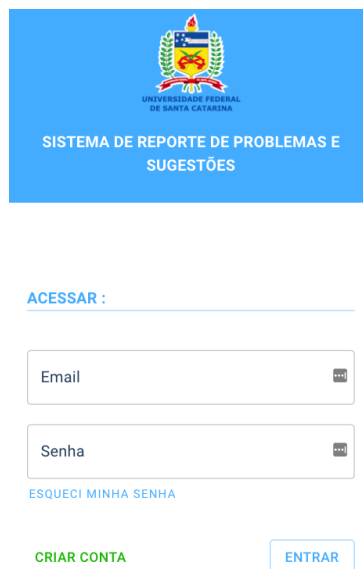
#### 4.5.2 Acesso ao sistema

A primeira parte do sistema que o Usuário tem contato ao acessar o sistema é a tela de acesso ao sistema que dispõe de três funcionalidades que são: acesso ao sistema, criação de conta de usuário e recuperação de senha. Essas funcionalidades são integradas com o Firebase que disponibiliza através de uma biblioteca, métodos que possibilitam realizar tais ações. Abaixo pode-se visualizar imagens que ilustram a tela inicial do sistema:

Figura 26 – Sistema - Acesso ao sistema versão *web*



Fonte: Autor.

Figura 27 – Sistema - Acesso ao sistema versão *mobile*

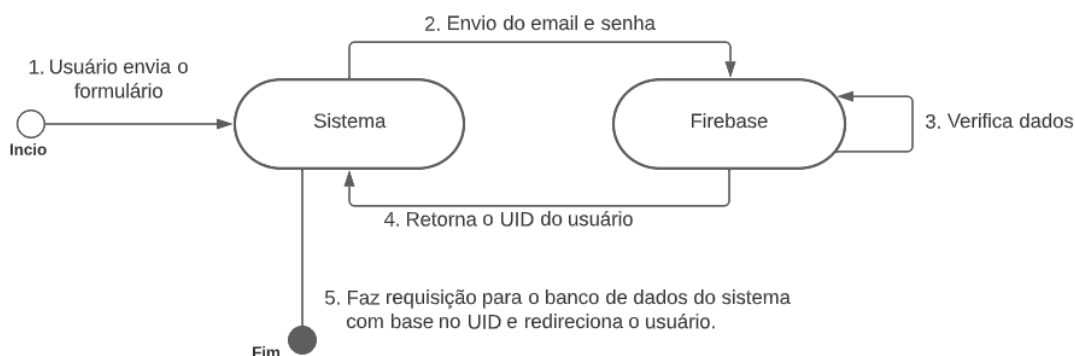
A tela de acesso ao sistema versão mobile apresenta o seguinte layout:

- Logo da Universidade Federal de Santa Catarina no topo central.
- Título: SISTEMA DE REPORTE DE PROBLEMAS E SUGESTÕES.
- Seção de acesso: ACESSAR :.
- Formulário de login com campos para Email e Senha, ambos com ícones de visibilidade (olho).
- Link de recuperação de senha: ESQUECI MINHA SENHA.
- Botões de ação: CRIAR CONTA (em verde) e ENTRAR (em azul).

Fonte: Autor.

Nesta tela há a opção de acesso ao sistema através dos dados de *email* e senha do usuário. Ao inserir os dados de acesso, o sistema envia os dados do formulário para o *Firebase* que realiza a validação dos dados. Caso seja válido, retorna uma resposta com os dados do usuário como o *UID* que é a identificação do usuário no Banco de Dados. Em seguida é feita uma requisição para o Banco a fim de obter os dados do Usuário e persistir esses dados na aplicação e no navegador através dos *cookies*. Isso permite que o Usuário permaneça com acesso ativo mesmo após atualizar a página do navegador ou até mesmo fechá-la. A figura a seguir ilustra o fluxo de informação que permite o acesso do Usuário à plataforma.

Figura 28 – Fluxo de informação - Acesso ao sistema

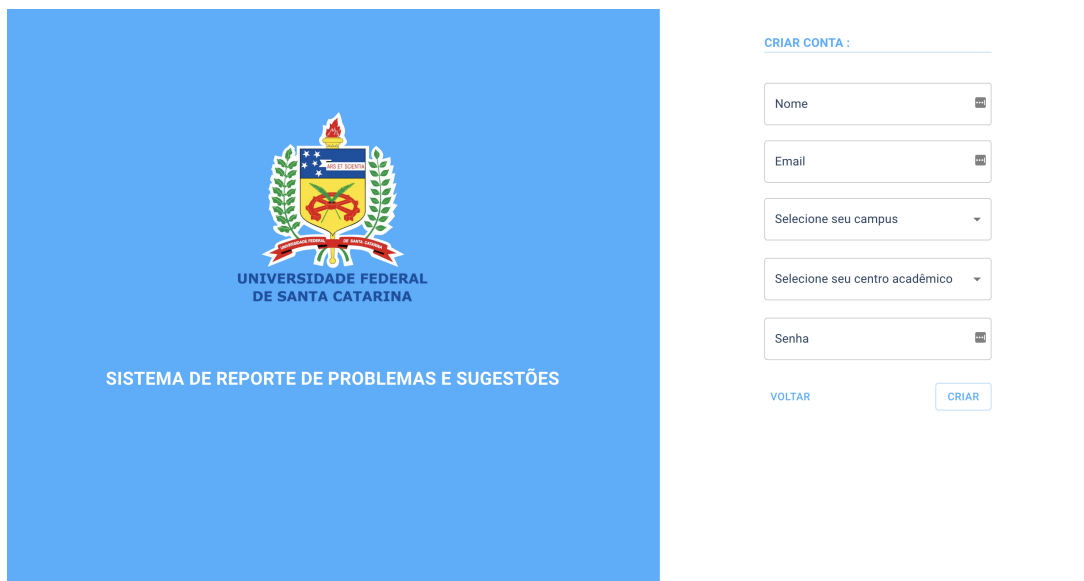


Fonte: Autor.

Após a validação, o Usuário será redirecionado com base no tipo de conta, através da propriedade *admin* definido na classe *User* do diagrama apresentado na Figura 23. Caso a conta do Usuário tenha a propriedade *admin* verificada como verdadeira, ele será redirecionado para o sistema voltado à Universidade, caso contrário será redirecionado para o sistema da Comunidade. Apesar de visualmente parecerem dois sistemas totalmente desacoplados, ambos foram implementadas e fazem parte do mesmo código de implementação, então na verdade temos somente um sistema com duas partes: Universidade e Comunidade. Na programação este conceito é conhecido como *monolito*, ou seja, uma aplicação desenvolvida neste modelo de arquitetura requer que todos os módulos seja executados em conjunto (DRAGONI *et al.*, 2017).

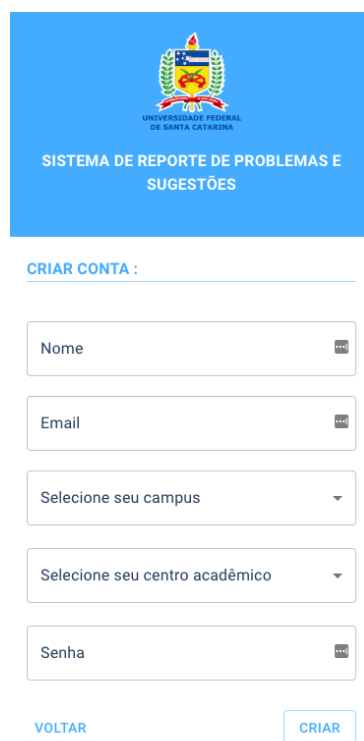
Outra funcionalidade presente na tela inicial do sistema é a criação de conta. Ela ocorre em duas etapas, iniciando com o envio do formulário, de forma análoga ao diagrama apresentado anteriormente: a primeira etapa consiste em inserir um *email* e senha. Com isso é realizada a inserção do usuário no serviço *Firebase Authentication* que irá retornar o *UID* do usuário. A segunda etapa consiste na inserção dos dados do Usuário no Banco de Dados com as outras informações do formulário, como nome, campus e centro acadêmico do Usuário. As telas da criação de conta podem ser visualizadas nas imagens abaixo. Outro ponto importante no desenvolvimento da aplicação foi a utilização de caixas de seleção para a inserção de alguns dados, a fim de prevenir a inserção de dados inválidos. Desta forma é necessário carregar todos os dados de Unidades e Prédios/Áreas do Banco de Dados no momento em que o Usuário acessa a tela inicial.

Figura 29 – Sistema - Criação de Conta versão *web*



Fonte: Autor.

Figura 30 – Sistema - Criação de Conta versão *mobile*

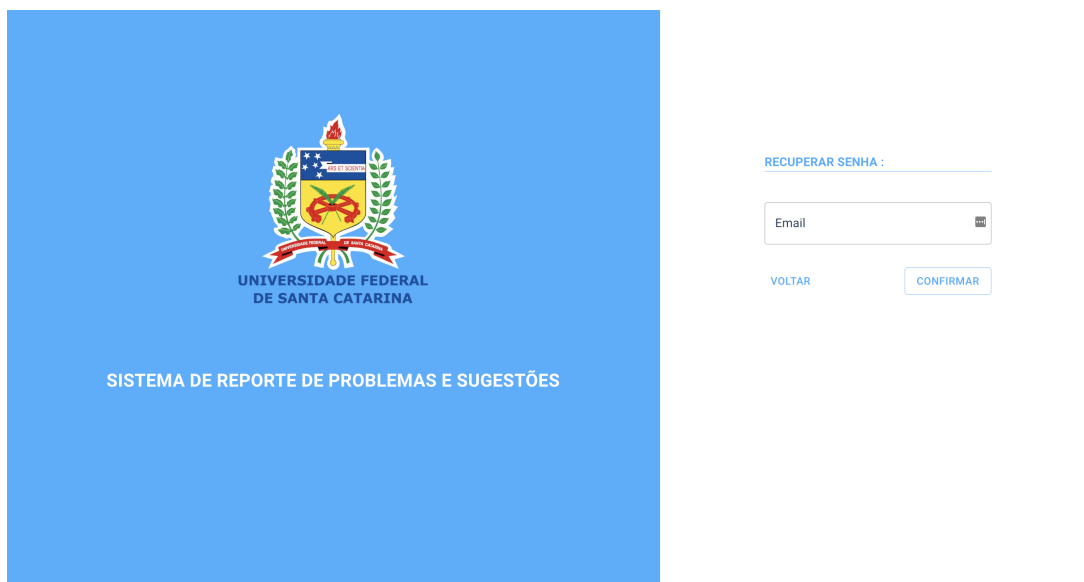


Fonte: Autor.

A última funcionalidade é a recuperação de senha, conforme ilustrado nas figuras abaixo. A partir da inserção do *email* do Usuário, o sistema *Firebase* envia um

*email* contendo um *link* que possibilita a criação de uma nova senha. O envio deste formulário é feito na página do próprio *Firebase*.

Figura 31 – Sistema - Recuperação de senha versão *web*



Fonte: Autor.

Figura 32 – Sistema - Recuperação de senha versão *mobile*



Fonte: Autor.

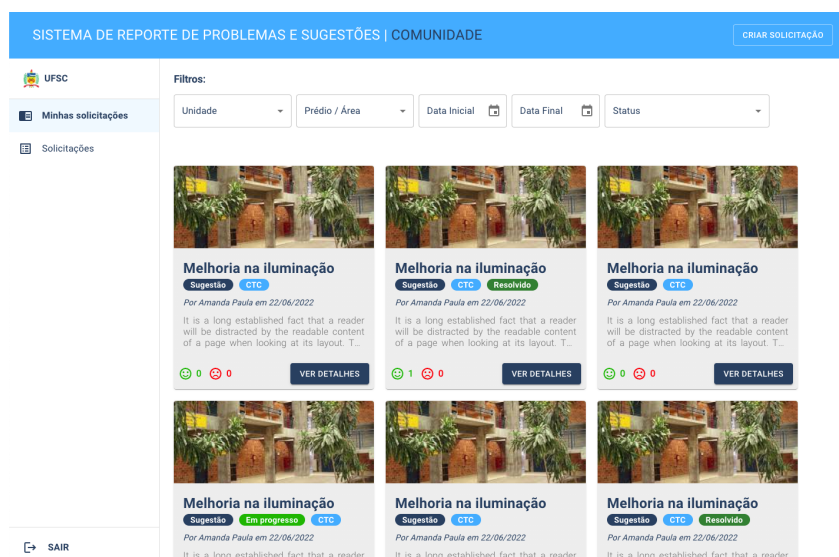
Estas três funcionalidades contém validações e tratamentos de erros nos cam-

pos do formulário como: validação de tamanho mínimo de senha, validação de *email*, verificação de *email* já cadastrado, verificação de senha, etc.

### 4.5.3 Comunidade

Uma vez que o Usuário entra com suas credenciais válidas ele será redirecionado para a parte protegida do sistema, ou seja o sistema interno que permite a criação e interação com as solicitações. Esta seção abordará o sistema voltado para Comunidade que é dividida em três páginas diferentes conforme as imagens a seguir: a página que contém as solicitações criadas pelo próprio usuário, a página que contém as solicitações criadas por toda Comunidade e a página de detalhes de uma solicitação que pode ser acessada através do cartão de solicitação.

Figura 33 – Sistema Comunidade - Minhas solicitações



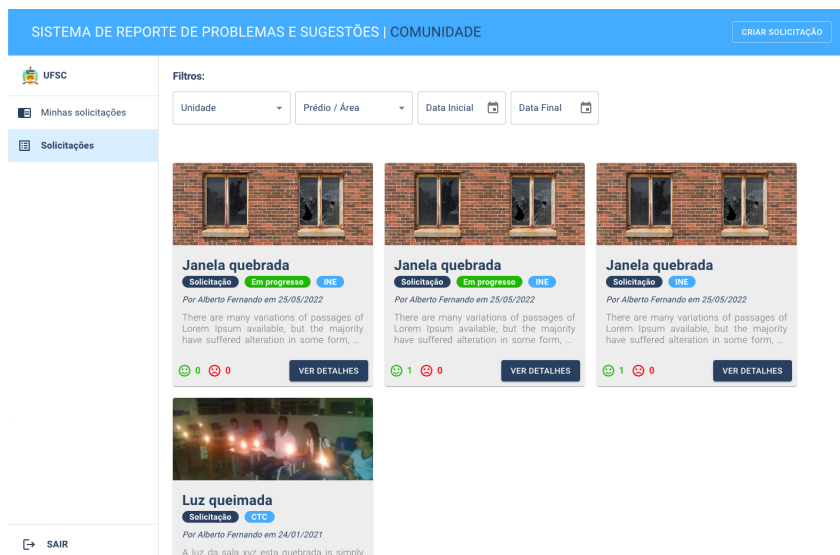
Fonte: Autor.

Figura 34 – Sistema Comunidade - Minhas solicitações versão mobile



Fonte: Autor.

Figura 35 – Sistema Comunidade - Solicitações da comunidade



Fonte: Autor.



Figura 36 – Sistema Comunidade - Detalhes de uma solicitação



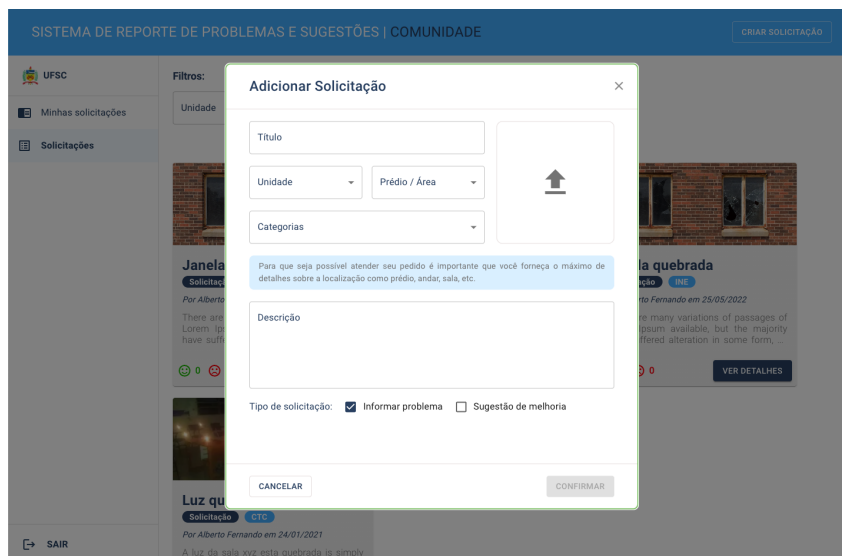
Fonte: Autor.

As páginas do sistema da Comunidade têm estruturas similares:

- Cabeçalho: o nome do sistema, o tipo de sistema em que o usuário está acessando (Comunidade ou Universidade) e um botão para criar uma nova solicitação;
- Menu lateral: contém a lista de páginas que o usuário pode acessar e no canto inferior esquerdo o botão para sair do sistema;
- Conteúdo da página: Nas páginas de listagem de solicitações, sejam elas solicitações do Usuário ou da comunidade em geral, os usuários tem acesso aos cartões de solicitações que mostram cada solicitação de forma resumida além dos filtros para facilitar a busca com características específicas. No caso da página de detalhes da solicitação é possível ver todos os dados da solicitação bem como a descrição completa e botões para interagir caso a solicitação não tenha sido criada pelo próprio usuário.

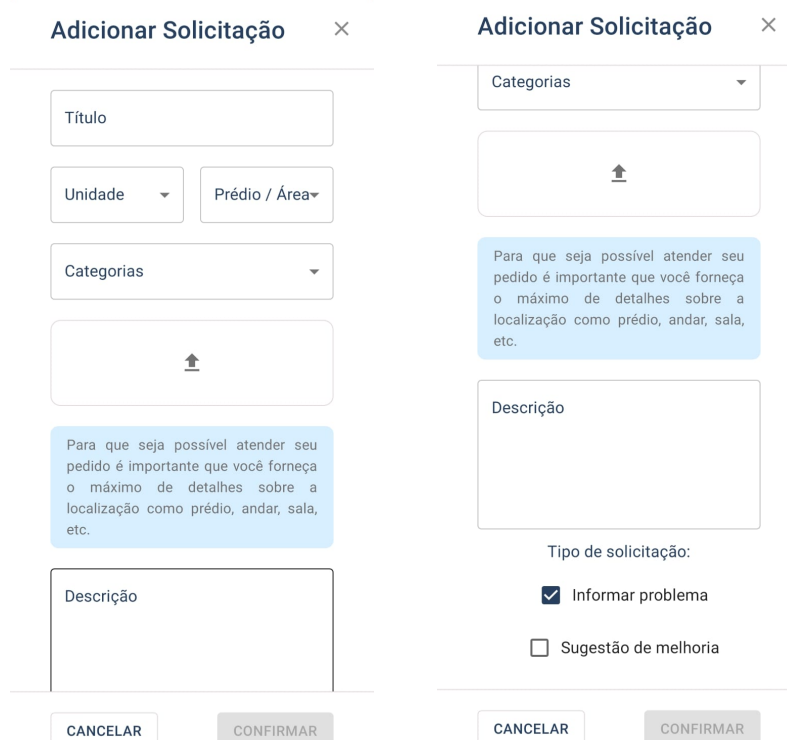
Outro ponto comum à todas as páginas, como citado anteriormente, é o botão de criação de novas solicitação presente no cabeçalho. A abordagem utilizada para o desenvolvimento desta funcionalidade foi uma caixa de sobreposição, o que possibilita ao usuário criar uma nova solicitação independente da página do sistema em que ele se encontra como apresentado na figura a seguir:

Figura 37 – Sistema Comunidade - Criação de solicitação versão *web*



Fonte: Autor.

Figura 38 – Sistema Comunidade - Criação de solicitação versão *mobile*

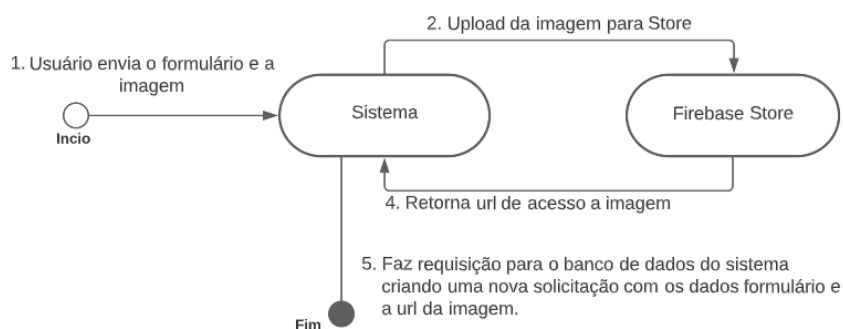


Fonte: Autor.

Para concretizar a etapa de criação de solicitação utilizou-se a integração com serviço Store do Firebase que possibilita a hospedagem de arquivos e disponibiliza uma *Uniform Resource Locator (URL)* de acesso direto ao arquivo. Esta abordagem

evita armazenar no banco de dados a imagem, tendo em vista que uma *URL* é um conjunto de dados mais simples, isso torna a aplicação mais rápida. A lógica para a criação de uma solicitação é ilustrado no fluxograma a seguir:

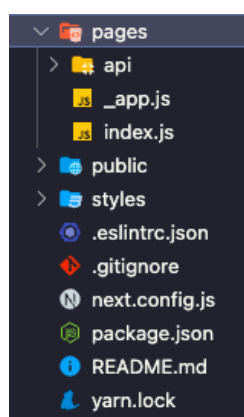
Figura 39 – Fluxo de informação - Criação de solicitação



Fonte: Autor.

Antes de discorrer sobre a lógica aplicada nas páginas é válido ressaltar que cada página tem uma *URL* de acesso diferente. O *framework* escolhido para o desenvolvimento do sistema facilita a separação de página por rotas através da própria arquitetura de arquivos do projeto. Ao instanciar um projeto utilizando *Next.js* o *framework* disponibilizará uma estrutura mínima como é possível observar na imagem a seguir:

Figura 40 – Estrutura inicial do projeto em NextJS

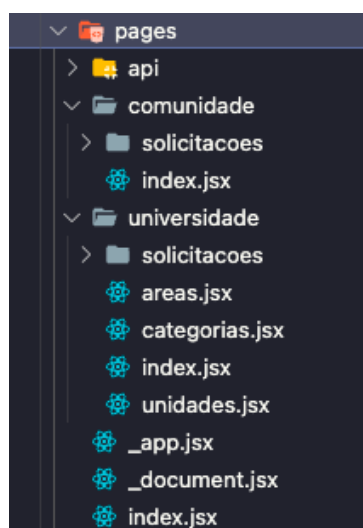


Fonte: Autor.

A pasta *pages* contém todas as páginas e os arquivos dentro desta pasta são tratados como rotas pela aplicação, na Figura 40 o arquivo *index.js* é responsável por renderizar todos os componentes e elementos presente que fazem parte do contexto de uma *url* hipotética *www.exemplo.com.br/*. Ao adicionar um arquivo denominado

user este corresponderá ao conteúdo presente na *url* `www.exemplo.com.br/user`. Outra funcionalidade utilizada para a elaboração do sistema são as *urls* dinâmicas, como por exemplo a página de detalhes `www.exemplo.com.br/solicitacao/2`. Nesta situação utilizamos um sistema de pasta no lugar do arquivo *javascript* diretamente, ou seja, criamos uma pasta denominada *solicitacao* e dentro adicionamos um arquivo que segue o padrão descrito na documentação próprio *framework*<sup>10</sup> que é `[nome-da-propriedade].js` uma vez que o conteúdo contido na página depende do parâmetro que será adicionado na *url*. Na Figura 41 vemos o exemplo da estrutura presente no desenvolvimento do sistema:

Figura 41 – Estrutura de arquivos/paginas do sistema



Fonte: Autor.

Na Figura 41 pode-se notar que é utilizada a funcionalidade de rotas dinâmicas para as páginas de solicitações, conforme abordado anteriormente, onde cada página receberá como propriedade o *id* de cada solicitação que o usuário deseja acessar.

Na pasta *API* podemos introduzir rotas que serão tratadas como *endpoints* possibilitando uma integração no lado do servidor, ou seja, quando acessado uma *URL* referente a essa pasta, ela será executada antes mesmo da página retornar o *browser* do navegador de maneira análoga a uma requisição de uma *API*. Essa funcionalidade é importante pois o *Next.js* disponibiliza por padrão a utilização de uma tecnologia denominada *Server Side Rendering (SSR)*, ou seja, renderização no lado do servidor.

A maneira mais comum utilizada por padrão nos *frameworks* como React, Angular ou Vue é denominada *Client Side Rendering (CSR)*, desta forma quando o usuário acessa uma página, o servidor envia todos os arquivos para serem computados no

<sup>10</sup><https://nextjs.org/docs/routing/dynamic-routes>

próprio navegador que está acessando a página. Este processo apesar de ser muito rápido para um usuário comum gera impacto quando a página é acessada por um *crawler*, como o sistema de indexação do Google.

#### 4.5.3.1 Minhas solicitações

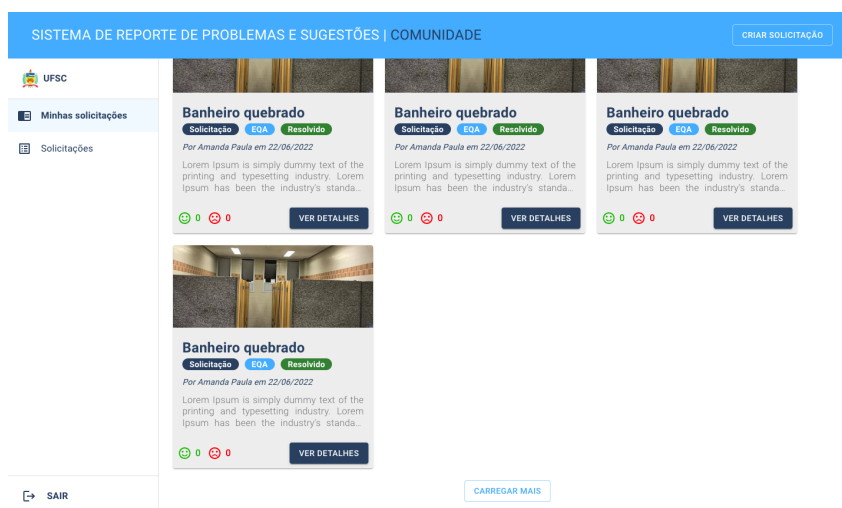
A página de solicitações criadas pelo próprio Usuário representada previamente através da Figura 33 pode ser acessada através da *URL /comunidade*. Esta página é protegida, ou seja, caso o sistema não identifique no estado atual da aplicação ou nos *cookies* do navegador as informações referente ao Usuário, o mesmo será redirecionado para a página de acesso do sistema.

Ao acessar a página devidamente logado, o Usuário tem a possibilidade visualizar cada solicitação enviada e suas respectivas informações.

Cada cartão contém as informações referente à solicitação como: título, descrição, tipo de solicitação, centro acadêmico, caso a solicitação já tenha sido resolvida também é adicionada essa informação, número de interações positivas e negativas da comunidade e por fim o botão que redireciona o usuário para tela de detalhes.

Neste primeiro acesso, são carregadas as primeiras dez solicitações criadas pelo usuário, ordenadas pela data de criação. Inicialmente não é aplicado nenhum filtro. Caso o usuário tenha mais que dez solicitações, a página adiciona automaticamente o botão para carregar mais como na Figura 42:

Figura 42 – Comunidade - Botão carregar mais



Fonte: Autor.

Para a implementação da lógica de paginação, o Hasura suporta parâmetros *offset* e *limit* que funcionam de maneira similar a um ponteiro, ou seja, *offset* refere-se ao cursor ou ponto de referência na lista e o *limit* a quantidade de limite que desejamos

obter com a requisição. No primeiro acesso, os parâmetros são *offset* igual a zero e o *limit* será sempre dez para que a cada iteração o sistema acrescente os próximos itens no final da lista similar a uma progressão aritmética.

Juntamente com esta primeira requisição, o sistema faz outra requisição para saber o número total de solicitações existente no banco de dados criadas por este usuário, desta forma caso o número contido na lista apresentada na página seja inferior ao número total, acrescenta-se o botão para que seja possível visualizar mais itens.

Caso o usuário deseje aplicar os filtros, todo o estado da paginação é restaurado. Realiza-se uma nova requisição para mostrar os primeiros dez itens e a quantidade total de solicitações que satisfazem as condições contidas nos filtros.

#### 4.5.3.2 Solicitações da comunidade

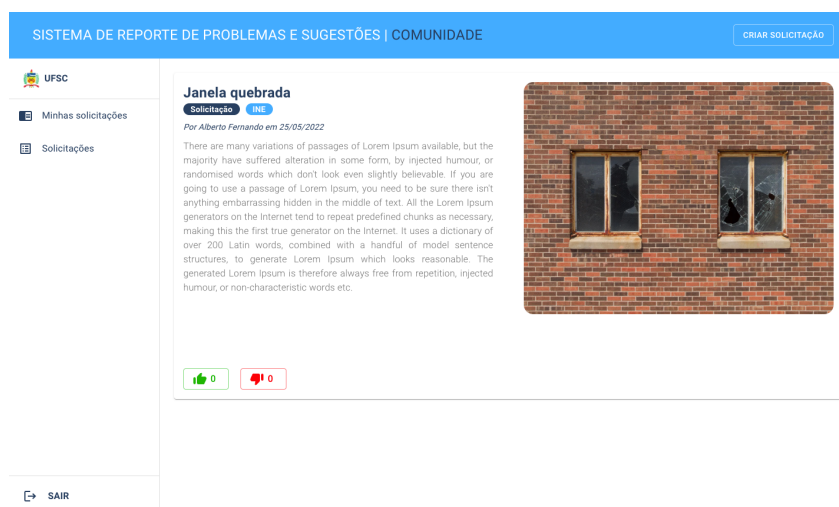
De forma análoga à página de solicitações do usuário, tem-se a mesma interface e a mesma lógica de paginação. Nesta parte do sistema a requisição das solicitações da comunidade contém todas as solicitações, exceto as do usuário que está efetuando a requisição.

Nesta aplicação, foi implementada uma ordenação da lista baseado no número de interações positivas dos outros usuários da comunidade e da data de criação respectivamente, ou seja, primeiramente serão apresentadas as solicitações com mais interação positiva e posteriormente serão ordenadas pela data de criação, da mais recente para mais antiga. Idealmente o sistema deveria implementar um algoritmo de recomendação como “Filtragem colaborativa” descrito por *Paul Resnick* e *Hal Varian* em *Recommender Systems* porém a complexidade da implementação deste algoritmo inviabilizaria o desenvolvimento deste projeto que tem como foco a elaboração de um sistema que proporcione a melhoria do ambiente acadêmico.

#### 4.5.3.3 Detalhes da solicitação

Esta página é onde o usuário tem uma melhor visualização da solicitação, além dos botões de interação que possibilitam votar positivamente atestando a veracidade e apoiando a solicitação ou votar negativamente. O usuário pode interagir uma única vez com cada postagem. Além dessas interações, nesta página a imagem para ilustrar o problema é apresentada de forma integral, juntamente com a descrição completa do problema, conforme a figura abaixo. Diferentemente das páginas anteriores que apresentava um resumo da solicitação que limitava a três linhas e uma miniatura da imagem.

Figura 43 – Comunidade - Detalhes da solicitação



Fonte: Autor.

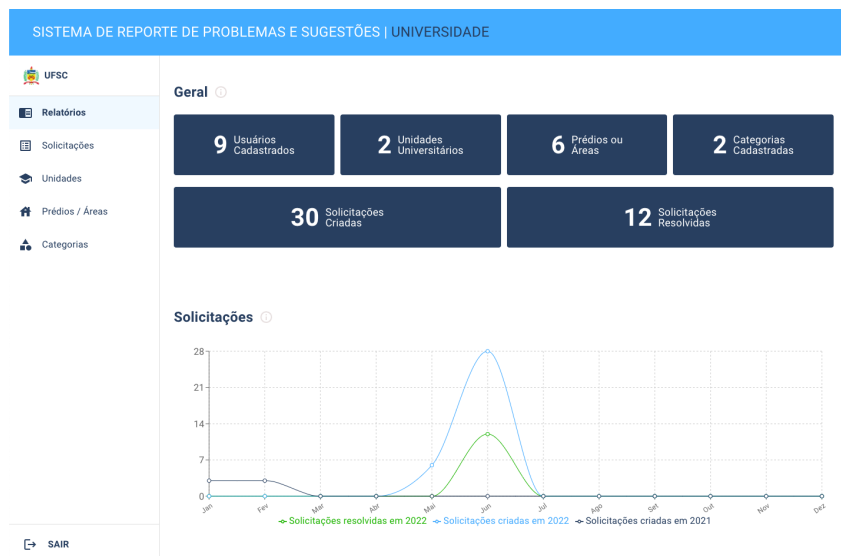
#### 4.5.4 Universidade

O sistema voltado para a gestão da Universidade é composto por seis páginas que possibilitam a manutenção das solicitações criadas pela comunidade, além de ter uma página dedicada a apresentação de dados gerais.

##### 4.5.4.1 Relatórios

A página de relatório visa fornecer à Universidade dados referente às solicitações criadas pela comunidade e dados gerais da plataforma. Ao acessar esta parte do sistema, são realizadas duas grandes requisições para a *API* que permitem acessar os dados referentes a todo o período como o número total de: usuários cadastrados na plataforma, categorias, unidades, prédios e áreas, solicitações criadas e solicitações resolvidas. Também é computado um comparativo entre o ano atual e o ano anterior, referente ao número de solicitações criadas por mês, sendo este apresentado na forma gráfico. Ambas seções de dados gerais e de comparação anual das solicitações são apresentadas nas figura a seguir.

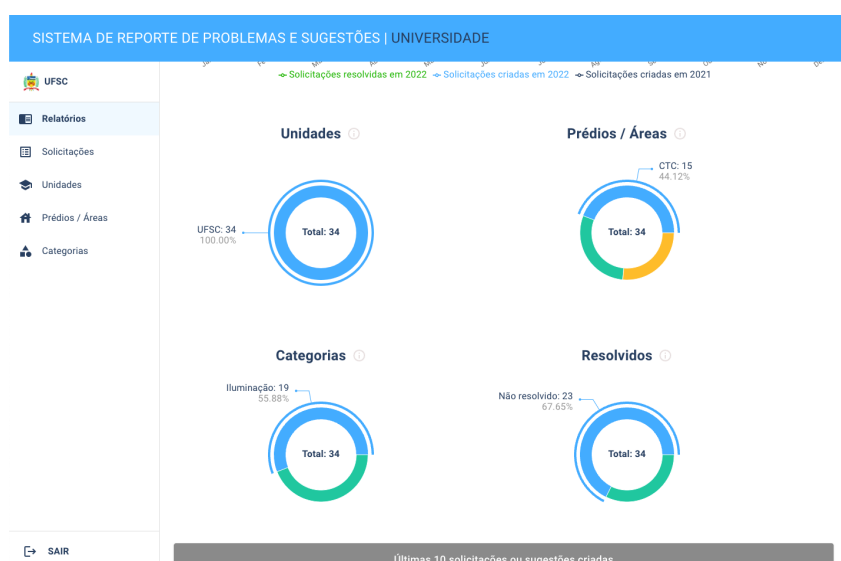
Figura 44 – Universidade - Seção de dados gerais e gráficos de solicitações



Fonte: Autor.

Na sequência, são apresentados dados do ano atual através de quatro gráficos de rosca referentes ao número de solicitações por: unidades, prédios e áreas, categorias e solicitações resolvidas. O gráfico de roscas possibilita interação ao passar o cursor no setor desejado, apresentando ao usuário o número de solicitações em cada setor e a porcentagem em relação ao todo.

Figura 45 – Universidade - Gráfico de rosca



Fonte: Autor.

Por fim, o gestor tem acesso à duas tabelas, que podem ser visualizadas abaixo.



A primeira contém as últimas dez solicitações criadas, as categorias que está inserida e um botão que redireciona para a página de detalhes da solicitação. A segunda tabela apresenta as dez solicitações criadas com mais confirmações feitas pela comunidade contendo, de maneira análoga, as categorias que cada solicitação está inserida e o botão de acesso a página de detalhes.

Figura 46 – Universidade - Tabela das últimas dez solicitações

SISTEMA DE REPORTE DE PROBLEMAS E SUGESTÕES | UNIVERSIDADE

Últimas 10 solicitações ou sugestões criadas

Data	Unidade	Prédio/Área	Categoria	Título	
23/06/2022	UFSC	INE	Iluminação	Lâmpada queimada	<a href="#">VER MAIS</a>
23/06/2022	UFSC	INE	Iluminação	Lâmpada queimada	<a href="#">VER MAIS</a>
23/06/2022	UFSC	INE	Iluminação	1231	<a href="#">VER MAIS</a>
23/06/2022	UFSC	INE	Iluminação	1231	<a href="#">VER MAIS</a>
23/06/2022	UFSC	CTC	Iluminação	Mobile	<a href="#">VER MAIS</a>
23/06/2022	UFSC	CTC	Iluminação	Mobile	<a href="#">VER MAIS</a>
22/06/2022	UFSC	CTC	Iluminação	Melhoria na iluminação	<a href="#">VER MAIS</a>
22/06/2022	UFSC	CTC	Iluminação	Melhoria na iluminação	<a href="#">VER MAIS</a>
22/06/2022	UFSC	CTC	Iluminação	Melhoria na iluminação	<a href="#">VER MAIS</a>
22/06/2022	UFSC	CTC	Iluminação	Melhoria na iluminação	<a href="#">VER MAIS</a>

Fonte: Autor.

Figura 47 – Universidade - Tabela das dez solicitações mais votadas

SISTEMA DE REPORTE DE PROBLEMAS E SUGESTÕES | UNIVERSIDADE

Últimas 10 solicitações ou sugestões criadas

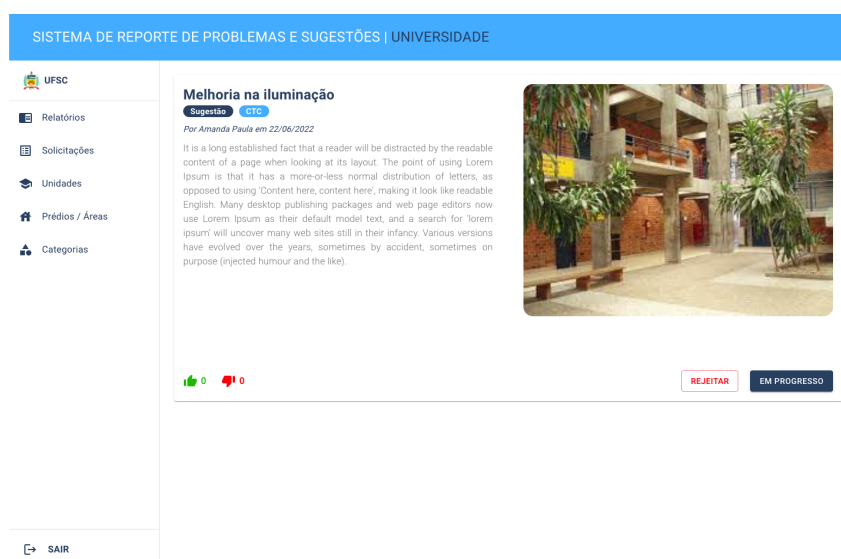
Data	Unidade	Prédio/Área	Categoria	Título	
23/06/2022	UFSC	INE	Iluminação	Lâmpada queimada	<a href="#">VER MAIS</a>
23/06/2022	UFSC	INE	Iluminação	Lâmpada queimada	<a href="#">VER MAIS</a>
23/06/2022	UFSC	INE	Iluminação	1231	<a href="#">VER MAIS</a>
23/06/2022	UFSC	INE	Iluminação	1231	<a href="#">VER MAIS</a>
23/06/2022	UFSC	CTC	Iluminação	Mobile	<a href="#">VER MAIS</a>
23/06/2022	UFSC	CTC	Iluminação	Mobile	<a href="#">VER MAIS</a>
22/06/2022	UFSC	CTC	Iluminação	Melhoria na iluminação	<a href="#">VER MAIS</a>
22/06/2022	UFSC	CTC	Iluminação	Melhoria na iluminação	<a href="#">VER MAIS</a>
22/06/2022	UFSC	CTC	Iluminação	Melhoria na iluminação	<a href="#">VER MAIS</a>
22/06/2022	UFSC	CTC	Iluminação	Melhoria na iluminação	<a href="#">VER MAIS</a>

Fonte: Autor.

#### 4.5.4.2 Solicitações e Detalhe de Solicitação

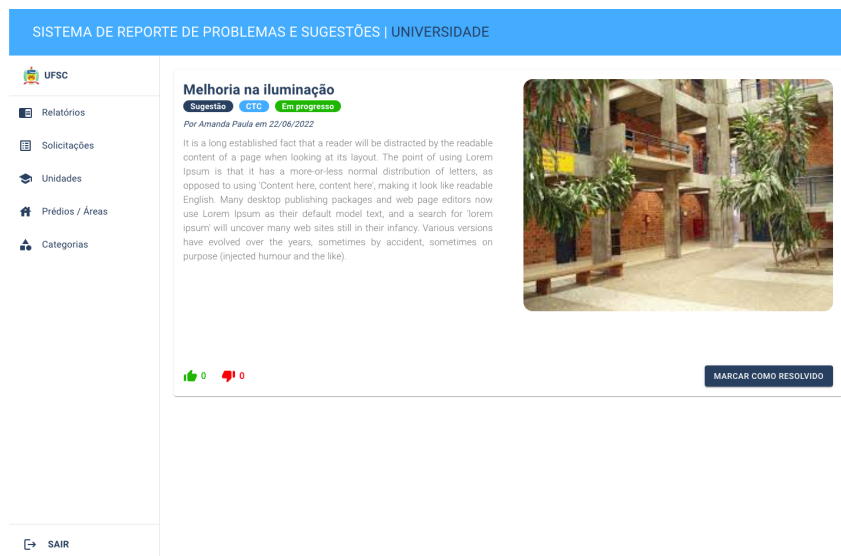
As páginas de solicitações e de detalhes de uma solicitação foram implementadas de maneira similar ao apresentado nas seções 4.5.3.2 e 4.5.3.3 respectivamente. Há uma diferença na página de detalhe da solicitação referente à interação com a solicitação: o gestor não pode votar positivamente ou negativamente. As únicas interações possíveis são marcar a solicitação como resolvida, rejeitada ou em progresso, conforme as imagens a seguir.

Figura 48 – Solicitação - Opções rejeitar e em progresso



Fonte: Autor.

Figura 49 – Solicitação - Opção marcar como resolvido



Fonte: Autor.

#### 4.5.4.3 Unidades, Prédios/Áreas e Categorias

Por fim, as páginas de unidades, prédios/áreas e categorias são acessíveis através do menu lateral e possibilitam o cadastro de novas opções para a comunidade e também a exclusão de opções previamente criadas, desde que não tenham sido utilizadas como referência para alguma solicitação ou usuário no caso das unidades e prédios/áreas. Abaixo estão representadas respectivamente as páginas de unidades, prédios/áreas e categorias:

Figura 50 – Universidade - Unidades

SISTEMA DE REPORTE DE PROBLEMAS E SUGESTÕES | UNIVERSIDADE

UFSC

Unidades + ADICIONAR UNIDADE

ID	Sigla	Nome	
1	UFSC	Universidade Federal de Santa Catarina	
5	JOINVILLE	Universidade Feral de Santa Catarina	

SAIR

Linhas por página: 100 1-2 de 2

Fonte: Autor.

Figura 51 – Universidade - Prédios/Áreas

SISTEMA DE REPORTE DE PROBLEMAS E SUGESTÕES | UNIVERSIDADE

UFSC

Prédios / Áreas + ADICIONAR PRÉDIO OU ÁREA

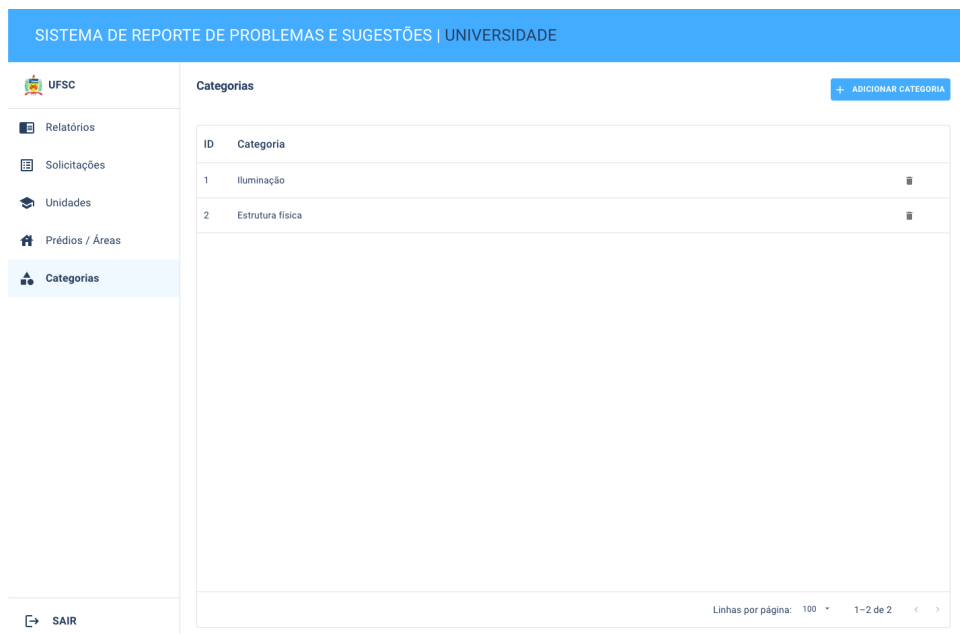
ID	Sigla	Nome	Unidade	
1	CTC	Centro Tecnológico	UFSC	
2	INE	Departamento de Informática e Estatística	UFSC	
3	EOA	Engenharia Química e Alimentos	UFSC	
4	CCE	Centro de Comunicação e Expressão	UFSC	
5	CSE	Centro Socioeconômico	UFSC	
6	CCS	Centro de Ciências da Saúde	UFSC	

SAIR

Linhas por página: 100 1-6 de 6

Fonte: Autor.

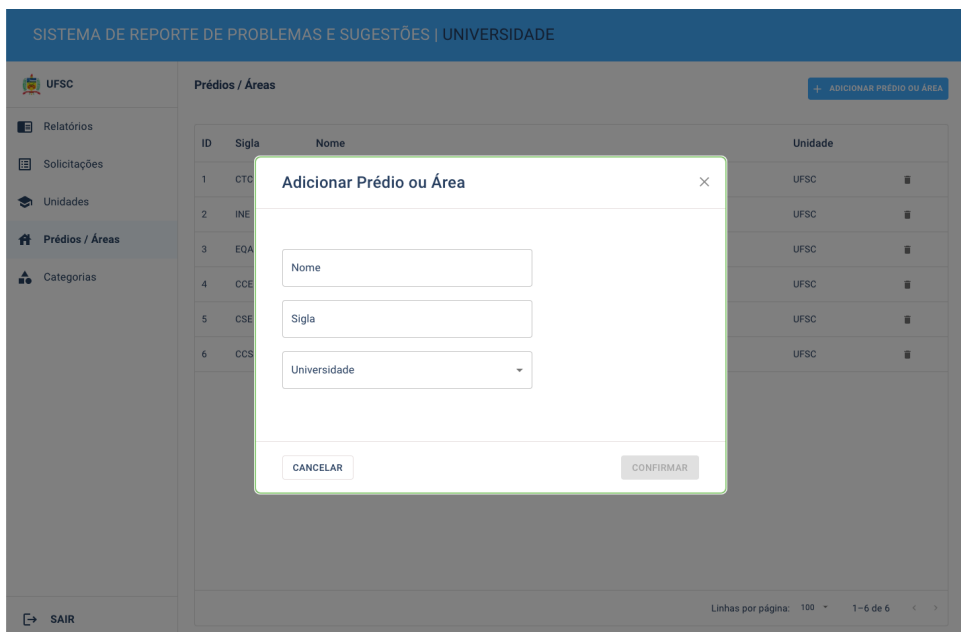
Figura 52 – Universidade - Categorias



Fonte: Autor.

De forma análoga à criação de solicitação, o botão para criar uma nova opção de unidade, prédio/área ou categorias abrirá uma janela sobreposta para que não seja necessário redirecionar o usuário para uma nova página, como ilustrado a seguir a tela de criação de um novo prédio ou área:

Figura 53 – Universidade - Criação de uma nova opção de prédio ou área



Fonte: Autor.

## 5 CONCLUSÃO

O presente trabalho elaborou um protótipo de sistema para auxiliar na gestão da manutenção de uma universidade a partir do estudo de sistemas de participação colaborativa já existentes, como a aplicação do modelo Open311 nos sistemas oferecidos pela Connected Bits e FixMyStreet. Também foram analisadas algumas iniciativas desenvolvidas nacionalmente, como as plataformas Colab e Participact. Um fator importante para a realização deste trabalho, foi a compreensão das diferentes estruturas organizacionais dentro das instituições de ensino a fim de elaborar um modelo que permita às universidades autonomia para tratar as solicitações.

A partir do desenvolvimento deste protótipo possibilitou revisar os conhecimentos e aprofundar os estudos sobre as arquiteturas e tecnologias disponíveis no mercado atual. Foi escolhido o *framework* *NextJS* que utiliza a linguagem *Javascript* para desenvolver o *Frontend* da aplicação. Para o *Backend* foi escolhido o *framework* *Hasura*, que por sua vez utiliza o banco de dados *PostgressSQL* e a linguagem de consulta *GraphQL*. O serviço *Authentication* da plataforma *Firebase* foi utilizado para realizar a autenticação, a criação e recuperação de contas no sistema. O serviço *Store* desta mesma plataforma foi utilizado para o armazenamento das imagens das solicitações. Todas as decisões necessitaram do estudo de diversas opções para que fosse possível integrar as tecnologias escolhidas a fim de produzir um produto funcional.

Por fim, espera-se que o presente trabalho possa servir de modelo para a criação de um projeto real dentro das universidades e que justifique a importância de uma atuação efetiva na manutenção de todo espaço comum, visto que esta efetividade disponibilizará mais recursos para serem aplicados no que há de mais importante dentro das instituições de ensino: a educação.

### 5.1 SUGESTÕES PARA TRABALHOS FUTUROS

O protótipo desenvolvido neste trabalho resulta em uma aplicação funcional, porém acredita-se que novas funcionalidades resultaram em um sistema mais completo. São elas:

- Integração com um sistema de autenticação da própria universidade, fazendo com que não seja necessária a criação de uma nova conta e seja vinculada à matrícula do estudante ou servidor universitário;
- Um algoritmo de recomendação para apresentar aos membros da comunidade as solicitações, baseado nas interações de outros usuários ou na localização do indivíduo;

- Desenvolvimento de uma aplicação mobile para a comunidade que possibilite interação com outras funcionalidades dos dispositivos móveis, como o acesso à localização do usuário;
- Funcionalidade de comentários para as solicitações, que permitam a interação entre os membros da comunidade.



## REFERÊNCIAS

BOSTON'S Mayor's Office of New Urban Mechanics. [S.l.: s.n.], 2019. [Online; accessed 12-março-2022]. Disponível em: <https://www.centreforpublicimpact.org/case-study/bostons-mayors-office-new-urban-mechanics>.

CARMEN, Malena; FORSTER, Reiner; SINGH, Janmejay. **Social Accountability - An Introduction to the Concept and Emerging Practice**. [S.l.: s.n.], 2004. [Online; accessed 10-janeiro-2022]. Disponível em: <https://www.ircwash.org/resources/social-accountability-introduction-concept-and-emerging-practice>.

DRAGONI, Nicola; GIALLORENZO, Saverio; LAFUENTE, Alberto Lluch; MAZZARA, Manuel; MONTESI, Fabrizio; MUSTAFIN, Ruslan; SAFINA, Larisa. **Microservices: Yesterday, Today, and Tomorrow**. [S.l.: s.n.], 2017. [Online; accessed 22-junho-2022]. Disponível em: [https://link.springer.com/chapter/10.1007/978-3-319-67425-4\\_12](https://link.springer.com/chapter/10.1007/978-3-319-67425-4_12).

JULNES, Patria De Lancer; GIBSON, Ed. **Innovation in the Public and Nonprofit Sectors: A Public Solutions Handbook**. 1. ed. USA: [s.n.], 2015. Disponível em: <https://books.google.com.br/books?id=zK40CwAAQBAJ&pg=PT105&lpg=PT105&dq#v=onepage&q&f=false>. Acesso em: 16 dez. 2021.

RELATÓRIO de Gestão 2019. [S.l.: s.n.], 2019. [Online; accessed 20-junho-2022]. Disponível em: <https://dplseplan.paginas.ufsc.br/files/2013/02/Relatorio-de-Gestao-2019.pdf>.

RODRIGUEZ, Alex. **RESTful Web services: The basics**. [S.l.: s.n.], 2008. [Online; accessed 12-março-2022]. Disponível em: <https://cs.calvin.edu/courses/cs/262/kvlinden/references/rodriguez-restfulWS.pdf>.

# **Apêndices**

## APÊNDICE A – ARTIGO SISTEMA PARA REPORTAR PROBLEMAS E SUGESTÕES NA ESTRUTURA FÍSICA DE UM CAMPOS UNIVERSITÁRIO

### Sistema para reportar problemas e sugestões na estrutura física de um campus universitário.

**José E. De Lucca, José A. da França**

Departamento de Informática e Estatística  
Universidade Federal de Santa Catarina (UFSC) – Florianópolis, SC – Brazil

jose.lucca@ufsc.br, j.franca@grad.ufsc.br

**Abstract.** *Universities are in constant process of maintenance and improvement of their physical structure in order to provide the entire academic community with an environment that supports teaching, research and extension. Thus, aiming to foster a collaborative process between community members and their institutions, this article implements a system that helps members of the academic community to report infrastructure problems and make suggestions for improvements to the physical structure of a university campus using as an example the context of the Campus Reitor João David Ferreira Lima of the Federal University of Santa Catarina.*

**Resumo.** *As universidades estão em constante processo de manutenção e melhoria de sua estrutura física a fim de proporcionar a toda comunidade acadêmica um ambiente que apoie o ensino, a pesquisa e a extensão. Desta forma, visando fomentar um processo colaborativo entre os membros das comunidades com suas instituições, este artigo implementa um sistema que ajude os membros da comunidade acadêmica a reportarem problemas de infra-estrutura e a fazerem sugestões de melhorias sobre estrutura física de um campus universitário utilizando como exemplo o contexto do Campus Reitor João David Ferreira Lima da Universidade Federal de Santa Catarina.*

#### 1. Introdução

Atualmente as instituições públicas e privadas buscam um melhor aproveitamento dos recursos, cortando gastos e investindo no que é realmente necessário. Com a popularização da tecnologia e o seu uso para situações corriqueiras, notou-se uma forte tendência mundial na adoção de serviços com vertente cívica, visando incluir os cidadãos a participar ativamente da melhoria da sociedade. Esses serviços cívicos não são novidade, no entanto com a popularização dos dispositivos móveis eles estão sendo cada vez mais usados ao redor do mundo, um exemplo é o protocolo 311 que está presente em diversas cidades ao redor do mundo,

About 34 municipalities across the world have Open 311 server endpoints (27 of which are in the U.S.) (as per Open 311 Wiki, visited on August 18, 2014). However, there are many private sectors firms specializing in Gov 2.0 solutions that provide 311 services to different local government entities (cities, counties, and other special districts). See ClickFix, for example, server nearly 290 such units; CitySourced features over 100 government entities;

PublicStuff claims presence in over 200 cities (Gailbraith, 2013). [...] (JULNES; GIBSON, 2015)

Devido a pandemia de COVID-19, as instituições de ensino superior federais no Brasil ficaram com os espaços ociosos nesses dois anos de restrições sanitárias. Desta forma, com o retorno das atividades presenciais nas universidades é esperado que o número de chamados de manutenção aumente. O presente artigo implementou um sistema que permite à comunidade universitária comunicar à Universidade, de forma centralizada, problemas na estrutura física e sugestões de melhoria. Foi realizado o mapeamento do fluxo de informação para solução das demandas de manutenção das estruturas, uma vez que a Universidade tem diferentes órgãos e/os setores que são responsáveis por manter e aprimorar todo ecossistema, tal mapeamento foi importante para esclarecer como está estruturada toda a cadeia de responsabilidades dentro da universidade.

O desenvolvimento deste protótipo se baseia em soluções já existentes no mercado voltadas para municípios, como a ParticipAct Brasil<sup>1</sup>, projeto desenvolvido pela Universidade do Estado de Santa Catarina (UDESC) em parceria com a Universidade Federal de Santa Catarina (UFSC) e a Universidade de Bologna (UNIBO) que tem como principal objetivo a criação de uma grande estrutura de banco de dados a partir das informações das tecnologias da informação e da comunicação, com o intuito de melhorar a gestão de cidades inteligentes. Além deste, há também o sistema desenvolvido pela startup Colab<sup>2</sup> que tem como objetivo “dar à população o poder de colaborar com questões do setor público através de publicações de zeladoria urbana, participação em tomadas de decisões e em consultas de avaliação de serviços públicos”. Outros exemplos de sistemas também estudados são os sistemas que aplicam o protocolo Open311<sup>3</sup>, tais quais as plataformas da empresa *Connected Bits*<sup>4</sup> e a plataforma *FixMyStreet*<sup>5</sup>.

## 2. Desenvolvimento

### 2.1. Análise da organização das Universidades

Pensar a elaboração de um sistema que possibilite as universidades lidarem de forma simples, transparente e mais colaborativa com os problemas e sugestões de melhorias de suas estruturas físicas, equipamentos, entre outros, é um trabalho complexo uma vez que cada universidade tem seu próprio fluxo de informação e gestão de problemas. Desta forma, utilizou-se neste trabalho o Campus Reitor João David Ferreira Lima como referência. A Universidade Federal de Santa Catarina conta com cinco campi (Araranguá, Blumenau, Curitibanos, Florianópolis e Joinville), o que requer uma grande estrutura de gerenciamento para a garantia de um bom funcionamento e uma gestão que possibilite a execução e oferta de ensino, pesquisa e extensão. A Figura 1 exemplifica o

---

<sup>1</sup> <http://participact.tecnologia.ws/>

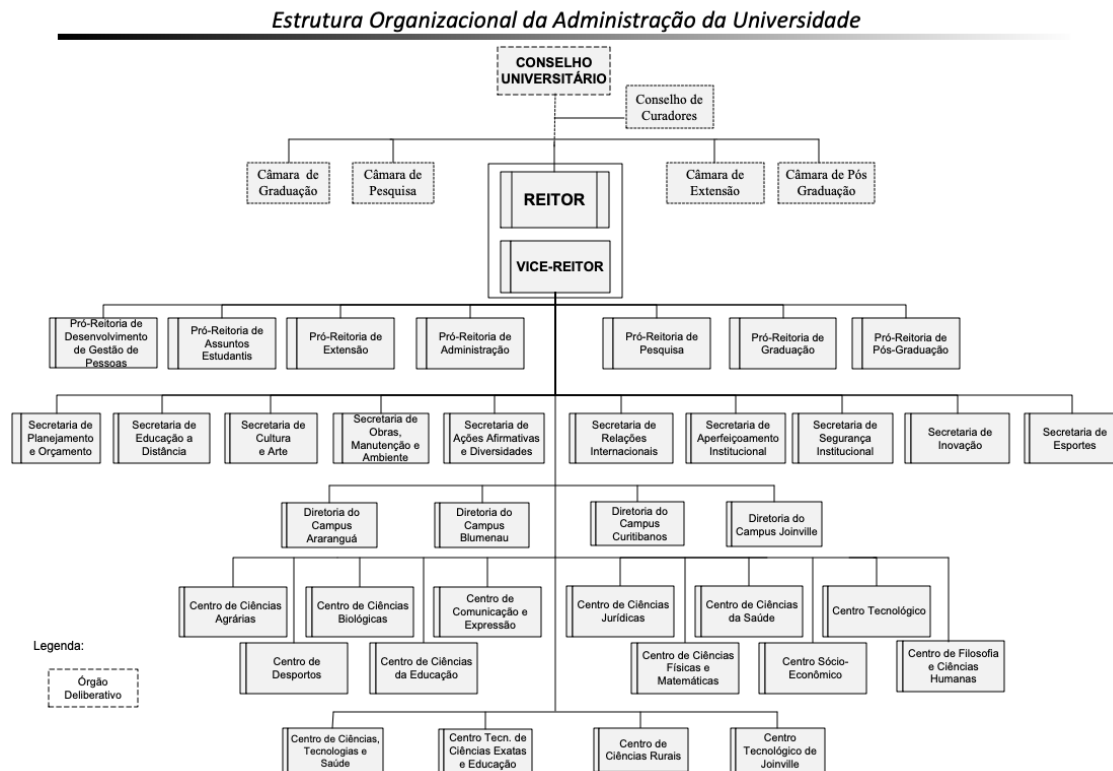
<sup>2</sup> <https://www.colab.re/>

<sup>3</sup> <https://www.open311.org/learn/>

<sup>4</sup> <http://grcity.spotreporters.com/#header>

<sup>5</sup> <https://www.fixmystreet.com>

tamanho e a complexidade da estrutura organizacional da universidade, nessa estrutura destaca-se a divisão de centros que são pontos onde se concentram as solicitações de manutenção.



**Figura 1. Estrutura Organizacional da Administração da UFSC.**

No caso da UFSC o gerenciamento da manutenção é de responsabilidade da Secretaria de Obras, Manutenção e Ambiente (SEOMA) que atua quando os chamados de manutenção são solicitados pelos centros e/ou departamentos. As realizações das manutenções, dependendo de sua natureza ou origem, podem ser atendidas pela Prefeitura Universitária (PU), através do Núcleo de Manutenção (NUMA) ou pelo Departamento de Manutenção Predial e de Infraestrutura (DMPI). Para cada solicitação, o solicitante deve enviar um requerimento junto ao respectivo centro (por exemplo CTC) e se aprovado pela direção do centro é encaminhado ao órgão responsável pela manutenção onde será dado prosseguimento ao processo de verificação do problema e, caso possível a resolução do problema, realiza-se a abertura de um chamado de manutenção.

Apesar de diferentes estruturas, outras universidades brasileiras também apresentam grandes estruturas organizacionais que formam grandes fluxos de informação entre quem solicita a resolução de um problema e quem de fato irá solucionar. A Universidade Federal do Sergipe, tem a Superintendência de Serviços de Infraestrutura (INFRAUFS) que por sua vez tem subunidades como: o Departamento de Manutenção (DEMAN) que é "responsável pela execução dos serviços de manutenção

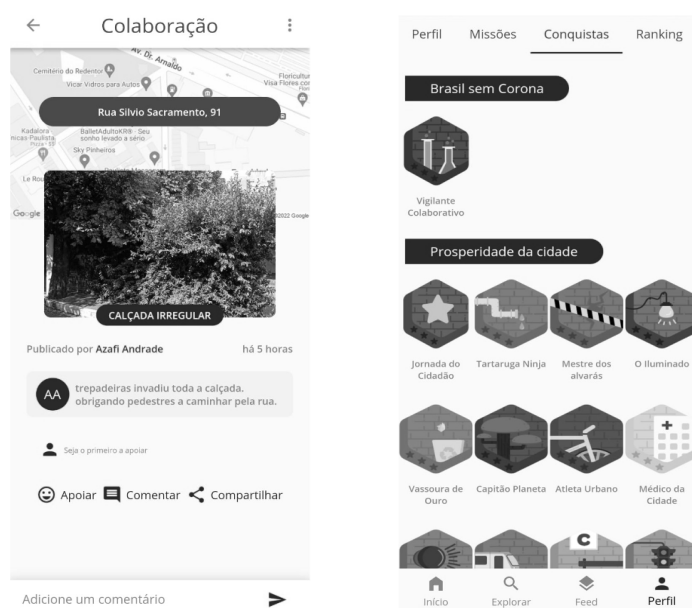
de máquinas, aparelhos, equipamentos, utensílios, móveis e imóveis"<sup>6</sup> e a Divisão de Manutenção de Máquinas e Equipamentos (DIMEQ).

A complexidade das organizações dificulta a interação entre a comunidade universitária e os setores responsáveis pelas manutenções, por outro lado se torna inviável a reestruturação das organizações a fim de suportarem um novo sistema, desenvolvido neste trabalho. Desta forma o sistema elaborado visa a implementação de uma via secundária de comunicação, onde após o recebimento de um pedido de manutenção ou sugestão de melhoria a Universidade tenha autonomia para tratar da melhor forma.

## 2.2. Análise de Sistemas Similares

### 2.2.1. COLAB

O Colab (<https://www.colab.re/>) é uma plataforma que busca que as pessoas exerçam cidadania por meio da sua aplicação. Eles oferecem um aplicativo gratuito, onde o usuário faz o login, compartilha sua localização e pode reportar problemas relacionados à sua cidade, podendo adicionar foto, descrição, endereço e título. Após a postagem, o usuário receberá um email informando se a Prefeitura está utilizando o sistema para receber a solicitação. Em caso positivo, a Prefeitura irá responder a solicitação e encaminhar para o setor responsável resolver. O aplicativo permite que outros cidadãos apoiem, comentem e compartilhem as publicações de outros usuários, funcionando com uma rede social. Além disso, o aplicativo traz uma experiência gamificada, onde os usuários ganham selos ao engajarem com a plataforma, tornando a experiência interessante e envolvente para o usuário.



Figuras 2 e 3. Solicitação feita no aplicativo Colab.

<sup>6</sup> <https://infra.ufs.br/pagina/21508-departamento-de-manutencao>

Há também a aplicação destinada aos órgãos governamentais (<https://www.colab.re/gov>) que permite a gestão das demandas recebidas através do módulo "Monitor" e o acesso aos relatórios de dados e indicadores através de um módulo denominado "Panorama", o que possibilita uma melhor tomada decisão por parte das instituições uma vez que essas decisões podem ser baseadas em dados.

### 2.2.2. Participact

O ParticipAct Brasil (<http://www.participact.com.br/>) é um projeto desenvolvido pela Universidade do Estado de Santa Catarina (UDESC) em parceria com a Universidade Federal de Santa Catarina (UFSC) e a Universidade de Bologna (UNIBO) que tem como principal objetivo a criação de uma grande estrutura de banco de dados a partir das informações das tecnologias da informação e da comunicação, com o intuito de melhorar a gestão de cidades inteligentes. O aplicativo permite que o usuário envie uma solicitação e adicione título, descrição, foto, áudio, vídeo e localização. A inclusão do envio de vídeo como ferramenta é interessante para ilustrar melhor o problema quando for necessário, e o envio de áudio como algo positivo em relação aos outros sistemas, visto que esse recurso auxilia na acessibilidade. O usuário consegue visualizar um mapa da cidade e ícones onde há solicitações, bem como as iniciais do problemas referido. O aplicativo também disponibiliza gráficos classificando os tipos de solicitações feitas no sistema em relação aos meses que ocorreram esses reportes.



Figuras 4 e 5. Mapa com as solicitações no aplicativo ParticipAct.

### **2.2.3. Protocolo OPEN311**

Nos EUA, várias cidades têm linhas telefônicas governamentais não emergenciais, acessíveis pelo número de telefone 311. Como consequência, "311" passou a significar todo o processo de manuseio de solicitações de serviço dos cidadãos em relação a questões não emergenciais, como lixo em local inadequado, buracos nas ruas e vizinhos barulhentos. Diversas municipalidades e empresas passaram a implementar soluções para gerenciar as demandas que chegam por esse serviço telefônico. Entretanto, com o desenvolvimento deste segmento de software, identificou-se a necessidade de um padrão para facilitar a adoção das diferentes aplicações criadas pelas empresas. Assim, surgiu a iniciativa Open311, que propõe um protocolo padronizado, colaborativo e baseado em localização para registro e acompanhamento de problemas. As tecnologias desenvolvidas em torno do protocolo Open311 usam a internet para transformar as antigas comunicações individuais síncronas feitas por um cidadão a um call center em um ambiente interativo, assíncrono e de muitos para muitos. Ou seja, diversas pessoas podem trocar informações abertamente sobre um problema local identificado, fornecendo muito mais informações para aqueles que precisam responder à solicitação. Ao tornar a informação pública, o sistema oferece transparência e automaticamente atribui responsabilidades. A transparência também garante que a voz de todos seja ouvida e isso encoraja a participação.

Durante a etapa de pesquisas sobre soluções similares à pensada neste artigo, foram encontradas diversas propostas ao redor do mundo que implementam os protocolos Open311, tais quais a Connected Bits e FixMyStreet.

#### **2.2.3.1 Connected Bits**

A Connected Bits é uma aplicação que possui diversas cidades do mundo como clientes, como: Dallas, Boston, Calgary, Austin, Edmonton, entre outras. A iniciativa de reporte de situações não emergenciais em Boston já existia a muitas décadas por meio de uma linha telefônica, mas percebeu-se que muitas pessoas não usavam esse serviço. Com o avanço da tecnologia e a ampliação dos aplicativos para celular, em 2010 o reporte de questões não emergenciais passou a ser feito por um aplicativo desenvolvido pela Connected Bits, a iniciativa teve muito sucesso e logo foi adotada pelos cidadãos de Boston. Hoje o aplicativo continua em funcionamento após muitas melhorias e tem muitas solicitações por dia, sendo possível ler sobre a equipe que resolveu o problema quando o reporte está concluído e em alguns casos é possível ver fotos do processo.

The Mayor's Office of New Urban Mechanics (MONUM) in Boston, MA was set up in 2010 to encourage greater civic engagement in the city through the use of innovation and experimentation, because Mayor Thomas Menino felt that government was too bureaucratic and lacked a human face. MONUM's main aim was to improve the quality of Bostonians' lives by involving them in participatory civic engagement. It has created many different projects to improve city services for citizens and workers, such as the app BOS:311 to report local maintenance issues to Boston City Hall.



MONUM has been widely successful and remained a feature of the Mayor's Office under the leadership of Mayor Marty Walsh. It currently works on a variety of projects across the city's departments, such as housing and education, and has even prompted other cities across America to adopt its approach. (BOSTON'S . . . , 2019)

Os sistemas pensados pela empresa Connected Bits possuem versão web e mobile, em ambas o usuário pode visualizar as solicitações feitas por outros usuários, filtrando-os em solicitações abertas (a resolver) e fechados (já resolvidos), além de tipo de serviço, como: estacionamento em local ilegal, buracos, calçadas, carro abandonado, entre outros. Os usuários também podem reportar problemas, para isso devem selecionar a categoria do problema, adicionar uma breve descrição, a geolocalização e uma foto. Ao acessar cada reporte é possível ler o título, a descrição, a foto (se houver), a API do Google Maps com a geolocalização do fato, o endereço e as coordenadas.

### 2.2.3.2. FixMyStreet

A aplicação FixmyStreet é utilizada na Inglaterra, França, Austrália, e outros. As funcionalidades da aplicação são similares às do Open311, mas a interface é diferente. O site filtra as solicitações por "council", portanto ao digitar o council escolhido aparecerá um mapa do local e pequenas bandeiras indicando onde há reportes. no menu lateral esquerdo pode-se ler do que se trata cada solicitação. É possível filtrar as solicitações por abertos, encerrados e resolvidos. Além disso, também é possível filtrar por categoria de problema e por ordenação. Ao selecionar uma solicitação, a bandeira referente a ele no mapa fica maior e é possível ler o título, a descrição, data que foi postado, status e foto se houver. Também é possível postar uma atualização sobre aquela solicitação, mostrando como está a situação no momento em que você está passando por aquele local. O FixmyStreet disponibiliza gráficos da quantidade de solicitações feitas e quantas foram resolvidas desde que foi lançada e nos últimos sete dias, é possível filtrar essas ferramentas por council.



Figuras 6. Solicitações no FixMyStreet do Council Merton.

## 2.2.4. Avaliação das ferramentas estudadas

Nesta análise foi possível identificar dois tipos de interface de design, a primeira é minimalista, concentra-se na comunicação entre o usuário e o sistema, proporcionando uma comunicação mais rápida na solicitação de problemas e melhorias. A gama de interações possíveis é, portanto, deliberadamente restringida; a escolha de categorias de serviço fixa. Como é o caso da SpotReporters da Connected Bits, da FixMyStreet e da ParticipAct.

O segundo tipo de interface se concentra em uma interação mais social, que proporciona mais interação entre os usuários e com a plataforma. Elas apresentam muitos elementos familiares de outras plataformas de mídia social. Cidadãos e funcionários públicos são representados por meio de usuários perfis; os usuários têm a possibilidade de avaliar e comentar sobre problemas existentes, seus feeds de atividades em um sistema de reputação simples.

Com essas informações pode-se perceber que as aplicações possuem muitas semelhanças, mas que cada uma contém suas particularidades. Utilizando esses modelos como inspiração para o presente trabalho foi acrescentada uma nova funcionalidade que não foi contemplada nas outras aplicações: o estado "em progresso". Esse estado serve para a Universidade notificar à comunidade acadêmica que aquela solicitação foi verificada e que está sendo tomada alguma providência para que seja resolvida o mais breve possível.

Nome da aplicação	Necessária autenticação?	Possível categorizar solicitações?	Possível filtrar solicitações?	Possível adicionar diversos tipos de mídia?	Possível interagir com solicitações de terceiros?	Fornecer dados estatísticos para o contratante do serviço?	Permite ao contratante marcar uma solicitação como "Em progresso"?
Colab	Sim, para enviar uma solicitação.	Sim, existem 68 categorias pré-definidas.	Não.	Apenas foto.	Sim, possível comentar, apolar e compartilhar.	Sim.	Não.
ParticipAct	Sim, para enviar uma solicitação.	Sim, existem 5 categorias pré-definidas.	Sim, por data e localização.	Sim, fotos, vídeos e áudios.	Sim, possível compartilhar e denunciar.	Sim.	Não.
Open311	Sim, para enviar uma solicitação.	Sim, existem 8 categorias pré-definidas e 22 subcategorias.	Sim, por status e tipo de solicitação.	Apenas foto.	Não.	Sim.	Não.
FixMyStreet	Não.	Sim, existem 20 categorias pré-definidas.	Sim, por data, status e tipo de solicitação.	Apenas foto.	Sim, possível enviar uma atualização e marcar como corrigida.	Sim.	Não.

**Tabela 1. Tabela comparativa das funcionalidades das aplicações pesquisadas.**

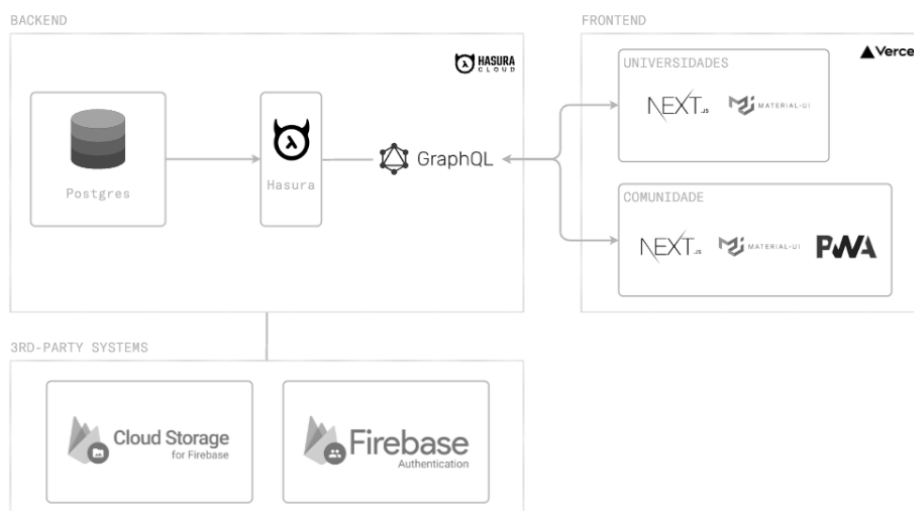
## 2.3. Desenvolvimento da aplicação

A pesquisa e o levantamento de requisitos foram realizadas com base na análise das plataformas de participação colaborativa (Open311, FixMyStreet, Colab e ParticipApt) e na análise das estruturas organizacionais das Universidades. Desta forma foram

levantados os requisitos e realizada a prototipação de um esboço da interface da aplicação com base nos sistemas citados anteriormente.

O sistema será separado em duas partes: o sistema voltado para o uso da comunidade acadêmica e o sistema voltado para a gestão da Universidade que será utilizado pela instituição para gerenciar as solicitações. Na parte voltada para a comunidade, as principais funcionalidades são: a criação das solicitações, ver a listagem de solicitações e poder interagir votando positivamente e negativamente. A parte do sistema voltada para a universidade tem como principais funcionalidades: acesso a lista de solicitações e acesso aos dados referentes às solicitações. Apesar de visualmente parecerem dois sistemas totalmente desacoplados, ambos foram implementados e fazem parte do mesmo código de implementação, então na verdade temos somente um sistema com duas partes: Universidade e Comunidade, como um monolito. Este conceito é uma aplicação desenvolvida neste modelo de arquitetura que requer que todos os módulos sejam executados em conjunto (DRAGONI et al., 2017).

A arquitetura do sistema foi pensada para suportar todos requerimentos e funcionalidades citadas previamente e entregar ao usuário final uma estrutura confiável e que esteja de acordo com as melhores práticas do desenvolvimento web. Apesar de ambos os sistemas, Universidade e Comunidade, serem desenvolvidos para web é necessário que o sistema da Comunidade tenha uma boa usabilidade nos *smartphones* colocando a praticidade como chave para um maior engajamento. Desta forma, a Figura a seguir apresenta arquitetura pensada para o desenvolvimento do sistema:



**Figura 7. Arquitetura do sistema**

### 2.3.1. Backend

Neste projeto vamos utilizar o Hasura como tecnologia para construção da *API*, que conecta um banco de dados *Postgress* com nossa aplicação *frontend* através da linguagem de consulta *GraphQL*.

Esta é uma tecnologia que permite a criação de interfaces de comunicação de maneira simples. Segundo a própria tradução da documentação:

*Hasura GraphQL Engine* é um servidor *GraphQL* extremamente rápido que fornece instantaneamente, e em tempo real *APIs GraphQL* no Postgres, com *webhook triggers* em eventos de banco de dados, e esquemas remotos para lógica de negócios. (DURAIRAJU, 2020).

Uma das grandes características é que o *Hasura* através do *GraphQL* permite que o desenvolvedor defina no lado do cliente o que será retornado pela *API* em cada instância de comunicação além de disponibilizar o *Hasura Cloud* que permite hospedar este serviço de maneira muito simples e faz toda a configuração do banco de dados através de uma interface interativa o que reduz muito o tempo de implementação da *API*.

### 2.3.2. Frontend

Para ambas partes do sistema, Universidade e Comunidade, será utilizado o *framework Next.js*, que foi desenvolvido a partir da biblioteca mais popular no desenvolvimento de aplicação web denominada *React* criada pela empresa *Facebook* em 2013. Ambas utilizam a mesma linguagem de programação *Javascript*, porém o *Next.js* oferece por padrão *Server Side Rendering*, configurações de rotas, entre outras características. Outra característica importante é que na implementação das interfaces foi utilizado a biblioteca de componentes *Material UI* que auxilia na padronização dos elementos de interface, resultando em uma aplicação padronizada.

O desenvolvimento do sistema da Comunidade foi executado utilizando um conceito denominado *Progressive Web App* (PWA), que prevê um padrão de desenvolvimento que possibilita as plataformas web funcionarem de forma muito semelhante aos aplicativos de dispositivos móveis, ou seja, apesar da execução ser através dos navegadores a apresentação se dá através do modo *FullScreen*, com possibilidade de adição de uma tela de apresentação (tela com o logotipo da empresa) antes abrir o sistema de fato, ícone como atalho para abertura direta do sistema, funcionamento offline, entre outras características que tornam essa aplicação muito similar a um aplicativo nativo. Este conceito se baseia na definição de um arquivo chamado *manifest* que contém diversas informações sobre a aplicação, este arquivo define diversas informações importantes como nome, abreviação do nome da aplicação, cores, ícones e modo de abertura.

### 2.3.3. Firebase

Para a elaboração deste projeto também foram utilizados dois serviços da plataforma *Firebase*: *Authentication* e *Store*. O *Authentication* disponibiliza o serviço de criação de conta, autenticação e recuperação de conta que será utilizado para simular o acesso do aluno e do gestor da universidade. O serviço *Store* é responsável pelo armazenamento das imagens de cada solicitação, ou seja, para cada solicitação criada primeiro será armazenado na *Store*, que retorna uma *URL* de acesso e esta será adicionado ao banco de dados *Postgress*.

## 2.4. Sistema

### 2.4.1. Banco de dados e API

A API de comunicação disponibilizará todas as informações presentes no banco de dados, desta forma a primeira etapa foi a construção de um diagrama que serviu de base para a criação de cada entidade no banco de dados. Este diagrama é apresentado na figura a seguir:

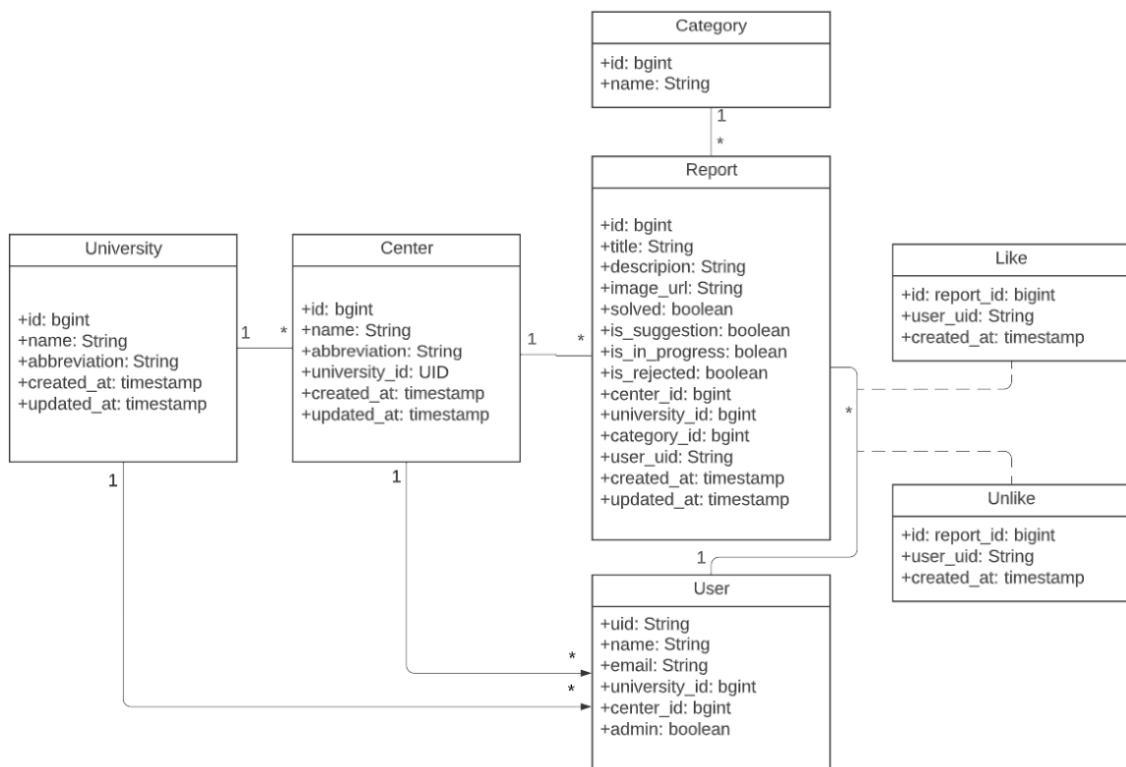


Figura 8. Diagrama UML

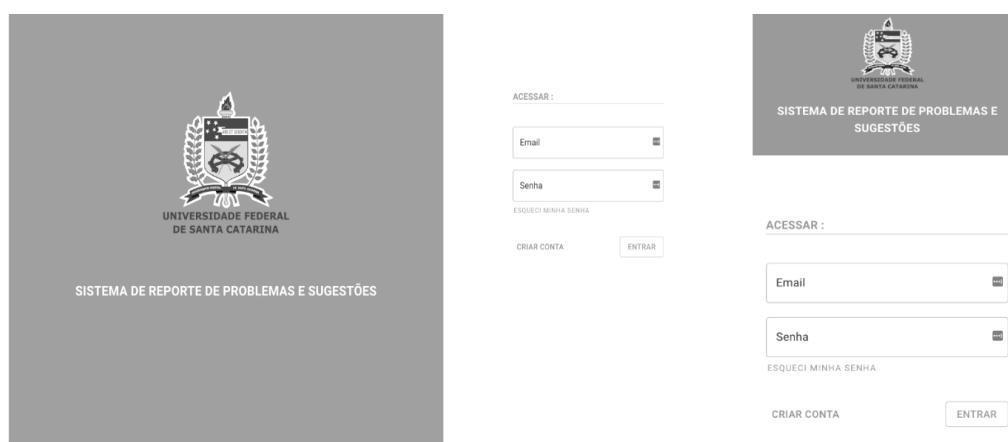
O ponto central deste diagrama é a classe Report que se refere a uma solicitação, que pode ser de dois tipos: sugestão ou um pedido de manutenção, representado no diagrama pela propriedade *is\_suggestion*. Uma solicitação também contém a referência de um usuário (*user\_uid*), de um centro acadêmico (*center\_id*), da categoria (*category\_id*) e de um campus (*university\_id*). Além disso há dados que descrevem a solicitação como: título (*title*), descrição (*description*), a url da imagem (*image\_url*), data de criação (*created\_at*) e de atualização (*updated\_at*) e por fim o seu estado atual (solicitação criada ou resolvida) é representada pela propriedade *solved*.

Para a classe User que contém as informações referentes ao Usuário, seja ele parte da Comunidade ou da Gestão da Universidade, nota-se a ausência de uma propriedade que faça referência à ligação do Usuário com o contexto real da

Universidade, como por exemplo o número da matrícula. Para a elaboração deste sistema idealizou-se a integração com o Sistema de Autenticação da Universidade, porém para o desenvolvimento prático foi necessário utilizar outro sistema de autenticação a fim de simular a experiência de um usuário de forma a obter um sistema usual. Portanto, o sistema de autenticação escolhido para substituir o Sistema de Autenticação Centralizada da UFSC foi o Firebase Authentication citado anteriormente na seção 2.3.3. As classes *University* e *Center* são responsáveis por armazenar as informações referentes às Unidades e Prédios/Áreas respectivamente. A classe *Category* é responsável por armazenar as informações referentes às categorias de solicitações e por fim as classes *Like* e *Unlike* referem-se às interações do Usuário com as solicitações.

## 2.4.2. Acesso ao Sistema

A primeira parte do sistema que o usuário tem contato ao acessar o sistema é a tela de acesso ao sistema que dispõe de três funcionalidades que são: acesso ao sistema, criação de conta de usuário e recuperação de senha. Essas funcionalidades estão presentes no *Firebase* e disponíveis através de uma biblioteca, que expõe os métodos que possibilitam realizar tais ações.



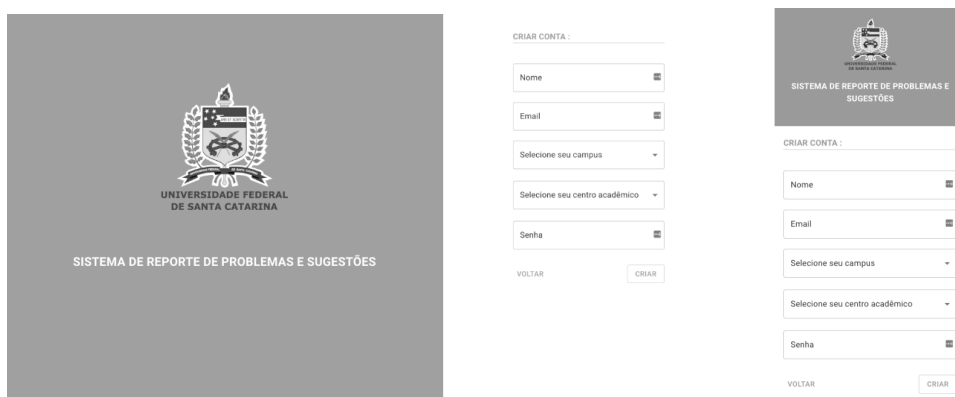
**Figuras 9 e 10. Telas de acesso ao sistema (desktop e mobile)**

O acesso ao sistema se dá através dos dados de *email* e senha do usuário. Ao inserir os dados de acesso, o sistema envia os dados do formulário para o Firebase que realiza a validação dos dados. Caso seja válido, retorna uma resposta com os dados do usuário como o *UID* que é a identificação do usuário no Banco de Dados. Em seguida é feita uma requisição para o Banco a fim de obter os dados do Usuário e persistir esses dados na aplicação e no navegador através dos *cookies*<sup>7</sup>. Isso permite que o usuário permaneça com acesso ativo mesmo após atualizar a página do navegador ou até mesmo fechá-la.

Após a validação, o usuário será redirecionado com base no tipo de conta, através da propriedade *admin* definida na classe User do diagrama UML apresentado na

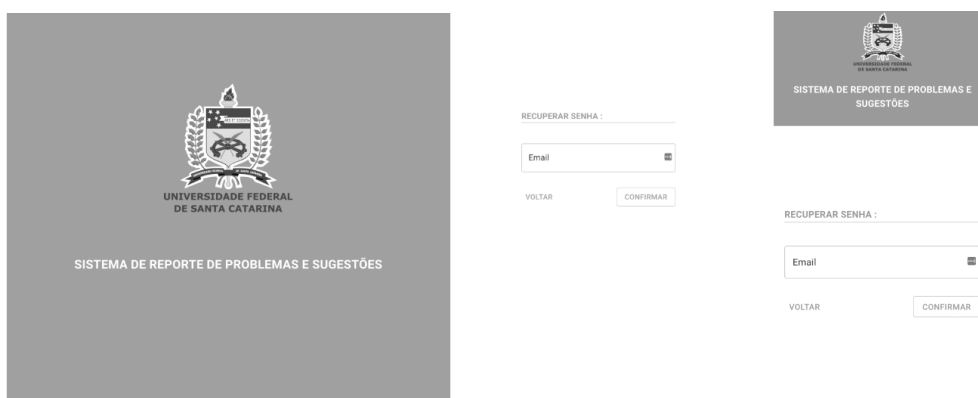
<sup>7</sup> Cookies são uma forma de armazenamento de dados no próprio navegador do usuário;

Figura 8. Caso a conta do Usuário tenha a propriedade admin verificada como verdadeira, ele será redirecionado para o sistema voltado à Universidade, caso contrário será redirecionado para o sistema da Comunidade. Outra funcionalidade presente na tela inicial do sistema é a criação de conta. Ela ocorre em duas etapas, iniciando com o envio do formulário, de forma análoga ao diagrama apresentado anteriormente: a primeira etapa consiste em inserir um *email* e senha. Com isso é realizada a inserção do usuário no *Firebase Authentication* que irá retornar o UID do usuário. A segunda etapa consiste na inserção dos dados do Usuário no Banco de Dados com as outras informações do formulário, como nome, campus e centro acadêmico do usuário. As telas da criação de conta podem ser visualizadas nas Figuras abaixo



**Figura 11 e 12. Telas de criação de conta (desktop e mobile)**

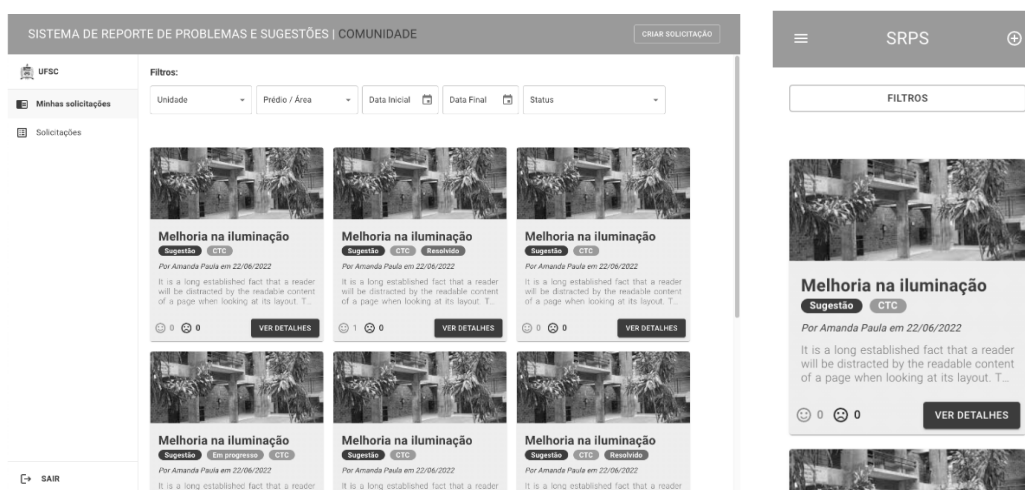
A última funcionalidade é a recuperação de senha, o sistema *Firebase* envia um email contendo um *link* que possibilita a criação de uma nova senha. O envio deste formulário é feito na página do próprio *Firebase*.



**Figuras 13 e 14. Telas de recuperação de conta (desktop e mobile)**

### 2.4.3. Sistema Comunidade

Uma vez que o usuário entra com suas credenciais válidas ele será redirecionado para a parte protegida do sistema, ou seja o sistema interno que permite a criação e interação com as solicitações. Esta seção abordará o sistema voltado para Comunidade que é dividida em três páginas diferentes conforme as imagens a seguir: a página que contém as solicitações criadas pelo próprio usuário, a página que contém as solicitações criadas por toda Comunidade e a página de detalhes de uma solicitação que pode ser acessada através do cartão de solicitação.



Figuras 15 e 16. Comunidade: Minhas solicitações (desktop e mobile)

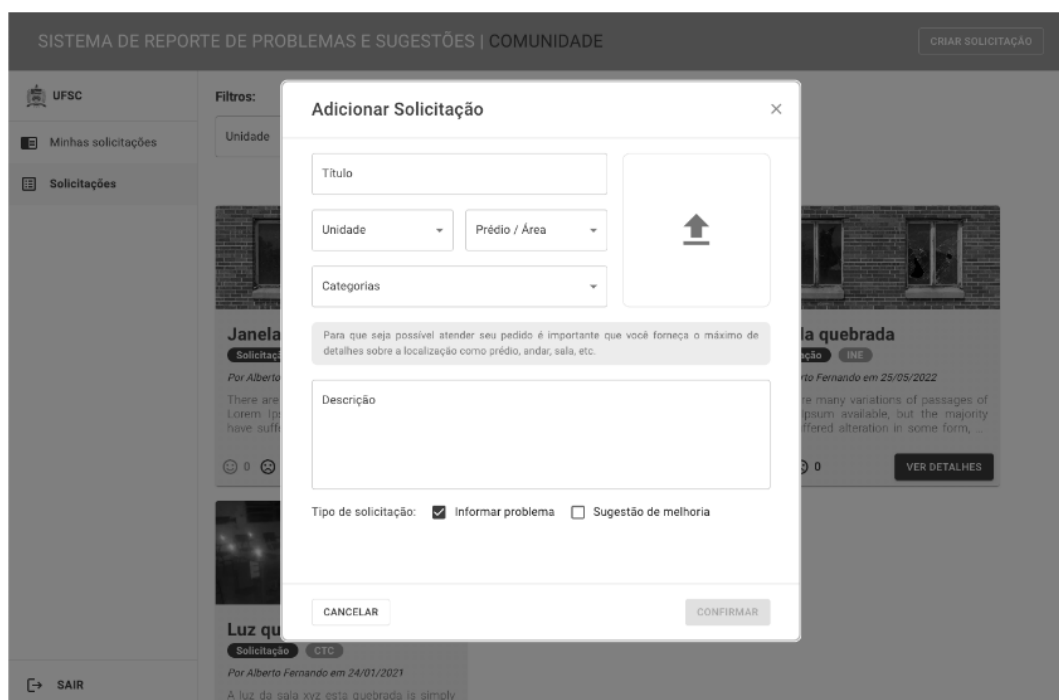


Figura 17. Comunidade: Detalhes da solicitação (desktop)



A página de detalhes da solicitação é onde o usuário tem uma melhor visualização da solicitação, além dos botões de interação que possibilitam votar positivamente, atestando a veracidade e apoiando a solicitação, ou votar negativamente. O usuário pode interagir uma única vez com cada postagem. Além dessas interações, nesta página a imagem para ilustrar o problema é apresentada de forma integral, juntamente com a descrição completa do problema. Diferentemente das páginas que listam as solicitações que apresentam somente as três primeiras linhas da descrição e uma miniatura da imagem.

Outro ponto comum a todas as páginas, como citado anteriormente, é o botão de criação de nova solicitação presente no cabeçalho. A abordagem utilizada para o desenvolvimento desta funcionalidade foi uma caixa de sobreposição, o que possibilita ao usuário criar uma nova solicitação independente da página do sistema em que ele se encontra como apresentado na figura a seguir:



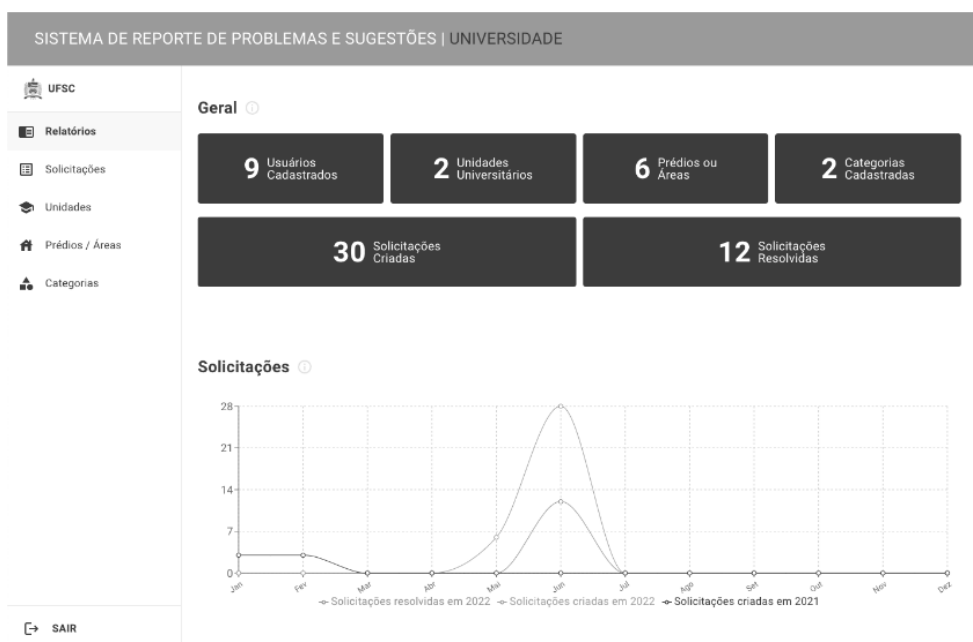
**Figura 18. Comunidade: Criação de solicitação (desktop)**

As páginas de detalhes de solicitação e listagem de solicitações também estão presentes no sistema voltado à Universidade, uma característica da listagem voltada para a comunidade é que na opção "Minhas Solicitações" o sistema carrega somente as solicitações criadas pelo usuário logado no sistema independente do *status* (criado, em progresso, rejeitado ou finalizado) atual da solicitação. Na página de "Solicitações", o sistema carrega todas as solicitações que não foram criadas pelo usuário logado no sistema e que não tenham sido finalizadas (rejeitada ou finalizada) ainda, ou seja, somente as solicitações com *status* criado ou em progresso.

## 2.4.4. Sistema Universidade

O sistema voltado para a gestão da Universidade é composto por seis páginas que possibilitam a manutenção das solicitações criadas pela comunidade, sendo elas as páginas: Relatório, Solicitações, Detalhes da solicitação, Unidades, Prédios/Áreas e Categorias.

A página de relatório visa fornecer à Universidade dados referente às solicitações criadas pela comunidade e dados gerais da plataforma. Ao acessar esta parte do sistema, são realizadas duas grandes requisições para a API que permitem acessar os dados referentes a todo o período, como o número total de usuários cadastrados na plataforma, categorias, unidades, prédios e áreas, solicitações criadas e solicitações resolvidas. Também é computado um comparativo entre o ano atual e o ano anterior, referente ao número de solicitações criadas por mês, sendo este apresentado na forma gráfico. Ambas seções de dados gerais e de comparação anual das solicitações são apresentadas na figura a seguir.



**Figura 19. Universidade: Relatórios (Dados gerais e Gráfico de solicitações)**

Na sequência, são apresentados dados do ano atual através de quatro gráficos de setores referentes ao número de solicitações por: unidades, prédios e áreas, categorias e solicitações resolvidas. O gráfico de setores possibilita interação ao passar o cursor no setor desejado, apresentando ao usuário o número de solicitações em cada setor e a porcentagem em relação ao todo.



Figura 20. Universidade: Relatórios (Gráficos de setores)

Por fim, o gestor tem acesso à duas tabelas, que podem ser visualizadas abaixo. A primeira contém as últimas dez solicitações criadas, as categorias que está inserida e um botão que redireciona para a página de detalhes da solicitação. A segunda tabela apresenta as dez solicitações criadas com mais confirmações feitas pela comunidade contendo, de maneira análoga, as categorias que cada solicitação está inserida e o botão de acesso à página de detalhes.

SISTEMA DE REPORTE DE PROBLEMAS E SUGESTÕES   UNIVERSIDADE					
Últimas 10 solicitações ou sugestões criadas					
Data	Unidade	Prédio/Área	Categoria	Título	Ver Mais
21/06/2022	UFSC	INE	Iluminação	Lâmpada queimada	Ver Mais
21/06/2022	UFSC	INE	Iluminação	Lâmpada queimada	Ver Mais
23/06/2022	UFSC	INE	Iluminação	1231	Ver Mais
23/06/2022	UFSC	INE	Iluminação	1231	Ver Mais
23/06/2022	UFSC	CTC	Iluminação	Módulo	Ver Mais
23/06/2022	UFSC	CTC	Iluminação	Módulo	Ver Mais
23/06/2022	UFSC	CTC	Iluminação	Melhoria na iluminação	Ver Mais
23/06/2022	UFSC	CTC	Iluminação	Melhoria na iluminação	Ver Mais
23/06/2022	UFSC	CTC	Iluminação	Melhoria na iluminação	Ver Mais
23/06/2022	UFSC	CTC	Iluminação	Melhoria na iluminação	Ver Mais

SISTEMA DE REPORTE DE PROBLEMAS E SUGESTÕES   UNIVERSIDADE					
Últimas 10 solicitações ou sugestões criadas					
Data	Unidade	Prédio/Área	Categoria	Título	Ver Mais
21/06/2022	UFSC	INE	Iluminação	Lâmpada queimada	Ver Mais
21/06/2022	UFSC	INE	Iluminação	Lâmpada queimada	Ver Mais
23/06/2022	UFSC	INE	Iluminação	1231	Ver Mais
23/06/2022	UFSC	INE	Iluminação	1231	Ver Mais
23/06/2022	UFSC	CTC	Iluminação	Módulo	Ver Mais
23/06/2022	UFSC	CTC	Iluminação	Módulo	Ver Mais
23/06/2022	UFSC	CTC	Iluminação	Melhoria na iluminação	Ver Mais
23/06/2022	UFSC	CTC	Iluminação	Melhoria na iluminação	Ver Mais
23/06/2022	UFSC	CTC	Iluminação	Melhoria na iluminação	Ver Mais
23/06/2022	UFSC	CTC	Iluminação	Melhoria na iluminação	Ver Mais

Figuras 21 e 22. Universidade: Relatórios (Tabelas)

As páginas de solicitações e de detalhes de uma solicitação foram implementadas de maneira similar ao apresentado na seção 2.4.1. Há uma diferença na página de detalhe da solicitação referente à interação com a solicitação: o gestor não pode votar positivamente ou negativamente. As únicas interações possíveis são marcar a solicitação como resolvida, rejeitada ou em progresso.

Por fim, as páginas de unidades, prédios/áreas e categorias são acessíveis através do menu lateral e possibilitam o cadastro de novas opções para a comunidade e também a exclusão de opções previamente criadas, desde que não tenham sido utilizadas como referência para alguma solicitação ou usuário no caso das unidades e prédios/áreas. Abaixo estão representadas respectivamente as páginas de unidades, prédios/áreas e categorias:

SISTEMA DE REPORTE DE PROBLEMAS E SUGESTÕES | UNIVERSIDADE

Unidades + ADICIONAR UNIDADE

ID	Sigla	Nome	
1	UFSC	Universidade Federal de Santa Catarina	⌵
5	JOINVILLE	Universidade Feral de Santa Catarina	⌵

Linhas por página: 100 1-2 de 2 < >

SAIR

**Figura 23. Universidade: Unidades**

SISTEMA DE REPORTE DE PROBLEMAS E SUGESTÕES | UNIVERSIDADE

Prédios / Áreas + ADICIONAR PRÉDIO OU ÁREA

ID	Sigla	Nome	Unidade	
1	CTC	Centro Tecnológico	UFSC	⌵
2	INE	Departamento de Informática e Estatística	UFSC	⌵
3	EQA	Engenharia Química e Alimentos	UFSC	⌵
4	CCE	Centro de Comunicação e Expressão	UFSC	⌵
5	CSE	Centro Socioeconômico	UFSC	⌵
6	CCS	Centro de Ciências da Saúde	UFSC	⌵

Linhas por página: 100 1-6 de 6 < >

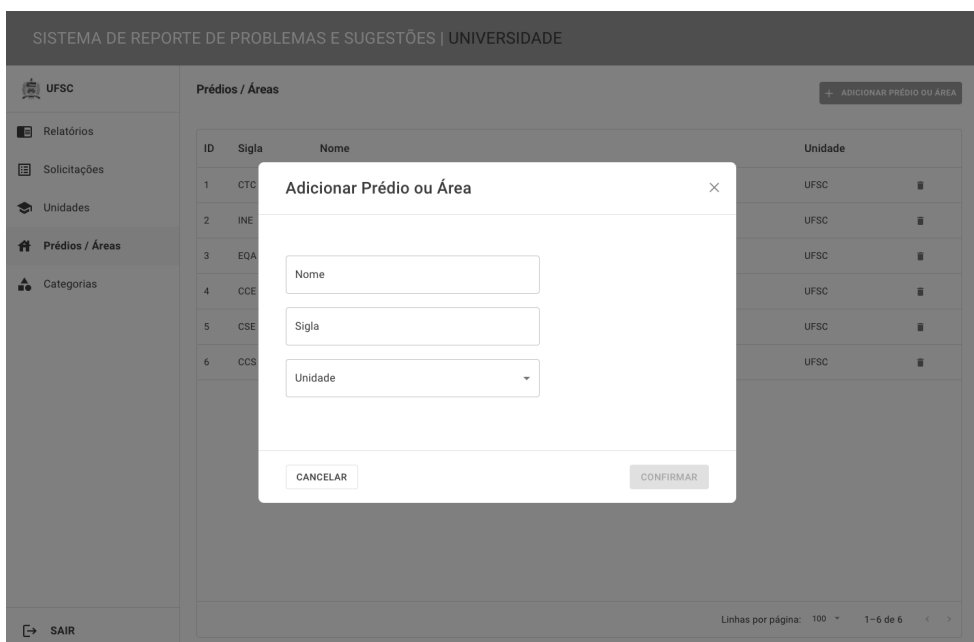
SAIR

**Figura 24. Universidade: Prédios/Áreas**



**Figura 25. Universidade: Categorias**

De forma análoga à criação de solicitação, o botão para criar uma nova opção de campi, centro ou categorias abrirá uma janela sobreposta para que não seja necessário redirecionar o usuário para uma nova página, como ilustrado a seguir a tela de criação de um novo prédio ou área:



### 3. Conclusão

O presente trabalho elaborou um protótipo de sistema para auxiliar na gestão da manutenção de uma universidade a partir do estudo de sistemas de participação colaborativa já existentes, como a aplicação do modelo Open311 nos sistemas oferecidos pela Connected Bits e FixMyStreet. Também foram analisadas algumas iniciativas desenvolvidas nacionalmente, como as plataformas Colab e Participact. Um fator importante para a realização deste trabalho, foi a compreensão das diferentes estruturas organizacionais dentro das instituições de ensino a fim de elaborar um modelo que permita às universidades autonomia para tratar as solicitações.

O desenvolvimento deste protótipo usual requereu conhecimento sobre as arquiteturas e tecnologias disponíveis no mercado atual. Foi escolhido o framework *NextJS* que utiliza a linguagem Javascript para desenvolver o *frontend* da aplicação. Para o *backend* foi escolhido o framework *Hasura*, que por sua vez utiliza o banco de dados *PostgressSQL* e a linguagem de consulta *GraphQL*. O serviço *Authentication* da plataforma *Firebase* foi utilizado para realizar a autenticação, a criação e recuperação de contas no sistema. O serviço *Store* desta mesma plataforma foi utilizado para o armazenamento das imagens das solicitações. Todas as decisões necessitaram do estudo de diversas opções para que fosse possível integrar as tecnologias escolhidas a fim de produzir um produto funcional.

Por fim, espera-se que o presente trabalho possa servir de modelo para a criação de um projeto real dentro das universidades e que justifique a importância de uma atuação efetiva na manutenção de todo espaço comum, visto que esta efetividade disponibilizará mais recursos para serem aplicados no que há de mais importante dentro das instituições de ensino: a educação.

#### 3.1. Sugestão para trabalhos futuros

O protótipo desenvolvido neste trabalho resulta em uma aplicação funcional, porém acredita-se que novas funcionalidades resultariam em um sistema ainda mais completo. São elas:

- Integração com um sistema de autenticação da própria universidade, fazendo com que não seja necessária a criação de uma nova conta e seja vinculada à matrícula do estudante ou servidor universitário;
- Um algoritmo de recomendação para apresentar aos membros da comunidade as solicitações, baseado nas interações de outros usuários ou na localização do indivíduo;
- Desenvolvimento de uma aplicação mobile para a comunidade que possibilite interação com outras funcionalidades dos dispositivos móveis, como o acesso à localização do usuário;
- Funcionalidade de comentários para as solicitações, que permitam a interação entre os membros da comunidade.

### 3. Referências

JULNES, Patria De Lancer; GIBSON, Ed. Innovation in the Public and Nonprofit Sectors: A Public Solutions Handbook. 1. ed. USA: [s.n.], 2015. [Online; accessed 16-dezembro-2021] Disponível em: <https://books.google.com.br/books?id=zK40CwAAQBAJ&pg=PT105&lpg=PT105&d#v=onepage&q&f=false>.

BOSTON'S Mayor's Office of New Urban Mechanics. [S.l.: s.n.], 2019. [Online; accessed 12-março-2022]. Disponível em: <https://www.centreforpublicimpact.org/case-study/bostons-mayors-office-new-urban-mechanics>.

DURAIRAJU, P. Hasura GraphQL Engine - GitHub. [Online; accessed 27-junho-2022] Disponível em: [https://github.com/hasura/graphql-engine/blob/master/translations/README.portuguese\\_br.md](https://github.com/hasura/graphql-engine/blob/master/translations/README.portuguese_br.md).

DRAGONI, Nicola; GIALLORENZO, Saverio; LAFUENTE, Alberto Lluch; MAZZARA, Manuel; MONTESI, Fabrizio; MUSTAFIN, Ruslan; SAFINA, Larisa. Microservices: Yesterday, Today, and Tomorrow. [S.l.: s.n.], 2017. [Online; accessed 22-junho-2022]. Disponível em: [https://link.springer.com/chapter/10.1007/978-3-319-67425-4\\_12](https://link.springer.com/chapter/10.1007/978-3-319-67425-4_12).

## APÊNDICE B – CÓDIGO FONTE

Os códigos fontes, resultados do protótipo desenvolvido neste trabalho, foram compactados em dois arquivos *.zip* referentes a aplicação *frontend* e *backend*. Ambos foram anexados no sistema de arquivos da UFSC. É possível realizar o *download* dos códigos através das seguintes URLs:

- Aplicação NextJS (*frontend*): <https://arquivos.ufsc.br/f/6bd4f2f07f9c4f829677/>
- Aplicação Hasura (*backend*): <https://arquivos.ufsc.br/f/7cf88cd9d5c64e41a66f/>