



**UNIVERSIDADE FEDERAL DE CAMPINA GRANDE
CENTRO DE ENGENHARIA ELÉTRICA E INFORMÁTICA
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO**

LUAN ROCHA SILVA

SISTEMA PARA GESTÃO DE PROJETOS URBANOS

CAMPINA GRANDE - PB

2023

LUAN ROCHA SILVA

SISTEMA PARA GESTÃO DE PROJETOS URBANOS

Trabalho de Conclusão Curso apresentado ao Curso Bacharelado em Ciência da Computação do Centro de Engenharia Elétrica e Informática da Universidade Federal de Campina Grande, como requisito parcial para obtenção do título de Bacharel em Ciência da Computação.

Orientador : Tiago Lima Massoni

CAMPINA GRANDE - PB

2023

LUAN ROCHA SILVA

SISTEMA PARA GESTÃO DE PROJETOS URBANOS

Trabalho de Conclusão Curso apresentado ao Curso Bacharelado em Ciência da Computação do Centro de Engenharia Elétrica e Informática da Universidade Federal de Campina Grande, como requisito parcial para obtenção do título de Bacharel em Ciência da Computação.

BANCA EXAMINADORA:

Tiago Lima Massoni

Orientador – UASC/CEEI/UFCG

Hyggo Oliveira de Almeida

Examinador – UASC/CEEI/UFCG

Francisco Vilar Brasileiro

Professor da Disciplina TCC – UASC/CEEI/UFCG

Trabalho aprovado em: 28 de JUNHO de 2023.

CAMPINA GRANDE - PB

RESUMO

Pequenas prefeituras normalmente fazem uso de empresas privadas para realização de serviços relacionados ao monitoramento da situação do Imposto Predial e Territorial Urbano (IPTU) de imóveis, ou seja, terceirizam estes serviços. Uma dessas empresas sentiu a necessidade da criação de um sistema web que viabilizasse o cadastro e manutenção dos dados que são coletados, permitindo uma otimização no gerenciamento dessas informações. Utilizando o sistema será possível concentrar todos os dados em um único local, abolindo assim a utilização e acúmulo de planilhas.

URBAN PROJECT MANAGEMENT SYSTEM

ABSTRACT

Small town administration normally resorts to private companies to carry out services related to monitoring the status of the Urban Property Tax of properties. One of these companies felt the need to create a web system that would allow the recording and maintenance of the collected data, allowing optimization in the management of this information. With the system, it will be possible to concentrate all the data in a single place, thus dispensing with the use and accumulation of spreadsheets.

Sistema para Gestão de Projetos Urbanos

Luan Rocha Silva

Universidade Federal de Campina Grande

luan.silva@ccc.ufcg.edu.br

Tiago Lima Massoni

Universidade Federal de Campina Grande

massoni@dsc.ufcg.edu.br

RESUMO

Pequenas prefeituras normalmente fazem uso de empresas privadas para realização de serviços relacionados ao monitoramento da situação do Imposto Predial e Territorial Urbano (IPTU) de imóveis, ou seja, terceirizam estes serviços. Uma dessas empresas sentiu a necessidade da criação de um sistema web que viabilizasse o cadastro e manutenção dos dados que são coletados, permitindo uma otimização no gerenciamento dessas informações. Utilizando o sistema será possível concentrar todos os dados em um único local, abolindo assim a utilização e acúmulo de planilhas.

PALAVRAS-CHAVE

Projeto urbano, sistema de gestão, cidades, imóveis.

LINKS ÚTEIS

<https://github.com/luanrsilva/projeto-urbano>

1. INTRODUÇÃO

Uma empresa de engenharia, que trabalha com georreferenciamento, definindo a forma, dimensão e localização de imóveis e propriedades rurais, através de métodos de levantamento topográfico, necessita criar e manter diversas planilhas com os dados de imóveis e cidades diferentes. Esta atividade envolve muitas informações, tornando-se algo custoso e cansativo de ser feito em planilhas, além de acarretar uma elevada probabilidade de ocorrência de erros, custando ainda mais tempo para revisões e possíveis correções.

Tendo em vista os problemas e custos acima citados, a empresa se deparou com a necessidade de um sistema onde fosse possível realizar o cadastro desses imóveis e ter acesso a esses dados na nuvem, para melhor gerenciar os projetos lá desenvolvidos.

Ferramentas de tecnologia são conhecidas em apoiar diversos tipos de atividades profissionais. Essas ferramentas podem auxiliar no controle e gerenciamento de quantidades significativas de dados e entre outras tantas coisas, concentrando todas essas informações em um único ambiente.

A solução oferecida por este trabalho, visa disponibilizar uma aplicação web, que permita o armazenamento de todos os dados coletados na nuvem e como consequência, possibilite uma melhor visualização e organização do que foi coletado.

2. PROBLEMA

A Ponto Fixo Engenharia, fundada em 2021, é uma empresa de Campina Grande que trabalha com certificação de imóveis, buscando parcerias com prefeituras de pequenas cidades

para auxiliar na verificação de conformidade dos imóveis das cidades, com relação aos valores pagos de IPTU (Imposto sobre a Propriedade Predial e Territorial Urbana ou Imposto Predial e Territorial Urbano) por parte dos proprietários.

A iniciativa para a coleta dos dados das cidades parte da própria empresa, que busca pequenas cidades que terceirizam serviços relacionados à essa questão. A empresa coleta dados de área dos terrenos, áreas contruídas, estima valor por metro quadrado de setores da cidade e calcula o que deveria ser pago de IPTU por cada propriedade. Após a organização dos dados, estes são vendidos às prefeituras, para que possam verificar a situação dos imóveis.

O principal problema a ser combatido é o acúmulo de arquivos utilizados para manter todos esses dados coletados, pois para cada cidade são criadas planilhas imensas, com mais de 1000 (um mil) linhas, contendo as informações adquiridas sobre cada imóvel, como ilustra a Figura 1. Todo esse material guardado vai ficando separado em diversas pastas e ocupando espaço no computador, tornando mais custosos a manutenção e gerenciamento desses dados. Fazendo uso do sistema será possível reunir todas as informações, facilitando o acesso, oferecendo uma melhor visualização e extinguindo os arquivos acumulados.

ÁREA DO TERRENO				VALORES			
IDENTIFICADOR	NUMERO DE CASA	ÁREA TERRENO (m²)	ÁREA CONSTRUIDA (m²)	VALOR M²	VALOR DO IMÓVEL	SETOR	Alíquota do IPTU
1	394	1341,6527	673,3796	R\$ 399,79	R\$ 536,39532	SETOR 3	R\$ 536,40
2	394	80,009	87,6662	R\$ 399,79	R\$ 33,295271	SETOR 3	R\$ 33,31
3	396	136,2267	83,9979	R\$ 399,79	R\$ 54,48227	SETOR 3	R\$ 54,49
4	394	212,3521	159,5029	R\$ 399,79	R\$ 135,95276	SETOR 1	R\$ 135,95
5	394	137,7703	123,8971	R\$ 399,79	R\$ 56,079119	SETOR 3	R\$ 56,08
6	394	128,4234	119,1535	R\$ 399,79	R\$ 47,346393	SETOR 3	R\$ 47,35
7	394	180,0841	180,0841	R\$ 399,79	R\$ 71,99582	SETOR 3	R\$ 72,00
8	396	236,4580	236,4580	R\$ 399,79	R\$ 94,54924	SETOR 3	R\$ 94,55
9	392	200,8381	124,1317	R\$ 399,79	R\$ 80,23326	SETOR 3	R\$ 80,23
10	396	26,4698	21,8189	R\$ 399,79	R\$ 8,512,79	SETOR 3	R\$ 8,52
11	399	203,676	203,676	R\$ 399,79	R\$ 80,98064	SETOR 3	R\$ 80,97
12	394	105,115	100,6645	R\$ 399,79	R\$ 42,02383	SETOR 3	R\$ 42,02
13	394	135,1022	122,8638	R\$ 399,79	R\$ 52,88264	SETOR 3	R\$ 52,89
14	394	140,2775	109,4743	R\$ 399,79	R\$ 56,08154	SETOR 3	R\$ 56,08
15	394	212,3521	159,5029	R\$ 399,79	R\$ 134,76021	SETOR 1	R\$ 134,75
16	398	218,3664	132,3715	R\$ 399,79	R\$ 87,30070	SETOR 3	R\$ 87,30
17	394	293,429	263,4569	R\$ 399,79	R\$ 118,56071	SETOR 3	R\$ 118,51
18	398	112,2638	87,2942	R\$ 399,79	R\$ 47,68027	SETOR 3	R\$ 47,68
19	396	124,6380	114,6690	R\$ 399,79	R\$ 50,55582	SETOR 3	R\$ 50,56
20	398	186,8158	141,1749	R\$ 399,79	R\$ 74,68739	SETOR 3	R\$ 74,69
21	394	180,118	114,8678	R\$ 399,79	R\$ 72,88841	SETOR 3	R\$ 72,89
22	399	103,5113	103,5113	R\$ 399,79	R\$ 41,37229	SETOR 3	R\$ 41,37
23	394	90,7477	80,7477	R\$ 399,79	R\$ 36,28030	SETOR 3	R\$ 36,28
24	399	86,4796	86,4796	R\$ 399,79	R\$ 34,43336	SETOR 3	R\$ 34,43
25	396	86,485	82,783	R\$ 399,79	R\$ 34,18429	SETOR 3	R\$ 34,16
26	394	108,814	104,814	R\$ 399,79	R\$ 41,90339	SETOR 3	R\$ 41,90
27	394	136,6556	94,2364	R\$ 399,79	R\$ 54,23375	SETOR 3	R\$ 54,23
28	399	83,2021	83,2021	R\$ 399,79	R\$ 33,15428	SETOR 3	R\$ 33,15
29	394	134,111	134,111	R\$ 399,79	R\$ 53,51251	SETOR 3	R\$ 53,51

Figura 1 - Exemplo de planilha

3. SOLUÇÃO

O objetivo final deste trabalho é desenvolver uma aplicação web que seja capaz de solucionar o problema citado. Por meio desta aplicação, o pessoal da empresa Ponto Fixo Engenharia será capaz de cadastrar os dados de cidades, propriedades e proprietários.

A aplicação visa ser responsiva e simples, com capacidade de executar em qualquer navegador web através de smartphones e desktops.

Utilizando a aplicação será possível organizar e manter de maneira mais eficiente os dados cadastrados, além de facilitar o acesso e manuseio, pois basta que o usuário acesse o sistema por

meio de alguma rede disponível, já que os dados serão salvos na nuvem.

3.1 Funcionalidades

A aplicação poderá ser acessada por meio de um usuário administrador padrão, criado pelo sistema. Ao realizar o login, o usuário é redirecionado para uma página que lista as cidades já cadastradas no sistemas, como mostra a Figura 2 abaixo. Nessa página o usuário terá a opção de cadastrar uma nova cidade e poderá também clicar em uma cidade específica para ver mais detalhes sobre a mesma.



Cidade	Alíquota IPTU	Setores
Brasília	0.8	2
Campina Grande	0.6	3
Ouro Velho	0.2	1

Figura 2 - Listagem de cidades

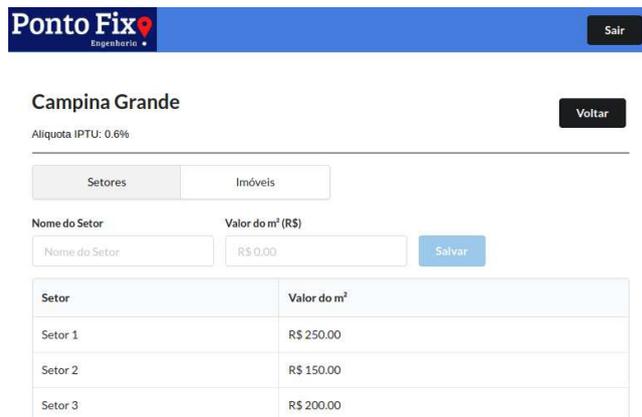
Ao clicar no botão de adicionar, o usuário é levado a uma página onde ele deverá informar o nome da nova cidade e a alíquota IPTU que é cobrada nela, como mostra a Figura 3. Ao clicar em salvar, o usuário retorna para a página que lista as cidades e pode observar a nova cidade já listada.



Figura 3 - Cadastro de nova cidade

Clicando em uma determinada cidade da lista, o sistema redireciona para uma página de detalhes dessa cidade, contendo as abas de Setores e Imóveis, onde o usuário poderá navegar entre elas e ter visualizações diferentes. Na aba de Setores será possível cadastrar novos, informando o nome do setor e o valor em reais

do metro quadrado dessa área, e ver a lista dos que já foram cadastrados nessa cidade, como mostra a Figura 4.



Setor	Valor do m ²
Setor 1	R\$ 250.00
Setor 2	R\$ 150.00
Setor 3	R\$ 200.00

Figura 4 - Lista de setores de uma cidade

Indo para a aba de Imóveis, o usuário poderá ver a listagem dos imóveis já cadastrados e terá a opção de adicionar novos, como mostra a Figura 5.



Identificador	Nº da casa	Área do Terreno(m ²)	Área do Construída(m ²)	Valor Estimado	IPTU
0	925	120	80	R\$ 24,000	R\$ 144

Figura 5 - Lista de imóveis

Na aba de adicionar um novo imóvel, como mostra a Figura 6, o usuário deverá informar um número identificador para o imóvel, número da casa, a área do terreno, o setor da cidade onde está localizado, área construída, nome e CPF do proprietário. Ao salvar, a aplicação irá calcular os valores estimados dos imóveis e seus IPTUs e esses dados poderão ser utilizados posteriormente para checar se os proprietários dos imóveis estão em conformidade com as prefeituras.

Ponto Fixo
Engenharia

Campina Grande

Alíquota IPTU: 0.6%

Setores Imóveis

Cadastrados Adicionar

Identificador: 0001 Número da Casa: 925 Área do Terreno (m²): 120 Setor: Setor 1

Área Construída (m²): 80 Nome do Proprietário: Luan Rocha CPF do Proprietário: 000.000.000-00

Salvar

Figura 6 - Cadastro de imóvel

3.2 Arquitetura

Foi utilizada a arquitetura Cliente-Servidor[1] para o desenvolvimento desta aplicação. Neste modelo de arquitetura, o processamento da informação é dividido em módulos distintos. O cliente neste caso é o frontend, responsável pela exibição da interface e pela obtenção dos dados do servidor. Este por sua vez é responsável pelo gerenciamento e manutenção dos dados, composto pelo banco de dados e pela API. A Figura 1 ilustra essa comunicação entre os módulos.

De acordo com o modelo C4[2] de documentação para arquitetura de software, o diagrama da Figura 7 ilustra os containers da aplicação. O diagrama mostra que no container geral da aplicação estão o frontend e o container da API, onde estão presentes o banco de dados e a API REST. A comunicação entre as partes também é representada no diagrama. Como exemplo de uma requisição, podemos citar a tela de exibição de uma cidade específica, onde ao acessar uma cidade, o frontend faz uma requisição para o backend contendo o ID da cidade desejada, o back faz a busca da cidade que possui o ID informado no banco de dados e devolve o objeto encontrado para ser exibido no front.

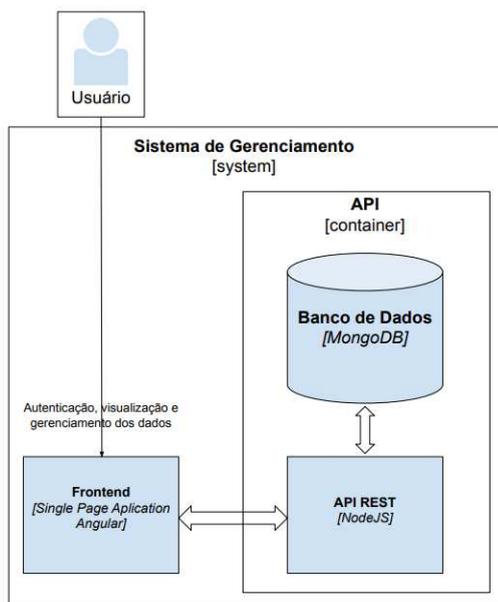


Figura 7 - Diagrama de Containers

3.3 Tecnologias utilizadas no servidor

Devido a familiaridade com as tecnologias, foram utilizados NodeJS e Express[3] no servidor. O NodeJS é um ambiente de código aberto que permite ferramentas do lado do servidor usando JavaScript. Já o Express é uma estrutura que fornece diversos recursos para a construção de aplicações web, facilitando configurações como a determinação dos modelos que serão utilizados nas respostas ou a porta que será usada para estabelecer conexão.

A familiaridade com a tecnologia também foi responsável pela definição de qual ferramenta ficaria com a responsabilidade da persistência dos dados. A escolhida foi o MongoDB[4], banco de dados orientado a documentos, projetado para armazenar uma grande quantidade de dados e com muita eficiência em enormes quantidades de informações.

Para a documentação dos endpoints criados foi utilizado o Swagger[5], que é um framework usado para descrição, consumo e visualização de serviços e gera a documentação automaticamente com base em anotações do código. A imagem da Figura 8 a seguir exibe a estrutura do endpoint de cadastro de uma nova propriedade com o comentário responsável por gerar a documentação do mesmo.

```

router.post("/", async (req, res) => {
  await propertyService.createProperty(req.body, (response) => {
    res.status(response.status).send(response);
    /* #swagger.tags = ['Property']
       #swagger.description = 'Endpoint to add a property.' */
  });
});
  
```

Figura 8 - Requisição POST de propriedade

3.4 Estrutura da API

Uma API é um conjunto de definições e protocolos usado no desenvolvimento e na integração de aplicações. Às vezes podem ser descritas como um contrato entre um provedor e um usuário de informações, estabelecendo o conteúdo exigido pelo consumidor (a chamada) e o conteúdo exigido pelo produtor (a resposta).

Seguindo a ordem dos endpoints exibidos abaixo na Figura 9, para a entidade Setor, relacionada à US04, temos as rotas de criação e recuperação de um setor específico. Para a entidade Cidade, temos as rotas relacionadas à US03, para criação de uma nova cidade, listagem das cidades cadastradas, recuperação de uma cidade específica e recuperação dos setores e propriedades de uma determinada cidade. E para a entidade Propriedade, que diz respeito à US05, existem as rotas de criação e recuperação de uma propriedade específica.

Sector	
POST	/api/sector/
GET	/api/sector/{id}
City	
POST	/api/city/
GET	/api/city/
GET	/api/city/{id}
GET	/api/city/{id}/sectors
GET	/api/city/{id}/properties
Property	
POST	/api/property/
GET	/api/property/
GET	/api/property/{id}

Figura 9 - Documentação da API

A estrutura da API, exibida a seguir na Figura 10, foi montada de acordo com experiências anteriores do autor, separando por pasta os tipos dos arquivos. Na pasta *bootstrap* está o arquivo responsável pela criação de um usuário padrão, que será o perfil de administrador do sistema. Na pasta *config* estão os arquivos para configuração do token de acesso, conexão com o banco de dados e padronização das respostas da API. Ainda na mesma pasta, temos um arquivo que concentra todas as rotas de acesso do sistema. Dentro da pasta *models* estão os arquivos de definição dos modelos para o banco de dados. A pasta *routers* contém os arquivos que definem os endpoints de acesso da API. Na pasta *services* estão os arquivos que se relacionam com o banco de dados e são responsáveis pelas lógicas no tratamento dos dados. Os arquivos relacionados a documentação da API foram colocados na pasta *doc*. Os arquivos *package*.json* contém as dependências necessárias para o funcionamento do sistema e o *index.js* é o arquivo responsável pela execução.

```

  v src
    v bootstrap
      JS admin.bootstrap.js
    v config
      {} auth.json
      JS mongoose.js
      JS responses.js
      JS router.js
    v doc
      {} swagger-output.json
      JS swagger.js
    v models
      JS admin.model.js
      JS blacklist-token.model.js
      JS city.model.js
      JS property.model.js
      JS sector.model.js
      JS user.model.js
    v routers
      JS city.controller.js
      JS login.controller.js
      JS property.controller.js
      JS sector.controller.js
    v services
      JS city.service.js
      JS login.service.js
      JS property.service.js
      JS sector.service.js
      JS index.js
  {} package-lock.json
  {} package.json

```

Figura 10 - Estrutura da API

3.5 Tecnologias utilizadas no frontend

Para o desenvolvimento das telas do sistema foi utilizado Angular[6], uma plataforma de aplicações web baseado na linguagem de programação TypeScript e liderado pela Equipe Angular do Google e por uma comunidade de indivíduos e corporações. O Angular é uma reescrita do AngularJS, feito pela mesma equipe. Essa tecnologia, a versão doze (12) sendo mais específico, foi escolhida considerando a experiência do autor em outros projetos que utilizaram a mesma ferramenta e pensando na estrutura baseada em componentes fornecida pela ferramenta, o

que facilita o desenvolvimento da aplicação. Além disso, foi utilizado *Bootstrap*[8] para auxiliar na estilização dos componentes das telas.

3.6 Tecnologias do sistema em uso

Na realização do deploy e disponibilização do sistema para homologação, foi utilizada uma instância do AWS EC2[9] com a plataforma Amazon Linux. A instância é do tipo t2.micro, com 1CPU e 1GB de memória. Como é um tipo mais simples, este serviço é oferecido gratuitamente pela Amazon, incluindo 750 horas de instâncias todo mês durante um ano.

Para o frontend foi utilizado o build do projeto Angular juntamente com o NGINX[10], que é um servidor leve de HTTP. A configuração é simples, pois basta instalar o serviço na instância e definir qual arquivo será exibido.

Já para o backend foi utilizado docker[11] e docker-compose, criando os containers para a API e para o banco de dados e permitindo a comunicação entre eles.

3.7 Sistema em uso

A aplicação foi disponibilizada durante dois dias. Tendo acesso ao sistema, um usuário realizou testes de login e cadastros de novas cidades, setores e imóveis.

Após o período de testes, foi solicitado ao usuário que respondesse algumas perguntas. A tabela abaixo exhibe quais foram essas perguntas e suas respectivas respostas:

O login funciona corretamente?	SIM
Foi possível cadastrar novas cidades, setores e imóveis?	SIM
Houve algum problema durante a utilização do sistema? Qual?	Sim, mas apenas alguns problemas de responsividade em dispositivos móveis
De 1 a 5 o quanto você acha que o sistema pode contribuir com suas atividades?	4

Foram resultados satisfatórios, pois todas as funcionalidades propostas funcionaram corretamente, tendo apenas alguns problemas de responsividade sendo reportados. A respeito da pergunta sobre a contribuição do sistema, onde não foi obtido o valor máximo, o usuário foi questionado quanto a isso e respondeu que para ele o que determinou esse valor foi a falta da funcionalidade de salvar dados offline, que já foi citado como limitação e incluído em trabalhos futuros em tópicos anteriores neste trabalho.

4. EXPERIÊNCIA

Aqui serão apresentadas as experiências do autor em relação ao processo de desenvolvimento, desafios, limitações e trabalhos futuros.

4.1 Processo de desenvolvimento

O processo de desenvolvimento foi iniciado com a apresentação da ideia pela Ponto Fixo Engenharia. Foram realizadas reuniões para melhor entendimento da demanda, coleta de requisitos e definição de quais eram as principais necessidades da empresa. Como resultante dessas reuniões, o autor produziu um *product backlog*, que norteou toda a implementação.

O autor definiu o Scrum[7] como metodologia ágil para o desenvolvimento, adotando *sprints* de uma semana, onde, ao final de cada *sprint*, era obtido um incremento para o sistema. Para ter o controle das versões do código foi utilizado o Github.

As *sprints* foram separadas como mostra a tabela abaixo e as *User Stories* foram priorizadas dessa forma, pois seguem o fluxo do cadastro de novos imóveis.

SPRINT 1	SPRINT 2	SPRINT 3
US01 - Setup de desenvolvimento US02 - Eu como usuário desejo me autenticar para ter acesso ao sistema	US03 - Eu como usuário desejo cadastrar uma nova cidade para poder gerenciá-la US04 - Eu como usuário desejo cadastrar um novo setor para poder gerenciá-lo	US05 - Eu como usuário desejo cadastrar um novo imóvel/terreno para poder gerenciá-lo

4.2 Desafios

O principal desafio enfrentado no desenvolvimento deste trabalho está relacionado a disponibilização do sistema para o usuário. Pela falta de experiência do autor com tecnologias utilizadas para este fim, foram feitas várias tentativas utilizando diversas ferramentas de uso comum como Heroku, Google Cloud Plataforma, Máquinas Virtuais e serviços da AWS.

Tendo em vista que o autor encontrou mais materiais de tutoriais e exemplos utilizando o Amazon EC2, decidiu adotar este serviço para a realização da atividade. Inicialmente tentou criar containers, usando docker e docker-compose, para todas as partes da aplicação, porém teve problemas com o container do frontend, que usava o NGINX como imagem base na criação, pois os containers eram criados e executados com sucesso, mas ao tentar acessar o IP da instância na porta definida nada era exibido. Após algumas pesquisa e observando alguns exemplos de uso do NGINX, o autor decidiu remover o container do frontend e tentar implantar essa parte do sistema na própria instância, instalando o NGINX e alterando o arquivo de configuração para exibir o arquivo desejado. Depois dessas configurações, o autor obteve sucesso ao tentar acessar a aplicação.

Como o autor só conseguiu realizar o deploy da aplicação após a finalização do desenvolvimento, as revisões das sprints por parte do cliente eram feitas por meio de vídeo chamadas com compartilhamento de tela ou encontros presenciais, onde a evolução do sistema era exibida na própria máquina do autor. Durante essas reuniões, o cliente apontava as correções que achava necessárias e estas eram realizadas nas sprints seguintes.

4.3 Limitações

Inicialmente, o uso da aplicação está limitado à conexão com a internet, não sendo possível cadastrar novas entidades estando sem conexão. Este ponto é algo a ser considerado em trabalhos futuros, pois como o sistema será usado em campo, há lugares mais remotos em algumas cidades onde a conexão com a rede é mais instável ou até não seja possível se conectar. Portanto, seria interessante poder salvar alguns dados offline e realizar o upload desses dados após conseguir uma conexão estável.

Ademais, no período em que o sistema foi disponibilizado para homologação, foram encontrados também alguns pequenos problemas de responsividade em dispositivos móveis, como por exemplo o tamanho do texto de alguns botões. A correção dessas falhas será incluída em trabalhos futuros e feita posteriormente.

4.4 Trabalhos futuros

Para o futuro, além da possibilidade de salvar dados offline e de correções de responsividade, como foi citado no tópico anterior, já foram mapeadas algumas funcionalidades, como a implementação de um módulo para viabilizar o download de um documento contendo as informações dos imóveis de uma cidade. Outra atividade seria poder adicionar novos usuários, que terão acesso específico a determinadas cidades e poderão apenas visualizar e extrair as informações existentes. Há também a possibilidade do surgimento de novas funcionalidades.

5. AGRADECIMENTOS

Aos amigos que iniciaram o curso e participaram de toda essa jornada comigo. Todos me ajudaram muito compartilhando conhecimento e tornando mais leves os dias na universidade. Ao professor Tiago Massoni, principalmente por sua paciência e disponibilidade como meu orientador. É uma excelente pessoa e um profissional exemplar dentro do ambiente acadêmico. Aos engenheiros da Ponto Fixo Engenharia, por me concederem permissão de utilizar a ideia desenvolvida neste trabalho. Ao meu pai Assis, minha irmã Letícia e minha esposa Anna Carolina, por todo apoio e suporte no meu caminho até aqui.

Por fim, os agradecimentos mais importantes vão para minha mãe Suely e meu tio Eduardo, ambos falecidos, que nunca duvidaram de mim e sempre me incentivaram nos momentos de maior fraqueza.

6. REFERÊNCIAS

- [1] Arquitetura cliente-servidor. **Canalti**, 2018. <<https://www.canalti.com.br/arquitetura-de-computadores/arquitetura-cliente-servidor/>>. Acesso em 13 de Agosto de 2022
- [2] O modelo C4 de documentação para Arquitetura de Software. **Simon Brown**. Disponível em: <<https://www.infoq.com/articles/C4-architecture-model/>>. Acesso em 13 de Agosto de 2022
- [3] Diferenças Node.JS e Express.JS. **Kristin Savage** <<https://blog.back4app.com/pt/diferencas-node-js-e-express-js/>>. Acesso em 23 de Janeiro de 2023

[4] O que é e para que serve o MongoDB? **Leandro Kovacs**

<<https://tecnoblog.net/responde/o-que-e-e-para-que-serve-o-mongodb/>>. Acesso em 23 de Janeiro de 2023

[5] Porque usar Swagger na sua API? **Aristoteles Lopes** <<https://medium.com/@arikardnoir/porque-usar-swagger-na-sua-api-e80c4ed15190>>. Acesso em 23 de Janeiro de 2023

[6] Introduction to the Angular docs <<https://angular.io/docs>>. Acesso em 11 de Março de 2023

[7] O que é Scrum? <<https://aws.amazon.com/pt/what-is/scrum/#:~:text=O%20Scrum%20%C3%A9%20um%20framework,uma%20entrega%20eficiente%20de%20projetos.>> Acesso em 14 de Março de 2023

[8] Get started with Bootstrap. Disponível em: <<https://getbootstrap.com/docs/5.3/getting-started/introduction/>>

[9] Amazon EC2. Disponível em: <<https://aws.amazon.com/pt/ec2/>>

[10] NGINX. Disponível em: <<https://www.nginx.com/>>

[11] Docker. Develop faster. Run anywhere. Disponível em: <<https://www.docker.com/>>