UNIVERSIDADE FEDERAL DO PARÁ

FACULDADE DE COMPUTAÇÃO

CURSO DE BACHARELADO EM SISTEMAS DE INFORMAÇÃO

CARLOS HENRIQUE DE OLIVEIRA FERREIRA

SEA POINT:
Sistema de automação de projetos de compartilhamento de infraestrutura

Campus Universitário Castanhal

Castanhal – Pará – Brasil

2019

UNIVERSIDADE FEDERAL DO PARÁ

FACULDADE DE COMPUTAÇÃO

CURSO DE BACHARELADO EM SISTEMAS DE INFORMAÇÃO

CARLOS HENRIQUE DE OLIVEIRA FERREIRA

SEA POINT:
Sistema de automação de projetos de compartilhamento de infraestrutura

Trabalho de Conclusão de Curso apresentado como requisito parcial para obtenção do título de Bacharel em Sistemas de Informação da Faculdade de Computação.

Orientador: Prof. Dr Igor Ruiz Gomes

Campus Universitário Castanhal

Castanhal – Pará – Brasil

2019

RESUMO

O presente trabalho intitulado **SEA POINT: Sistema de automação de projetos de compartilhamento de infraestrutura** tem como objetivo geral: desenvolver, testar e implantar uma ferramenta no cenário atual de provedores de internet regionais, visando uma melhoria na implantação de redes de fibra óptica. Os objetivos específicos são: facilitar a coleta de dados por meio de um processo simples, confiável e de maior exatidão; agilizar a disponibilidade das informações utilizando sincronismo através da internet; permitir acompanhamento dos avanços de projetos em tempo real. A metodologia dessa pesquisa se dá pela apresentação das ferramentas que serão utilizadas durante a execução do sistema proposto, como: Desenvolvimento Android, OpenStreetMap, ReactJS, etc.. Uma vez completa a versão inicial do sistema foram realizados testes internos para homologação das funcionalidades e avaliação dos resultados, de modo que os usuários se habituaram ao uso diário do aplicativo, com a última leitura sendo de 71 usuários ativos mensalmente, com aproximadamente 100% de estabilidade do sistema. Com o uso do sistema desenvolvido nesse trabalho, todo o processo de coleta de informações é unificado entre as obrigações do operador que captura as informações, o administrador que elabora e envia o projeto e a concessionária que analisa e faz a aprovação do mesmo.

Palavras-Chaves: Compartilhamento de infraestrutura. Provedores de internet. Fibra óptica.

ABSTRACT

The following work titled SEA POINT: infrastructure sharing project automation has the general objective to: develop, test and deploy a tool in the current internet service providers scenario. The specific objectives are: make it easier to collect data using a simples process, trustworthy and more precise; speed up the availability of data using internet based synchronization; allow supervision of the project progression in real time. The research methodology is done by presenting the tools used during the execution of the proposed system, for example: Android Development, OpenStreetMap, ReactJS, etc... Once the first version is finished there was internal testing for functionality and results validation, this way the users get used to daily use and the last reading was 71 active monthly users with nearly 100% system stability. Using the system developed by this work, the entire process of data gathering is unified within the operator's information capture and the administrator that prepare and send the project to the infrastructure owner for analysis and approval.

Keywords: infrastructure sharing, internet provider, optical fiber.

LISTA DE ILUSTRAÇÕES

Figura 1:Método de gerenciamento de projetos Scrum 7
Figura 2: Representação gráfica da função do Sync Gateway 13
Figura 3:Diagrama de caso de uso exibindo administrador e operador do sistema 19
Figura 4: Distribuição da base de usuário conforme versão do Android 21
Figura 5: Visão geral do sistema 23
Figura 6: Processo de Login 24
Figura 7: Processo de captura de ponto 25
Figura 8: Tela do projeto no painel web 26
Figura 9: REST endpoints do back-end web 27
Figura 10: Pontos geográficos coletados com o SEA Point 28
Figura 11: Estatísticas de usuários ativos no sistema 29
Figura 12: Distribuição da base de usuário por fabricante 30
Figura 13: Distribuição da base de usuários por versão do Android 31

LISTA DE TABELAS

Tabela 1: Métodos HTTP 8
Tabela 2: Código de retorno HTTP 9
Tabela 3:Operações REST..... 10
Tabela 4:Requisitos funcionais..... 17
Tabela 5:Requisitos não funcionais..... 18

LISTA DE ABREVIATURAS E SIGLAS

HTTP	Hypertext Transfer Protocol
REST	Representational State Transfer
JSON	JavaScript Object Notation
DBMS	Database Management System
DOM	Document Object Model
CAPEX	Capital Expenditure
OPEX	Operational Expenditure
GPS	Global Positioning System

SUMÁRIO

Abstract 4

Lista de ilustrações 5

Lista de tabelas 6

Lista de abreviaturas e siglas 7

1 Introdução 1

1.1 Problematização 1

1.2 Solução proposta 1

1.3 Objetivo geral 2

1.4 Objetivos específicos 2

1.5 Metodologia de pesquisa 3

1.6 Organização da monografia 3

2 Fundamentação teórica 5

2.1 Processo de solicitação 5

2.2 Scrum 5

2.2.1 Papeis 6

2.2.2 Eventos 6

2.2.3 Artefatos 6

2.3 HTTP (*Hypertext Transfer Protocol*) 7

2.4 REST (*Representational State Transfer*) 9

2.5 JSON (JavaScript Object Notation) 10

3 Tecnologias e recursos utilizados 11

3.1 Desenvolvimento Android 11

3.2 OpenStreetMap 11

3.3 ReactJS 11

3.4 Couchbase 12

3.4.1 Database 12
3.4.2 Sync Gateway 12
3.4.3 Couchbase Lite 13
3.5 MySQL 14
3.6 ASP.NET Core 14
4 Desenvolvimento 15
4.1 Definição de escopo 15
4.2 Elicitação de requisitos 15
4.2.1 Requisitos funcionais 15
4.2.2 Requisitos não funcionais 18
4.3 Diagramas de caso de uso 18
4.4 Metodologia 19
4.5 Codificação 20
4.5.1 Aplicativo mobile 20
4.5.2 Painel web 21
4.5.3 Web API 21
5 Resultados obtidos 23
5.1 Aplicativo mobile 23
5.1.1 Login 23
5.1.2 Coleta de ponto 24
5.2 Web (*front-end*) 25
5.3 Web (*back-end*) 26
6 Avaliação de resultados 28
6.1 Teste e feedback 28
6.2 Perfis de usuários 29
6.3 Resultados de Usabilidade 29
7 Considerações finais 32

7.1 Dificuldades encontradas 32

7.2 Trabalhos futuros 33

Referências 34

1 INTRODUÇÃO

Telecomunicação é uma das áreas da tecnologia da informação com maior demanda de inovação para atender os fluxos de dados cada vez maiores, afinal hoje em dia tudo está conectado na internet, sejam smartphones, televisões e até mesmo lâmpadas. Assim, para atender essa demanda o uso de redes de fibra óptica (FTTx) capazes de altas velocidades de transmissão é cada vez mais presente.

O processo de implantação dessas redes é extremamente complexo e possui várias etapas desde fazer planejamento, passando pela implantação e manutenção. A fase de planejamento é a base para todo o projeto e nela está a elaboração do projeto de compartilhamento de infraestrutura, nele a empresa que está implantando a rede fornece informações à concessionária detentora da infraestrutura da qual se deseja fazer uso, normalmente essa estrutura consiste em postes para lançamento de cabos de fibra óptica aéreos e as informações são os postes que se deseja utilizar e as especificações dos equipamentos e cabos que serão instalados.

Em um projeto desse tipo normalmente a empresa interessada no uso da estrutura coleta informações, como pontos geográficos do poste (latitude e longitude), características do poste (tipo, altura, esforço), acessórios (transformadores, bancos de capacitores, ...) e atuais ocupantes do poste. Para isso, normalmente uma equipe é deslocada para o local com uma prancheta e algum dispositivo capaz de capturar pontos de GPS para coletar as informações necessárias.

1.1 Problematização

A lentidão na execução de cada etapa de elaboração do projeto causa além da demora de disponibilização e atualização dos projetos, prejuízos financeiros, uma vez que se faz necessária a alocação de mão de obra para execução e eventuais correções, o que gera custos operacionais com pessoas, deslocamento e equipamentos.

A irregularidade na ocupação dos postes também é um problema crescente, estimasse que dos 46 milhões de postes em todo o país, apenas 9 milhões estejam com sua ocupação regularizada. (AGÊNCIA ESTADO, 2018)

1.2 Solução proposta

Com base no cenário apresentado, foi elaborado um projeto composto por vários subsistemas que possui o objetivo de desburocratizar, organizar, padronizar e tornar os

processos de solicitação do lado do requerente, e análise, do lado do permissionário eficaz, eficiente e fidedigno.

O SEA Point é um sistema composto por várias ferramentas que facilitam o trabalho desde o operador em campo coletando as informações, passando pela operadora que fará uso da estrutura, até a concessionária detentora da infraestrutura compartilhada. Cada um com as ferramentas adequadas para coleta, documentação e controle onde cada parte é beneficiada.

1.3 Objetivo geral

O objetivo deste trabalho é desenvolver, testar e implantar uma ferramenta no cenário atual de provedores de internet regionais. Com o intuito de automatizar o processo de elaboração de projetos de compartilhamento de infraestrutura, e, coletar informações de como essas ferramentas melhoram a execução dos projetos de implantação de redes de fibra ótica.

Além disso, gerar impacto financeiro positivo proporcionado pela economia em OPEX, que é o orçamento que descreve as despesas operacionais necessárias para manter ou melhorar as atividades da empresa.

Outro impacto esperado em maior escala é ampliar a regularização da ocupação dos pontos de fixação (postes) uma vez que isso traria maior controle por parte da concessionária e qualidade do serviço prestado ao usuário final.

1.4 Objetivos específicos

- Facilitar a coleta de dados por meio de um processo simples, confiável e de maior exatidão.
- Agilizar a disponibilidade das informações utilizando sincronismo através da internet.
- Possibilitar o uso do aplicativo em aparelhos *low end* e em condições onde a conectividade com a internet estiver instável ou indisponível.
- Permitir acompanhamento dos avanços de projetos em tempo real.
- Agilizar a entrega das informações coletadas no formato exigido para elaboração e aprovação do projeto.

1.5 Metodologia de pesquisa

A metodologia desta pesquisa se dá pela apresentação das ferramentas que serão usadas durante a execução do sistema proposto. Utilizando técnicas que possibilitarão a resolução dos problemas referentes à gestão de informação, precisão na coleta de dados e agilidade. Visto que, anteriormente o colhimento dessas informações era feito de forma manual, o que resultava em um mau armazenamento das mesmas por conta da falta de suporte que havia durante a coleta.

Durante o desenvolvimento do sistema, foram usadas ferramentas de telemetria, para coleta das estatísticas de usabilidade e funcionalidade, visando um melhor entendimento dos respectivos pontos em que o sistema teria dificuldade de operação ou instabilidade durante sua execução. As ferramentas utilizadas para a produção do sistema foram selecionadas de acordo com sua usabilidade em empresas atuais focadas em inovações, como Facebook, Microsoft e Linkedin, etc.. Outro fator que contribuiu para essa escolha foi o seu teor moderno, visto que são bastante utilizadas na atualidade.

As ferramentas que serviram de base para o desenvolvimento do sistema foram as seguintes: Desenvolvimento Android, OpenStreetMap, ReactJS, Couchbase, Database, Sync Gateway, Couchbase Lite, MySQL e ASP.NET Core. A respeito das mesmas, elas serão explicadas detalhadamente de acordo com sua usabilidade no projeto na sessão intitulada **Recursos utilizados.**

1.6 Organização da monografia

O restante deste trabalho está dividido em seis sessões conforme descrito na lista abaixo:

- Fundamentação teórica: Essa sessão faz uma revisão do embasamento teórico de várias áreas do desenvolvimento de sistemas e regras de compartilhamento de infraestrutura considerados importantes para o desenvolvimento deste trabalho.
- Tecnologias e recursos utilizados: Essa sessão apresenta algumas ferramentas, tecnologias e recursos tecnológicos utilizados durante o desenvolvimento para entregar uma versão mínima viável da solução proposta nesse trabalho.
- Desenvolvimento: Essa sessão descreve o emprego das ferramentas e metodologias no processo de produção da documentação e desenvolvimentos dos softwares descritos na solução proposta.

- Resultados obtidos: Essa sessão faz uma análise dos resultados obtidos após o término da fase de desenvolvimento, descrevendo cada sistema e suas funcionalidades conforme atendem ou não o proposto.
- Avaliação de resultados: Essa sessão apresenta a fase de testes e coleta de métricas junto à base de usuários de testes, discutindo se os objetivos projetados foram alcançados de forma satisfatória e os pontos mais importantes observados de usabilidade e funcionalidade.
- Considerações finais: Essa sessão faz uma apresentação do trabalho de forma geral, discutindo os objetivos citados nos capítulos anteriores e demonstrando como os objetivos foram atingidos ou não. Também menciona as dificuldades encontradas e trabalhos futuros planejados a partir dos resultados obtidos.

2 FUNDAMENTAÇÃO TEÓRICA

Para que fosse possível o planejamento e execução deste projeto, foram feitas diversas pesquisas em várias áreas do desenvolvimento de software. Nessa sessão é feita uma revisão bibliográfica sobre os temas para elaboração do projeto de desenvolvimento da solução proposta.

2.1 Processo de solicitação

A prestadora que deseja solicitar o compartilhamento de infraestrutura deve observar as regras estabelecidas pela Resolução Conjunta nº 1/1999 (Aneel, Anatel e ANP), os passos estão descritos nessa Resolução[1].

Em resumo a solicitação de compartilhamento de postes (pontos de fixação) se dá da seguinte maneira:

1. A prestadora de telecomunicações deve entrar em contato com a distribuidora de energia elétrica para se informar sobre a disponibilidade de infraestrutura e as condições de compartilhamento.
2. As condições apresentadas pela distribuidora não podem ser discriminatórias e devem abranger informações sobre preços e prazos.
3. A solicitação de compartilhamento deve ser feita formalmente pelo meio indicado pela distribuidora, que solicitará as informações necessárias para a análise de viabilidade técnica.
4. Caso a distribuidora aprove o compartilhamento, ela emitirá o contrato, que será analisado pela Anatel e enviado à Aneel para homologação (aprovação).
5. O art. 20 da Resolução Conjunta 1/1999 contém cláusulas essenciais que deverão constar no contrato e que são indispensáveis para o parecer favorável à homologação.

2.2 Scrum

Visando uma maior agilidade na produção e execução do projeto SEA point, o Scrum foi escolhido para contribuir na elaboração do sistema, com o intuito de obter um melhor desenvolvimento do mesmo. A saber, Scrum é uma metodologia ágil de gestão e planejamento de projetos de software onde os projetos são divididos em ciclos chamados de Sprint que são unidades de trabalho que levam normalmente de 24h a 1

[1] (ANEEL, ANATEL e ANP, 1999)

mês e resultam em uma versão incremental do software (SCHWABER e SUTHERLAND, 2017)

2.2.1 Papeis

Os papeis do Scrum são atribuições assumidas por grupos integrantes do time e são definidos conforme a seguir:

- **Product Owner**: É responsável por definir as tarefas e prioridades que virão a compor o backlog.
- **Time de Desenvolvimento**: O Time de Desenvolvimento consiste em profissionais que realizam o trabalho de entregar uma versão usável que potencialmente incrementa o produto "Pronto" ao final de cada Sprint.
- **Scrum Master**: O Scrum Master é responsável por garantir que o Scrum seja entendido e aplicado. O Scrum Master faz isso para garantir que o Time Scrum adere à teoria, práticas e regras do Scrum.

2.2.2 Eventos

- **Sprint**: É um evento que dura um mês ou menos onde o time de desenvolvimento executa todas as tarefas selecionadas no Backlog da Sprint, do qual resulta uma versão incremental potencialmente utilizável do produto. Uma Sprint inicia logos após a conclusão da Sprint anterior.
- **Reunião de Planejamento da Sprint**: Os itens que compõem o Backlog da Sprint e como esse trabalho será executado é definido na reunião de planejamento da Sprint. Todo o time Scrum pode participar dessa reunião.
- **Reunião Diária**: É um evento do Scrum que ocorre diariamente e tem duração média de 15 minutos, nela são discutidas as atividades concluídas pelo time de desenvolvimento e planejadas as atividades para as próximas 24 horas até a próxima reunião.
- **Revisão da Sprint**: A revisão da Sprint é uma atividade executada ao final da Sprint para analisar o incremento e adaptar o backlog de acordo com os ajustes necessários.

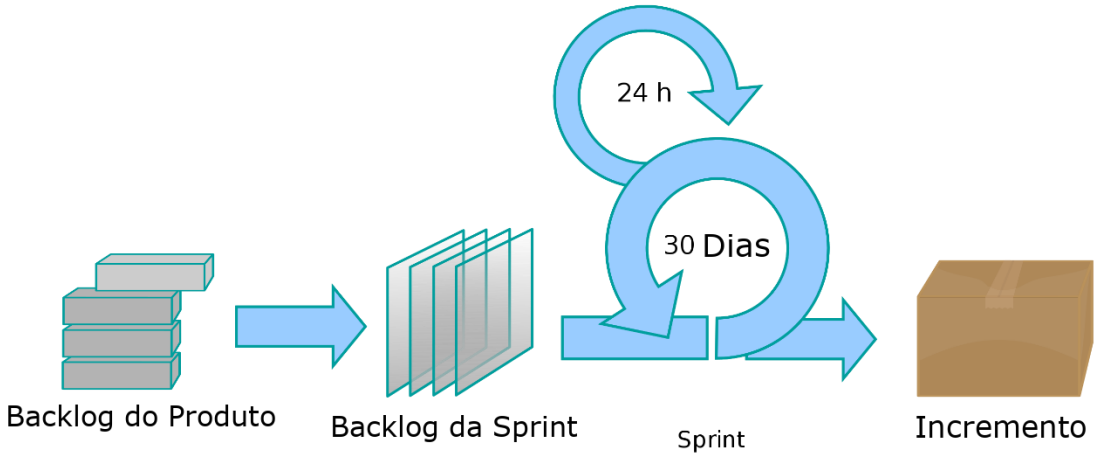
2.2.3 Artefatos

- **Backlog do Produto**: O Backlog do produto é uma lista dos requisitos que devem fazer parte do produto finalizado, qualquer alteração nos requisitos ou

atividades deve ser feita diretamente nessa lista que é gerenciada pelo *Product Owner*.

- **Backlog da Sprint**: O Backlog da Sprint é um subconjunto dos itens do Backlog do produto selecionados para a Sprint e que devem resultar no incremento.
- **Incremento**: O incremento é resultado do trabalho executado na Sprint completa, inclui todos os itens da Sprint e é uma versão incremental utilizável do sistema.

Figura 1:Método de gerenciamento de projetos Scrum



Fonte: Adaptação do autor

Na figura 1 mostrada acima, podemos ver detalhadamente a representação gráfica dos eventos que ocorrem no Scrum e seus respectivos papéis durante o processo de desenvolvimento.

Uma versão adaptada às circunstâncias do desenvolvimento foi utilizada para entregar o sistema proposto, uma vez que não havia profissionais suficientes para integrar um time Scrum completo e a agenda de execução do projeto não estava firmemente definida, isto é, possuía alguma flexibilidade.

2.3 HTTP (*Hypertext Transfer Protocol*)

HTTP é um protocolo de aplicação desenvolvido para sistemas de informações de *hypermedia* e é a base de toda a comunicação na *World Wide Web*. O HTTP funciona

como um protocolo de requisição-resposta em um ambiente cliente-servidor onde o cliente executa uma aplicação que faz requisições ao servidor, o cliente envia uma mensagem de requisição que representa algum recurso que se deseja obter ou operação que se deseja executar, o servidor então envia uma mensagem de resposta ao cliente indicando o resultado da requisição e o recurso obtido (FIELDING, ET AL., 1999).

Na tabela 1 são apresentados os tipos de mensagens que o cliente pode enviar ao servidor de modo a obter uma resposta que pode conter o recuso que se deseja ter acesso ou o resultado de uma operação executada no servidor.

Tabela 1: Métodos HTTP

GET	O método GET requisita uma representação do recurso especificado. Requisições usando GET devem apenas recuperar dados e não devem ter qualquer outro efeito.
HEAD	Variação do GET em que o recurso não é retornado
POST	Envia dados para serem processados para o recurso especificado. Os dados são incluídos no corpo do comando.
PUT	O método PUT envia os dados de forma semelhante ao POST, através do corpo do HTTP a diferença entre os 2 métodos é que o PUT atualiza um recurso existente ao invés de criar um
DELETE	Exclui o recurso.
TRACE	Ecoa o pedido, de maneira que o cliente possa saber o que os servidores intermediários estão mudando em seu pedido.
OPTIONS	Recupera os métodos HTTP que o servidor aceita.
CONNECT	Serve para uso com um proxy que possa se tornar um túnel SSL e TLS
PATCH	Atualiza parcialmente o recurso

De acordo com a requisição enviada, o servidor responde com uma mensagem que contém o resultado da operação, além do conteúdo da mensagem existe também um código que indica a natureza da mensagem sendo enviada de volta para o cliente, essa mensagem pode conter vários significados conforme exibido na tabela 2.

Tabela 2: Código de retorno HTTP

1xx	Informational (Informação) – utilizada para enviar informações para o cliente de que sua requisição foi recebida e está sendo processada.
2xx	Success (Sucesso) – indica que a requisição do cliente foi bem-sucedida.
3xx	Redirection (Redirecionamento) – informa a ação adicional que deve ser tomada para completar a requisição.
4xx	Client Error (Erro no cliente) – avisa que o cliente fez uma requisição que não pode ser atendida.
5xx	Server Error (Erro no servidor) – ocorreu um erro no servidor ao cumprir uma requisição válida.

Neste projeto, o protocolo HTTP é utilizado como meio de transporte das informações trocadas entre a API *back-end* e o painel web apresentado ao usuário.

2.4 REST (*Representational State Transfer*)

REST é uma arquitetura criada em cima do protocolo HTTP (*Hypertext Transfer Protocol*) utilizada para criação de *web services* ela utiliza os métodos CRUD presentes no protocolo para definir uma interface de manipulação textual de recursos web, sistemas que implementam essa arquitetura são chamados de serviços RESTful (WORLD WIDE WEB CONSORTIUM, 2004).

Recuso web são definidos na *World Wide Web* por documentos ou arquivos que podem ser identificados por sua URL (Uniform Resource Locator).

As operações CRUD (*Create, Read, Update, Delete*) são definidas por verbos que representam a ação que se deseja tomar sobre o recurso web, no protocolo HTTP esses verbos são descritos na tabela 3.

Tabela 3:Operações REST

URL exemplo	GET	PUT	POST	DELETE
http://api.exemplo.com/colecao	Lista os itens presente na coleção	Substitui todos os itens da coleção com itens novos	Cria uma coleção	Apaga toda a coleção
http://api.exemplo.com/colecao/item	Obtém a representação do item específico na coleção	Substitui apenas o item identificado pela URL	Não utilizado	Exclui apenas o item identificado na coleção

2.5 JSON (JavaScript Object Notation)

JSON é um formato de troca de dados leve e simples de escrever e entender, baseado na linguagem de programação Java Script que utiliza elementos da mesma para representar dados, porém não possui nenhuma dependência da linguagem (2019).

JSON é normalmente utilizado para transportar dados simples em ambientes onde há comunicação tipo cliente-servidor, por ser simples de entender torna fácil o trabalho do desenvolvedor de depurar e identificar possíveis problemas nos dados enviados.

Neste projeto JSON é utilizado como formato de dados na comunicação entre os subsistemas além de ser o formato de armazenamento das informações na base de dados NoSQL.

3 TECNOLOGIAS E RECURSOS UTILIZADOS

3.1 Desenvolvimento Android

Android é uma plataforma mobile aberta e gratuita desenvolvida pelo Google, sua natureza *open source* e compatibilidade com uma imensa variedade de dispositivos lhe proporcionou grande popularidade entre desenvolvedores e usuários de smartphones. Por esses motivos essa plataforma foi escolhida para desenvolver o aplicativo utilizado para captura das informações em campo, isto é, dados dos postes.

Para desenvolver o aplicativo foi utilizada o SDK nativo do sistema baseado na linguagem Java além de outras bibliotecas que estendem suas funcionalidades de acordo com os requisitos do sistema.

O aplicativo mobile SEA Point faz uso dos seguintes recursos do dispositivo para funcionar corretamente:

- Acesso à internet
- Geolocalização (Posição GPS)
- Câmera
- Armazenamento interno

3.2 OpenStreetMap

O OpenStreetMap é um projeto colaborativo de mapeamento aberto que fornece dados a várias aplicações sejam elas em plataformas web, mobile ou desktop.

O OpenStreetMap é desenvolvido por uma comunidade voluntária de mapeadores que contribuem e mantêm atualizados os dados sobre estradas, trilhos, cafés, estações ferroviárias e muito mais por todo o mundo. É constituído por dados abertos, ou seja, qualquer pessoa tem a liberdade de usar os dados para qualquer fim desde que credite a autoria do OpenStreetMap e os seus contribuidores.

3.3 ReactJS

ReacJS é uma biblioteca Java Script para desenvolvimento de aplicações web *front-end* baseada em componentes para a construção de aplicações do tipo SPA (*single page applications*).

Desenvolvido pelo Instagram e posteriormente adquirido pelo Facebook, é uma biblioteca que tem ganhado muita popularidade nos últimos anos.

O ReactJS tem ganhado bastante popularidade no desenvolvimento de aplicações web e recentemente aplicação mobile através do *React Native*. Se destaca pela arquitetura e performance proporcionada pelo chamado Virtual DOM.

Virtual DOM é uma técnica de representação do DOM real do documento na memória e armazenada no formato de objetos Java Script, quando alguma alteração é feita no documento apenas o Virtual DOM é atualizado e quando as alterações são concluídas um algoritmo calcula as diferenças entre o DOM atual e o novo aplicando apenas as partes relevantes.

3.4 Couchbase

Sistema de gerenciamento de banco de dados (DBMS) *open source* de alto desempenho baseado em NoSQL, utiliza o conceito de documentos para armazenar os dados e não impõe um esquema de dados como nos bancos de dados relacionais baseados em tabelas, o que permite mais flexibilidade. O Couchbase se divide em vários módulos, os utilizados no desenvolvimento desse projeto foram os seguintes:

3.4.1 Database

Roda em um servidor dedicado e é responsável por cuidar do armazenamento e recuperação de dados além de manter um histórico de revisões com todas as alterações efetuadas nos documentos, possui uma linguagem de consulta própria baseada em SQL chamada N1QL que é utilizada para realizar consultas de maneira relacional como nos bancos de dados tradicionais.

3.4.2 Sync Gateway

Figura 2: Representação gráfica da função do Sync Gateway

Fonte: Couchbase[2]

É a ponte entre o banco de dados e os clientes mobile, o Couchbase Sync Gateway é responsável por autorizar e garantir o sincronismo e integridade das bases de dados remotas.

Graças a um protocolo próprio chamado BLIP baseado em WebSockes o Sync Gateway é capaz de sincronizar dados próximo ao tempo real e o sistema de revisões possibilita que apenas as alterações sejam recebidas economizando banda e se tornando mais eficiente em ambientes de conectividade escassa.

A figura 2 traz uma representação dos elementos envolvidos no processo de sincronismo do Sync Gateway que são os dispositivos móveis e a base de dados remota.

3.4.3 Couchbase Lite

Versão mobile do banco de dados, feita para rodar em dispositivos com hardwares mais modestos como smartphones, tablets e smartwatches.

Essa base de dados é criada e mantida pelo aplicativo SEA Point e armazena os dados coletados pelos operadores sincronizando os mesmos assim que houver conectividade ao servidor remoto.

---

[2] Disponível em: <https://www.couchbase.com/products/sync-gateway>. Acesso em 16 jul. 19

3.5 MySQL

Sistema de gerenciamento de bancos de dados (DBMS) relacional extremamente popular e com bastante tempo de mercado, utiliza a linguagem SQL como interface de manipulação de dados.

Usado mundialmente por empresas de grande porte como HP, Nokia, Sony, Cisco Systems, Google, entre outras sua popularidade se deve a várias características que a diferenciaram das outras soluções disponíveis dentre elas:

- *Open source* e gratuito, baseado na licença GPL, seu código fonte está aberto para qualquer pessoa explorar e sugerir melhorias.
- Multiplataforma, roda nas principais plataformas do mercado seja para servidores ou usuário domésticos.
- Desempenho e estabilidade, considerando seu tempo de mercado e maturidade.
- Compatibilidade, possui integração nativa com as principais linguagens de programação como PHP, Java, C#, Python...

3.6 ASP.NET Core

Plataforma *open source* e multiplataforma para desenvolvimento de aplicações web de alto desempenho desenvolvida pela Microsoft. O ASP.NET Core baseia-se na plataforma ASP.NET MVC que fazia parte do .NET Framework e pode ser considerado uma evolução dele, foi desenvolvido com o objetivo de tornar a plataforma mais moderna e compatível com o atual conceito de *cloud-computing*.

No projeto SEA Point o ASP.NET Core foi utilizado para desenvolver uma API REST para o *back-end* que serve de repositório de informações ao *front-end* e é responsável pelo controle de acesso baseado em permissões.

4 DESENVOLVIMENTO

4.1 Definição de escopo

O sistema é apresentado ao usuário final no formato de um painel web e um aplicativo mobile.

A portal web possui ferramentas de gerenciamento de projetos, nele o administrador pode criar, visualizar e atualizar projetos. Os projetos são atribuídos a operadores que utilizam o aplicativo mobile para coletar as informações relevantes ao projeto, além disso, na portal web também é possível exportar os projetos no formato adequado conforme exigido pelas concessionárias para verificação e aprovação dele.

O aplicativo mobile possibilita ao operador coletar informações como geolocalização, detalhes e foto dos pontos, aqui o operador visualiza todos os projetos que estão atribuídos ao mesmo e pode visualizar, adicionar e excluir pontos já coletados afins de concluir o projeto que depois será avaliado e submetido pelo administrador à concessionária.

A web API é utilizada tanto pelo painel web quanto pelo aplicativo mobile, seu papel é servir de origem de dados para o usuário autenticado que faz as requisições de acordo com seu nível de acesso e permissões de consulta e alterações. Funciona como uma abstração de toda a complexidade por trás de consultas a bancos de dados e conversões de formatos, utiliza o protocolo HTTP como meio de transporte das informações.

4.2 Elicitação de requisitos

Para elaboração do projeto de desenvolvimento do sistema foi feito um processo de obtenção dos requisitos para desenvolvimento da etapa inicial do projeto.

Esses requisitos foram obtidos através de reuniões com o diretor de operações da empresa onde foram apresentadas as funcionalidades prioritárias a serem implementadas no sistema.

4.2.1 Requisitos funcionais

Define as funções que um sistema deve possuir. A especificação de um requisito funcional descreve o que se espera que o sistema faça, isto é, uma saída específica com base em um conjunto de entradas e não define como essa atividade será implementada, apenas o resultado esperado, conforme descrito por (VIDAL, 2017). Com base nesse

conceito foram levantados os requisitos funcionais e prioridade de implementação para entrega da versão inicial do sistema a partir da observação das necessidades diárias do envolvidos nos projetos, conforme tabela 4.

Tabela 4:Requisitos funcionais

Número	**Requisito**	**Prioridade**
01	O administrador do sistema deve coletar as seguintes informações dos postes: Tipo, Altura, Esforço, Acessórios, Ocupantes, Geolocalização e uma foto.	Alta
02	O sistema deve permitir ao usuário cadastrar novos projetos, tipos de acessórios e ocupantes dos postes que foram mapeados.	Alta
03	O sistema deve permitir ao operador visualizar a lista de pontos e suas respectivas informações, como também excluir pontos anteriormente capturados.	Alta
04	O sistema deve sincronizar todas as informações capturadas com um servidor remoto quando houver conectividade com a internet ou, quando não houver, observar a disponibilidade da mesma para fazer o sincronismo.	Alta
05	O sistema deve possuir uma tela com informações do projeto e estatísticas	Alta
06	O sistema deve possuir uma tela com um mapa indicando os pontos já capturados no projeto.	Alta
07	O sistema deve permitir ao administrador cadastrar empresas, operadores e projetos.	Alta
08	O sistema deve permitir ao administrador atribuir operadores a projetos.	Alta
09	O sistema deve permitir ao administrador atribuir projetos a operadores, para que esses tenham acesso no aplicativo mobile e possam fazer as capturas.	Alta
10	O sistema deve permitir ao administrador visualizar as estatísticas do projeto, onde estará localizado: A quantidade, a lista e o mapa com todos os pontos coletados.	Alta
11	O sistema deve permitir ao administrador exportar os dados do projeto nos formatos: Excel (XLSX), Google Earth (KML) e AutoCAD (DWG).	Alta
12	O sistema deve exigir autenticação do operador no aplicativo mobile	Alta
13	O sistema deve utilizar poucos recursos do aparelho como bateria e memória, uma vez que o operador pode estar e local isolado durante a captura.	Baixa

4.2.2 Requisitos não funcionais

Durante o desenvolvimento de um sistema, é necessário que se leve em consideração a capacidade técnica, os requisitos que constituirão a interface externa, a confiabilidade, segurança, etc.. Dessa maneira, o sistema executará suas funções de acordo com suas finalidades, antes estabelecidas pela necessidade de cada usuário de software.

Com base nisso, foram levantados os requisitos funcionais e prioridade de implementação para entrega da versão inicial do sistema conforme tabela 5.

Tabela 5:Requisitos não funcionais

Número	Requisito	Prioridade
01	O sistema deve verificar a conectividade e informar o usuário através de notificações na barra de notificações do aplicativo.	Média
02	O sistema deve restringir a precisão mínima do GPS a 3 metros.	Alta
03	O sistema só poderá salvar o ponto coletado assim que todas as informações foram inseridas, inclusive a foto do poste.	Alta
04	O sistema deve alertar o operador e o projeto selecionado para agilizar a retomada de coleta de pontos.	Alta
05	O sistema deve apresentar uma interface limpa e intuitiva ao operador.	Média
06	O aplicativo mobile deve funcionar na plataforma Android	Alta
07	O painel web deve funcionar na plataforma web.	Alta

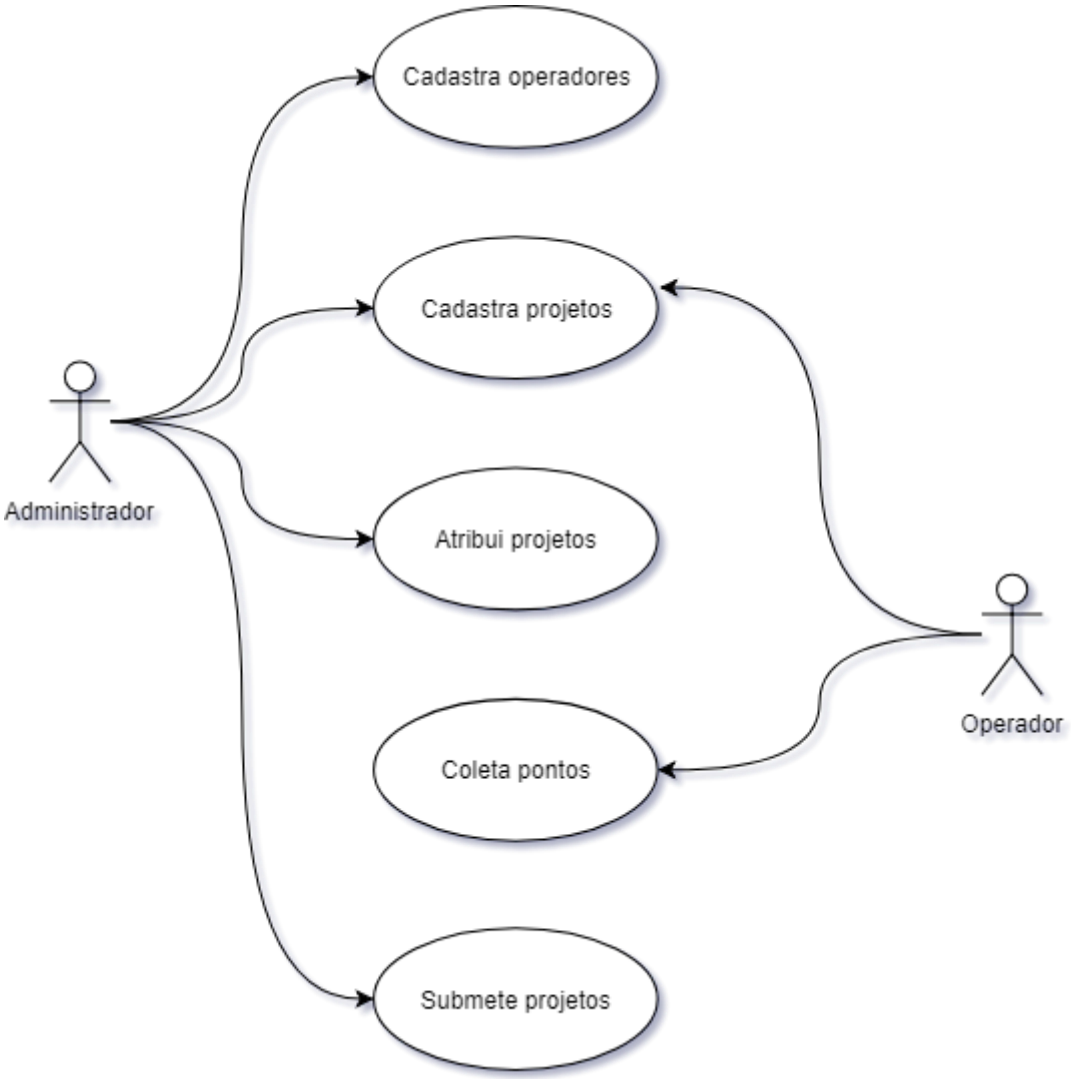
4.3 Diagramas de caso de uso

“O diagrama de casos de uso permite descrever os requisitos funcionais de um sistema de maneira simples e direta para usuários e desenvolvedores, pois ele deixa claro o que o sistema faz e quais são as funcionalidades que devem ser desenvolvidas” (MILETTO e BERTAGNOLLI, 2014).

Nesta seção são apresentados os diagramas que representam as funcionalidades do sistema além de todos os atores envolvidos na operação dele.

Na figura 3 é descrita a operação do sistema e cada tipo de usuário ligados a seus papéis. O administrador é responsável pelo cadastro, atribuição e acompanhamento do progresso de coleta das informações pelo operador que por sua vez utiliza o aplicativo mobile para coletar e visualizar os dados do projeto ao qual está atribuído.

Figura 3:Diagrama de caso de uso exibindo administrador e operador do sistema



Fonte: O autor

4.4 Metodologia

Todo o processo de desenvolvimento foi elaborado e executado segundo os princípios da metodologia ágil Scrum, onde uma etapa (Sprint) é quebrada em várias

tarefas menores (itens do backlog) a serem executadas para concluir esta etapa de maneira rápida e organizada e de maneira a agilizar a entrega e detecção de problemas.

4.5 Codificação

Cada etapa do desenvolvimento do projeto exigiu o uso de ferramentas e metodologias específicas.

Para o aplicativo mobile foi utilizado a IDE Android Studio tanto para prototipação das telas quanto para codificação, o Android Studio é ferramenta oficial de desenvolvimento para plataforma Android, utiliza principalmente a linguagem de programação Java com alguns outros elementos como JSON e XML.

4.5.1 Aplicativo mobile

O aplicativo foi desenvolvido utilizando predominantemente as linguagens Java, XML e JSON. Foi estruturado utilizando os conceitos de MVP (Model, View e Controller) e possui algumas dependências externas.

A ferramenta escolhida foi a IDE Android Studio, uma ferramenta popular para desenvolvimento de aplicativos para a plataforma Android, possui ferramentas que auxiliam na criação de classes, métodos e layouts de tela.

O projeto possui algumas dependências externas que facilitaram a entrega de algumas funcionalidades, como exemplo temos:

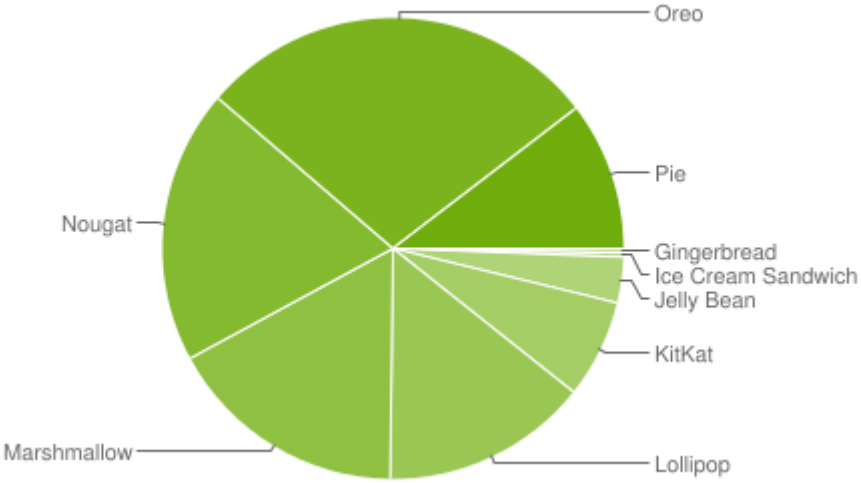
- **Retrofit**: Cliente REST para APIs baseadas em HTTP, tranforma o conjunto de endpoint em classes Java.
- Couchbase Lite: Banco de dados NoSQL com suporte a consultas e sincronismo com servidor remoto.
- **Osmdroid**: Ferramentas para inclusão de mapas OpenStreetMap em aplicações Android.
- **Crashlytics**: Ferramenta de telemetria de aplicações, coleta informações de uso e falhas durante o uso da aplicação.

Um dos pontos a se levar em consideração ao desenvolver aplicações para plataforma Android é a versão mínima do Sistema Operacional necessária para instalar e exeutar o aplicativo. Cada nova versão do sistema possui melhorias e novos recursos

porém, devido a baixa frequencia de atualizações liberadas pelos fabricantes nem sempre estão disponíveis.

Com base nos requisitos do sistema e porcentagem de disponibilidade do sistema na versão necessária, conforme exibido na figura 4, foi definido que a versão mínima exigida pelo sistema seria a 4.1 codenome Jelly Bean.

Figura 4: Distribuição da base de usuário conforme versão do Android



Fonte: Site Android Developers[3]

4.5.2 Painel web

O painel web foi desenvolvido utilizando a linguagem TypeScript (com sintaxe JSX) e CSS. É baseado principalmente no framework ReactJS para aplicações web possuindo algumas poucas outras dependências, como por exemplo:

- **Ant Design**: É uma linguagem de design que define diretrizes e recursos para um melhor UX com foco em aplicações corporativas.
- **TypeScript**: É um superconjunto de JavaScript desenvolvido pela Microsoft que adiciona tipagem e alguns outros recursos a linguagem.
- **Leaflet**: Biblioteca para inclusão de mapas interativos em aplicações web baseadas em JavaScript.

4.5.3 Web API

Para o painel web, tanto o *front-end* em ReactJS quanto o *back-end* em ASP.NET Core, foi utilizado o editor de texto Visual Studio Code.

[3] Disponível em: <https://developer.android.com/about/dashboards?hl=pt-br>. Acesso em 16 jul. 19

É baseada nos componentes de WebAPI do ASP.NET Core possui algumas dependências externas para entender suas funcionalidades, como por exemplo:

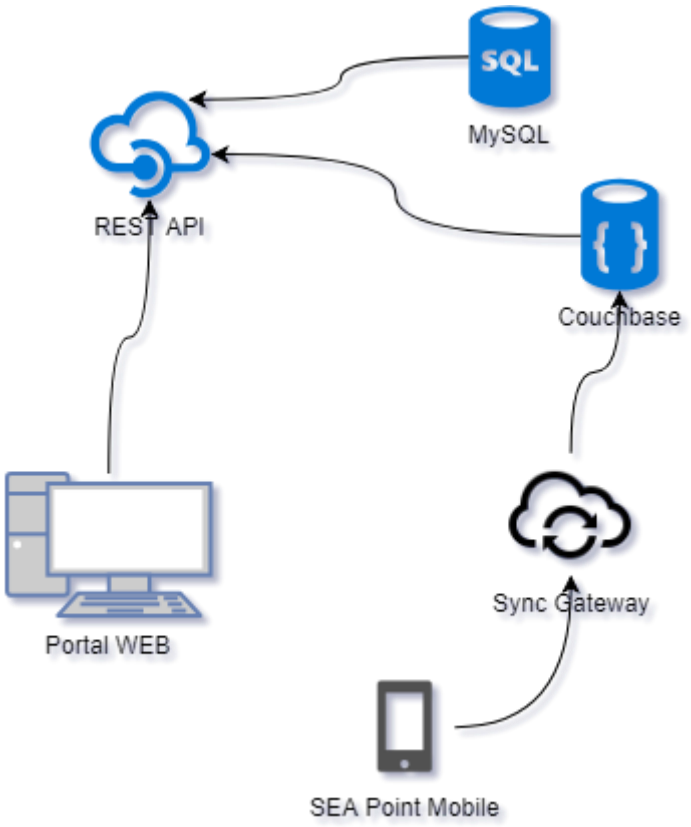
- **ASP.NET Identity**: É um conjunto de extensões para o ASP.NET Core que adiciona funcionalidades de autenticação e autorização com vários algoritmos de criptografia de senhas e modelos de permissões.
- **Couchbase**: Versão C# da biblioteca oficial do banco de dados, utilizada para fazer consultas e cadastros na base de dados.
- **Swagger**: Ferramenta de geração de documentação automática para APIs em diversas plataformas, possui também um painel interativo que permite fazer testes em tempo real com o *back-end*.

5 RESULTADOS OBTIDOS

A partir do projeto descrito nesse trabalho foi possível entregar uma versão mínima viável do sistema para avaliação da empresa.

Em resumo, o sistema e seus principais componentes são relacionados conforme figura 5, os demais recursos serão descritos detalhadamente nas seções subsequentes dessa seção.

Figura 5: Visão geral do sistema



Fonte: O autor

5.1 Aplicativo mobile

Nessa seção serão demonstradas as funcionalidades do aplicativo desenvolvido como resultado deste trabalho.

A seguir são apresentadas imagens dos principais fluxos de utilização do aplicativo:

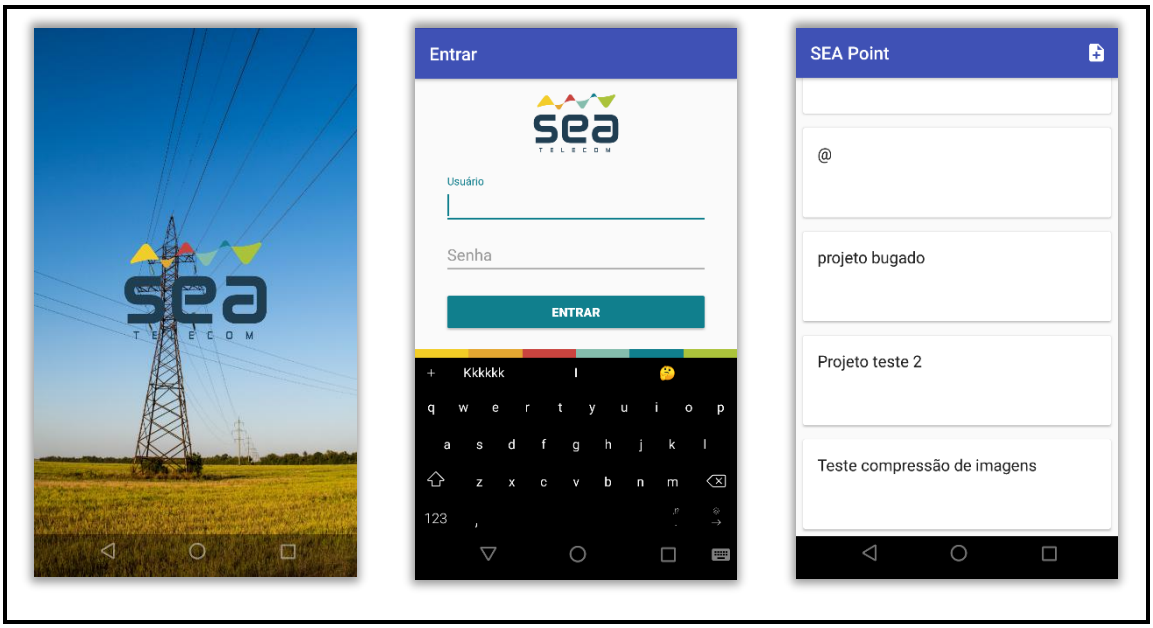
5.1.1 Login

O processo de Login é composto pela tela de *splash screen* que é apresentada ao abrir o aplicativo seguida pela tela onde o operador deve informar seu usuário e senha e,

após autenticação, a tela onde o operador deve selecionar o projeto no qual se deseja trabalhar.

As telas apresentadas ao operador durante o processo de login são exibidas na figura 6.

Figura 6: Processo de Login



Fonte: O autor

5.1.2 Coleta de ponto

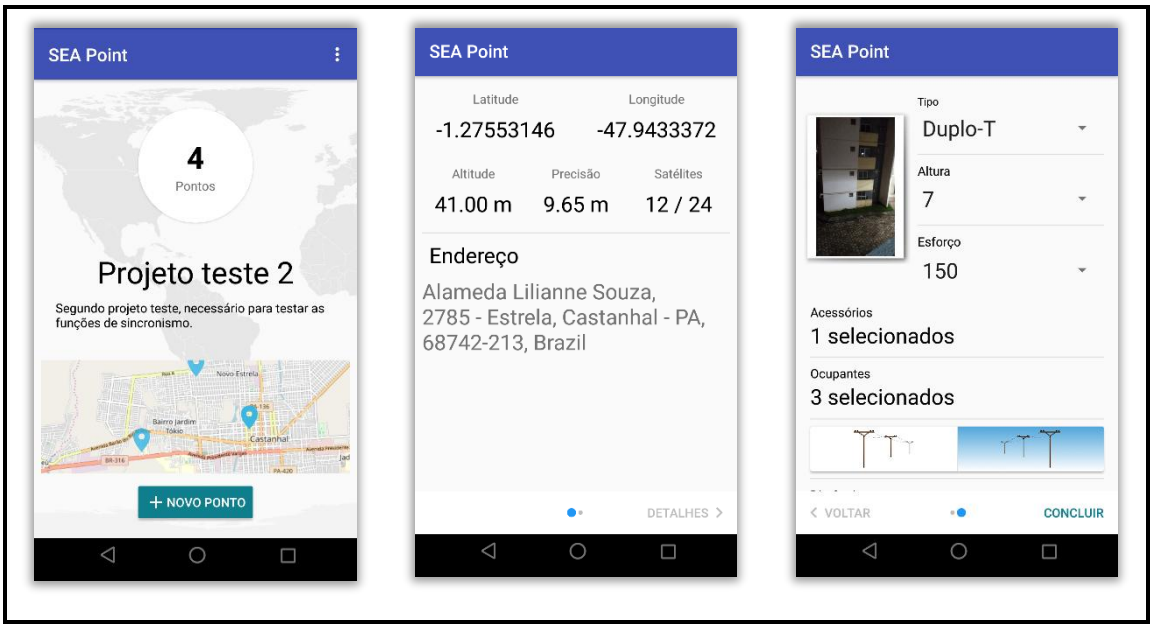
Na tela de resumo do projeto o operador deve clicar no botão de novo ponto para iniciar a captura das informações, o primeiro passo é a captura de geolocalização, nela o operador aguarda obter a melhor precisão possível para as coordenadas geográficas e então prosseguir para a tela de captura de foto.

Após capturar uma foto do poste e confirmar o uso da mesma o usuário é apresentado a tela de detalhes do poste.

Na tela de detalhes o usuário insere as características do poste conforme são exibidas no formulário, assim que todas as informações forem preenchidas o usuário clica no botão concluir no canto inferior direito para confirmar os dados coletados.

As telas apresentadas ao operador para coleta dos dados dos pontos são apresentadas na figura 7.

Figura 7: Processo de captura de ponto



Fonte: O autor

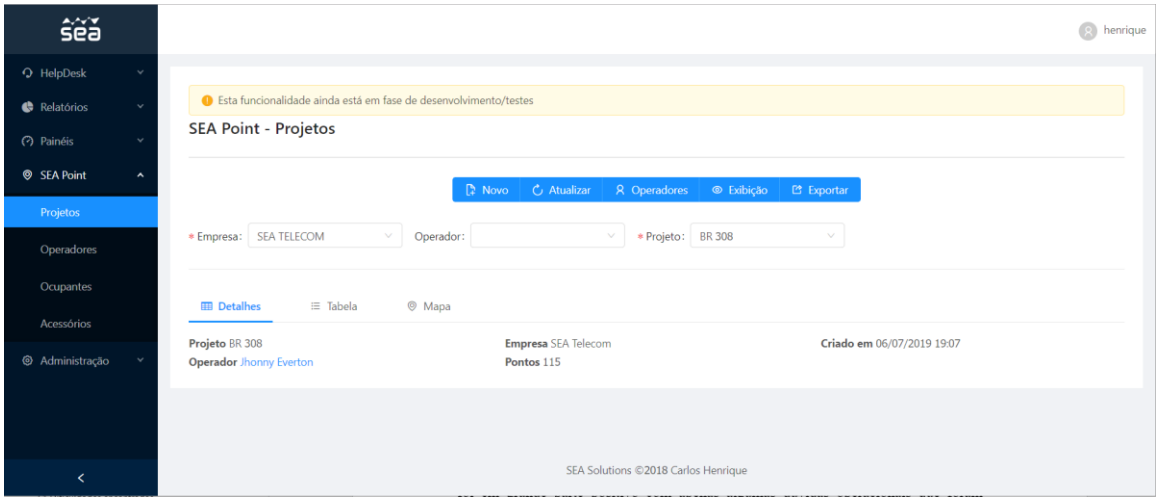
5.2 Web (*front-end*)

No painel web é possível administrar os projetos, isto é, criar, atribuir alterar e excluir projetos.

O front-end é uma aplicação do tipo SPA desenvolvida em JavaScript utilizando a biblioteca ReactJS, funciona em conjunto com o *back-end* através de comunicação com a API através de REST.

Na figura 8 é apresentado a tela de resumo que é um das telas disponíveis ao administrador para acompanhamento do projeto.

Figura 8: Tela do projeto no painel web



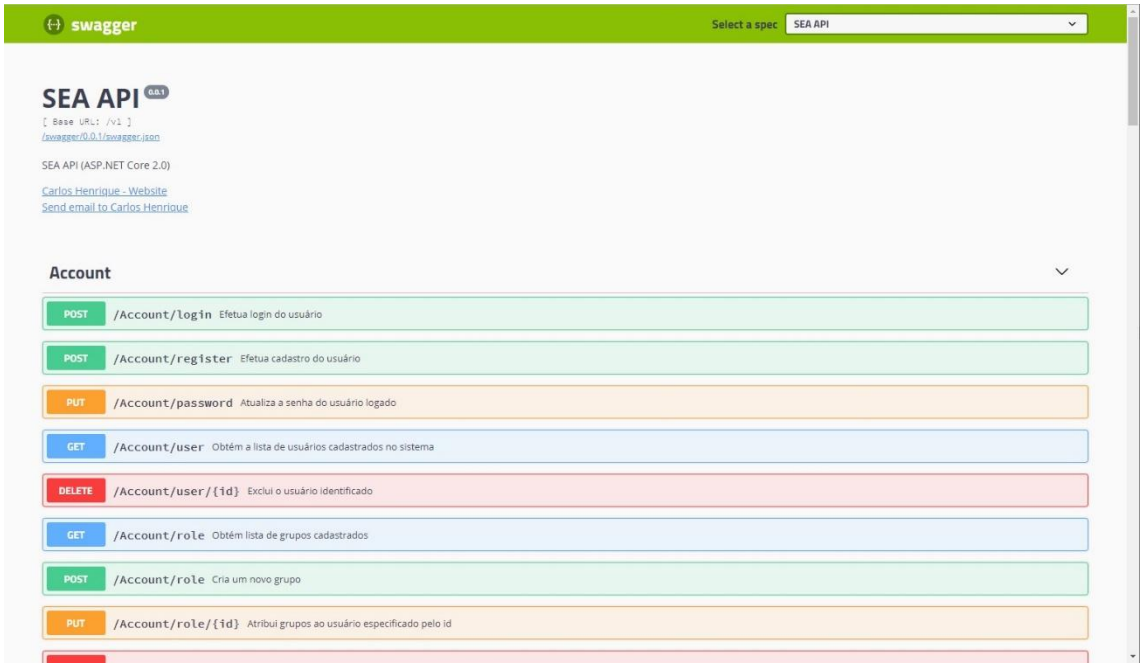
Fonte: O autor

5.3 Web (*back-end*)

A API RESTFul desenvolvida para o *back-end* da aplicação serve tanto para o painel web quanto para aplicação mobile, é utilizada na leitura dos dados do projeto e autenticação dos operadores e administradores.

A API possui mais de 40 *endpoints* que permitem vários tipos de operações sobre os dados conforme exibido na figura 9.

Figura 9: REST endpoints do back-end web



Fonte: O autor

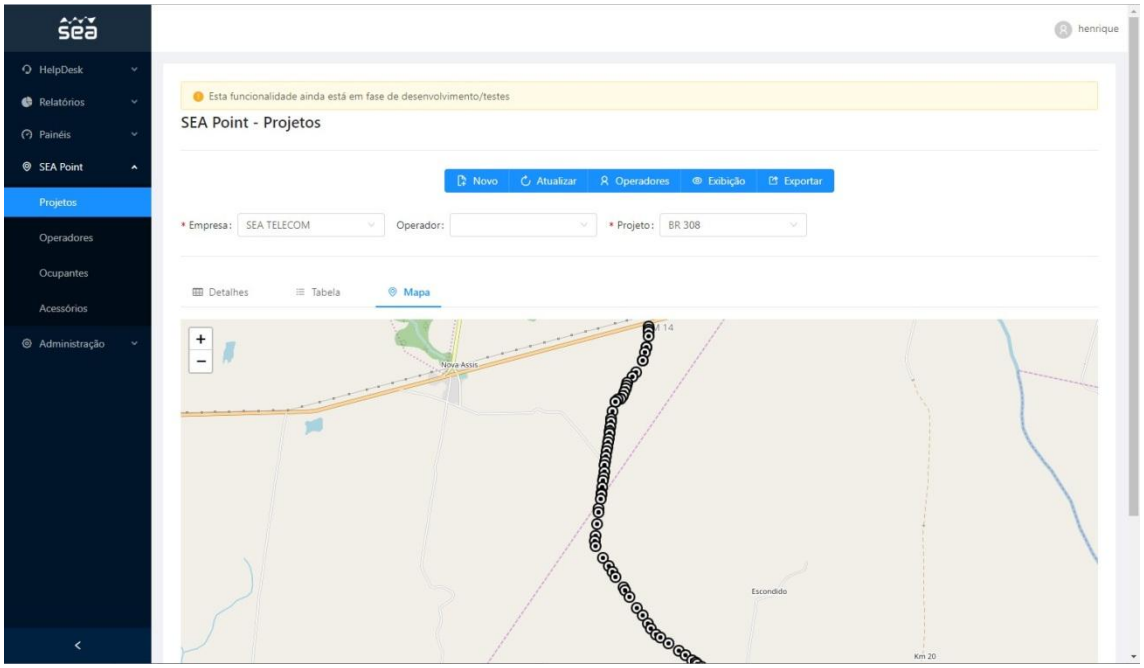
6 AVALIAÇÃO DE RESULTADOS

Uma vez completa a versão inicial do sistema foram feitos testes internos para homologação das funcionalidades e posterior uso em um ambiente real.

Foram executados alguns projetos em conjunto com provedores parceiros que estavam trabalhando em projetos relacionados, sempre com o acompanhamento de perto do desenvolvedor para responder as dúvidas e obter feedback dos usuários.

Na figura 10 é apresentado a tela de mapa que é um das telas disponíveis ao administrador para acompanhamento do projeto.

Figura 10: Pontos geográficos coletados com o SEA Point



Fonte: O autor

6.1 Teste e feedback

Os testes consistiram no levantamento de pontos para projetos reais, o *feedback* foi em grande parte positivo com apenas algumas dúvidas operacionais que foram rapidamente resolvidas.

Não foi detectada incompatibilidade com nenhum dispositivo móvel específico, porém houve casos em que a precisão do GPS e a qualidade da foto do poste foram menos precisas em alguns fabricantes.

6.2 Perfis de usuários

Os usuários do sistema eram em sua maioria funcionários operacionais de campo e coordenadores de projetos que estão habituados ao processo manual de levantamento de pontos então, após um período de adaptação, tiveram uma boa recepção ao sistema.

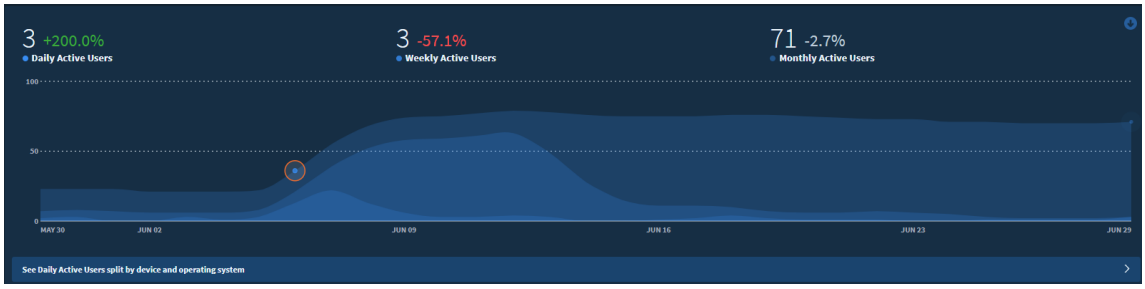
As facilidades que o sistema propõe chamaram a atenção das instituições envolvidas no processo de elaboração de projetos de compartilhamento de infraestrutura, portanto não foi difícil encontrar base de usuário para testes.

As organizações onde os profissionais exercem suas atividades são parceiros em projetos e trabalham sob o regime de acordo de confidencialidade quanto ao uso do sistema, as empresas são principalmente: TOPPNET, ZUM Telecom, WIFI Tecnologia e Informática, RPNet Soluções em Internet e Telefônica Brasil.

6.3 Resultados de Usabilidade

De maneira geral, os usuários se habituaram ao uso diário do aplicativo, com a última leitura sendo de 71 usuários ativos mensalmente, com aproximadamente 100% de estabilidade do sistema conforme observado na figura 11.

Figura 11: Estatísticas de usuários ativos no sistema

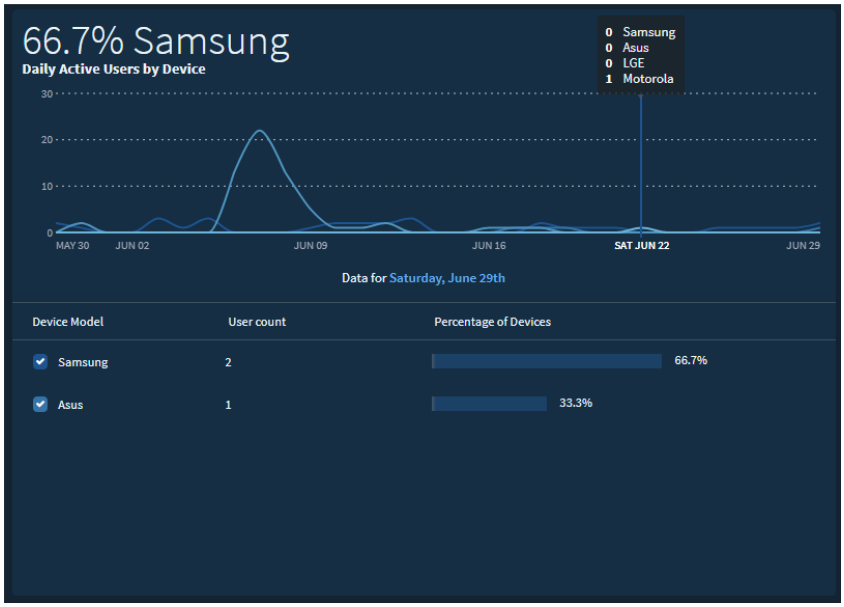


Fonte: O autor

A distribuição de usuário por plataforma mostrou que a maioria da base de usuários utilizava aparelhos da fabricante Samsung que é líder mundial de vendas de smartphones de acordo com o IDC [4], porém também registramos smartphones da marca Asus que além de ser uma marca pouco conhecida, utiliza processadores de arquitetura x86 os quais tivemos a oportunidade de testar a estabilidade do sistema, a distribuição dos usuários por fabricante de dispositivo móvel pode ser observada na figura 12.

[4] (INTERNATIONAL DATA CORPORATION (IDC), 2019)

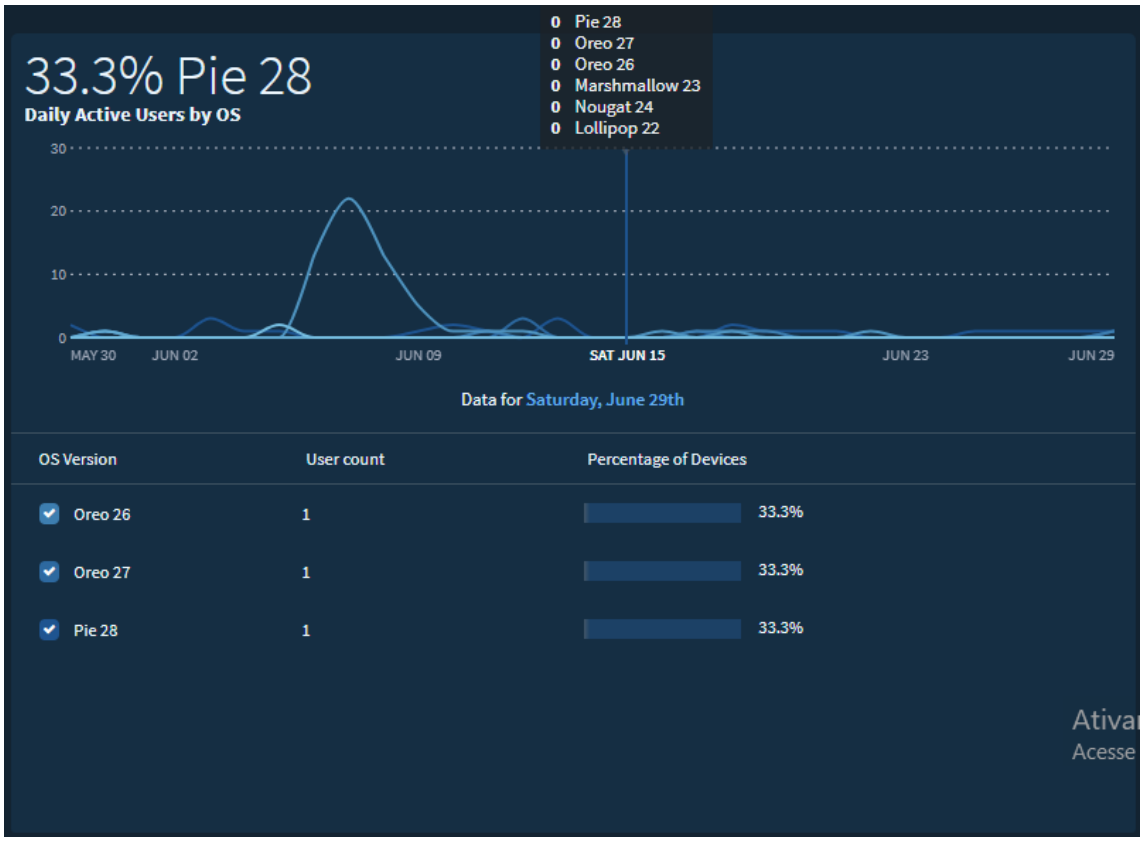
Figura 12: Distribuição da base de usuário por fabricante



Fonte: O autor

Outra informação importante coletada durante os testes foi a distribuição da base de usuário por versão do sistema operacional utilizado, uma vez que cada versão do Android apresenta mudanças nos recursos disponíveis aos desenvolvedores o que gera uma grande causa de problemas de compatibilidade. A distribuição de usuário por versão do sistema pode ser observada na figura 13 que traz um gráfico temporal de linhas com a quantidade de usuários em cada versão indicada.

Figura 13: Distribuição da base de usuários por versão do Android



Fonte: O autor

Com base nos dados coletados, conclui-se que os resultados observados atingem os objetivos de funcionalidade e usabilidade propostas durante o planejamento do projeto.

7 CONSIDERAÇÕES FINAIS

Nesse trabalho foi desenvolvido um conjunto de sistemas que trabalham de maneira combinada para automatizar todo o processo de elaboração de projetos de compartilhamento de infraestrutura, uma atividade muito comum em provedores de acesso à internet, essa atividade ainda está passando por muitas adequações, novas leis e regras, por isso é um processo bastante manual.

A execução dessa atividade é, muitas vezes, um processo extremamente manual, lento e impreciso. Com o uso do sistema desenvolvido nesse trabalho, todo o processo é unificado entre as obrigações do operador que captura as informações, o administrador que elabora e envia o projeto e a concessionária que analisa e faz a aprovação.

A partir do trabalho produzido foi possível, em conjunto com outras empresas da região, automatizar cerca de 40 projetos e coletar mais de 3000 pontos, isso gerou uma economia significativa no tempo e mão de obra necessários para coleta, tratamento e disponibilização dos dados. Tivemos feedback positivo de toda a base de usuário e sugestões de melhorias que serão implementadas nas próximas versões do sistema.

7.1 Dificuldades encontradas

A natureza multidisciplinar do sistema, uma vez que se está trabalhando em três plataformas distintas e utilizando dois bancos de dados com modelos de indexação e consultas diferentes, tornou o processo de pesquisa e planejamento extremamente complexo.

Além disso, apenas um desenvolvedor esteve disponível para fazer a arquitetura, gerencia de projeto, codificação, design e teste o que causou certa lentidão no processo de detecção de problemas, correções e criação de correções e melhorias.

Uma dificuldade bastante encontrada pelos operadores é o sincronismo das imagens dos postes capturadas, uma vez que a concessionária de energia detentora da infraestrutura exige que essas imagens tenham alta resolução e boa qualidade, a má qualidade de conectividade de conexão de algumas regiões causou lentidão no sincronismo dos projetos.

7.2 Trabalhos futuros

Existem muitas melhorias e correções já planejadas para o SEA Point, que envolvem a formação de uma equipe de desenvolvedores especializadas, uma vez que o sistema no estado atual foi desenvolvido apenas por uma pessoa.

Além disso, existe interesse por parte de operadoras e concessionárias, portanto há viabilidade comercial para o produto, isso gera um novo desafio na próxima fase de planejamento que trata das formas de monetização das ferramentas e segurança da informação.

REFERÊNCIAS

JSON, 2019. Disponivel em: <https://www.json.org/json-pt.html>. Acesso em: 06 Julho 2019.

AGÊNCIA ESTADO. **Anatel e Aneel decidem rever regras de compartilhamento de postes**, 2018. Disponivel em: <https://www.correiobraziliense.com.br/app/noticia/economia/2018/09/26/internas_economia,708377/anatel-e-aneel-decidem-rever-regras-de-compartilhamento-de-postes.shtml>. Acesso em: 27 Junho 2019.

ANEEL; ANATEL; ANP. **Resolução Conjunta nº 1**, 24 Novembro 1999. Disponivel em: <http://www.anatel.gov.br/legislacao/resolucoes/resolucoes-conjuntas/84-resolucao-conjunta-1>.

FIELDING, ET AL. **Hypertext Transfer Protocol -- HTTP/1.1**, 1999. Disponivel em: <https://tools.ietf.org/html/rfc2616>. Acesso em: 06 Julho 2019.

INTERNATIONAL DATA CORPORATION (IDC). **IDC**, 2019. Disponivel em: <https://www.idc.com/getdoc.jsp?containerId=prUS45042319>. Acesso em: 29 Junho 2019.

MILETTO, E. M.; BERTAGNOLLI, S. D. C. **Desenvolvimento de Software II:** Introdução ao Desenvolvimento Web com HTML, CSS, JavaScript e PHP - Eixo: Informação e Comunicação - Série Tekne. [S.l.]: Bookman Editora, 2014.

SCHWABER, K.; SUTHERLAND, J. **The Scrum Guide**, Novembro 2017. Disponivel em: <https://scrumguides.org/scrum-guide.html>.

VIDAL, A. **Agile Think Canvas**. [S.l.]: Brasport, 2017.

WORLD WIDE WEB CONSORTIUM. **Relationship to the World Wide Web and REST Architectures**, 2004. Disponivel em: <https://www.w3.org/TR/2004/NOTE-ws-arch-20040211/#relwwwrest>. Acesso em: 06 Julho 2019.