



UNIP - UNIVERSIDADE PAULISTA

ADRIANNE SOUSA OLIVEIRA - D7048D1

ALAN DA SILVA QUEIROZ - N228GJ6

ANA CAROLINA MARTINS SIMAL - N2734E2

ATIVIDADE PRÁTICA SUPERVISIONADA

**Desenvolvimento de uma aplicação de sistema distribuído
para dispositivo móvel**

SÃO PAULO

2021

UNIP - UNIVERSIDADE PAULISTA

ADRIANNE SOUSA OLIVEIRA - D7048D1

ALAN DA SILVA QUEIROZ - N228GJ6

ANA CAROLINA MARTINS SIMAL - N2734E2

ATIVIDADE PRÁTICA SUPERVISIONADA

**Desenvolvimento de uma aplicação de sistema distribuído
para dispositivo móvel**

Projeto de desenvolvimento de um aplicativo mobile a fim de colocar em prática os conteúdos da disciplina Qualidade de Software apresentado ao curso de Ciência da Computação da UNIP (Universidade Paulista) para a implantação e prática dos conteúdos vistos no semestre.

SÃO PAULO

2021

“Se tu o desejas, podes voar, só tens de confiar muito em ti.”

Steve Jobs

Sumário

RESUMO.....	6
ABSTRACT	7
Lista de Ilustrações.....	8
Lista de tabelas	9
1 OBJETIVO.....	10
2 INTRODUÇÃO	11
3 CONTEXTUALIZAÇÃO	12
3.1 Reciclagem e Reutilização de Lixo eletrônico.....	12
4 FUNDAMENTOS DAS TECNOLOGIAS.....	18
4.2 Sistemas Distribuídos	18
4.3 Microserviços	19
4.4 Web Services.....	19
4.5 Container	20
4.6 Firebase	22
4.7 Google Cloud	22
4.8 Normas ISO.....	24
• SEI / CMM.....	24
4.9 DDD	25
4.10 Segurança do Software.....	27
4.10.1 Segurança em Sistemas Distribuídos	27
5 DESENVOLVIMENTO DO PROTÓTIPO	29
5.1 Tecnologias.....	29

5.2 Código Fonte	30
5.3 Protótipo.....	47
6 CONCLUSÃO	58
REFERÊNCIAS	59
7 FICHAS	62
7.1 Ficha Adrienne.....	62
7.2 Ficha Alan Queiroz	63
7.3 Ficha Ana Simal.....	64

RESUMO

O objetivo deste documento é apresentar uma aplicação mobile desenvolvida para auxiliar os usuários com o descarte correto de seus resíduos eletrônicos. Por se tratar de uma aplicação Web consumindo recursos de APIs, o usuário é capaz de usá-la em seus aparelhos androids, assim conseguindo contactar as empresas que fazem o descarte de lixo eletrônico através de uma busca pelo mapa de sua região sem a necessidade de aplicações de terceiros. A ideia principal é facilitar o encontro entre aqueles que desejam fazer a ação ecológica correta com as empresas especializadas no assunto, através de um software com interface amigável.

Palavras-chave: Aplicação, Mobile, Resíduos Eletrônicos, Lixo Eletrônico, Empresas.

ABSTRACT

The purpose of this document is to present a mobile application developed to help users with the correct disposal of their electronic waste. As it is a web application consuming API resources, the user is able to use it on their android devices, thus being able to contact companies that dispose of electronic waste through a map search of their region without the need to third-party applications. The main idea is to facilitate the meeting between those who wish to carry out the correct ecological action with companies specialized in the subject, through software with a user-friendly interface.

Keywords: Application, Mobile, Electronic Waste, Electronic Junk, Business.

Lista de Ilustrações

Figura 1	15
Figura 2	16
Figura 3	18
Figura 4	21
Figura 5	22
Figura 6	23
Figura 7	24
Figura 8	25
Figura 9	26
Figura 10	44
Figura 11	45
Figura 12	46
Figura 13	47
Figura 14	48
Figura 15	49
Figura 16	50
Figura 17	51
Figura 18	52
Figura 19	53
Figura 20	54
Figura 21	55
Figura 22	56
Figura 23	57

Lista de tabelas

Tabela 1	45
Tabela 2	45
Tabela 3	46

1 OBJETIVO

O projeto apresentado tem como objetivo solucionar o problema do descarte correto de resíduos eletrônicos por meio de pessoas comuns com o intermedio de empresas especializadas no assunto. Para tal, foi desenvolvido uma aplicação mobile que fará através de dados dessas empresas especializadas em descarte o encontro com o usuário, ao realizar uma busca interna em seu banco de dados, representado visualmente através de um mapa a empresa mais próxima do mesmo.

2 INTRODUÇÃO

Nós encontramos atualmente em um planeta com uma enorme leva de aparelhos eletrônicos formados por diversos componentes que, aos leigos, não são ao menos conhecidos. Todavia, até o menor destes componentes pode prejudicar o meio ambiente e as pessoas que vivem à sua volta.

É notório que estamos rodeados de aparelhos que, após sua vida útil, são descartados em qualquer lugar por falta de informação. Mas, o descarte incorreto dos mesmos pode provocar a contaminação dos lençóis freáticos e, eventualmente, à saúde da população que vive nas proximidades.

Iniciativas que ajudam a diminuir os danos causados pelo descarte incorreto destes aparelhos e/ou transmitir informações sobre a maneira correta de se armazenar e eliminar estas peças de forma clara para a população são de vital importância para boa parte da recuperação do meio ambiente.

O presente trabalho foi realizado tendo dois objetivos principais. O primeiro é dissertar sobre a reciclagem, reutilização e descarte de componentes eletrônicos. O segundo é desenvolver um site com base nestas dissertações, apresentando meios fáceis e acessíveis para que as pessoas possam contribuir com uma sociedade mais sustentável.

3 CONTEXTUALIZAÇÃO

3.1 Reciclagem e Reutilização de Lixo eletrônico

- **Lixo Eletrônico**

O conceito “lixo eletrônico” é abrangente, engloba questões que vão desde o denominado lixo informacional que circula na *Web* até o problema do descarte dos aparelhos e componentes eletrônicos. No Brasil, ainda não há um estudo especializado que nos permita entender, calcular e registrar o consumo de aparelhos eletroeletrônicos, para, possivelmente, poder prever um número futuro para o seu descarte, assim como o seu destino.

Segundo a UMICORE, lixo eletrônico “é aquele formado por aparelhos eletrônicos como baterias recarregáveis, celulares e placas de circuito impresso (placa mãe, de rede, de modem e de vídeo) que chegaram ao final de sua vida útil” ou que foram descartadas por se tornarem tecnologicamente obsoletas. Pilhas, lâmpadas, eletrodomésticos, freezer, geladeiras, televisão, micro-ondas e outros também fazem parte deste conceito caracterizando-se por conter substâncias potencialmente tóxicas, tais como chumbo, mercúrio, cádmio e etc., assim como materiais de valor econômico, entre os quais estão o cobre, ouro, prata e alumínio.

Habitamos em uma sociedade globalizada, conectada e interligada através de variados instrumentos como o computador, o celular, a televisão e etc., efeito da globalização que configurou o espaço geográfico internacional em redes, não somente de comunicação. O lixo eletrônico é um lixo de origem tecnológica e se apresenta como um problema especialmente a partir dos anos de 1980, nas novas descobertas científicas e na ampliação dos padrões de consumo com a oferta de novos produtos baseados na tecnologia da microeletrônica. Consequentemente temos a necessidade de descartá-los de maneira correta e segura, tanto os aparelhos inteiros como somente seus componentes.

Dados apontam que nos últimos 30 anos, o consumo de recursos naturais para fabricação de equipamentos eletrônicos tem aumentado preocupantemente, cerca de um

terço dos mesmos disponíveis no planeta foram transformados também em energia, produtos de consumo e semelhantes.

A cultura de produção e consumo em massa é uma das causas responsáveis pela grande quantidade de lixo eletrônico a ser descartado no mundo. Este meio de produção permite que os produtos fabricados sejam comercializados a preços mais acessíveis a grande maioria da população, desencadeando num maior número de consumidores do comércio eletrônico e futuramente muitos produtos a serem descartados, reciclados e até mesmo reutilizados.

O relatório da ONU publicado em 22 de fevereiro de 2010 para a imprensa mundial “Reciclando - Do lixo eletrônico a recursos, divulgado pelo Programa das Nações Unidas para o Meio Ambiente (PNUMA)” mostra o tamanho do problema no mundo. O relatório informa que: Atualmente, o mundo gera 40 milhões de toneladas de lixo eletrônico por ano. São pilhas e pilhas de velhos e abandonados computadores, impressoras, telefones celulares, pagers, câmeras fotográficas digitais, tocadores de música digitais, geladeiras, televisores e brinquedos. A China, com 2,3 milhões de toneladas anuais, já é a vice-campeã mundial em quinquilharias descartadas, perdendo apenas para os Estados Unidos (3 milhões de toneladas anuais). Até 2020, esse número deve crescer entre 200% e 400%. Na Índia, o salto dever ficar em torno de 500%.

Entre os 11 países avaliados pelo estudo da ONU, a posição do Brasil não é nada confortável com relação a este problema, e, segundo o relatório, ocupando o “primeiro lugar no ranking de descarte de computadores per capita, que chega a pouco mais de 0,5kg por pessoa ao ano – como comparação, o índice na China é de 0,2kg por pessoa ao ano”.

De acordo com a Agência Brasileira de Desenvolvimento Industrial (ABDI), os equipamentos eletroeletrônicos compreendem quatro linhas de produtos:

- A) Linha branca: refrigeradores, fogões, secadoras e lavadoras.
- B) Linha marrom: monitores, TVs, equipamentos de áudio e filmadoras.
- C) Linha azul: batedeiras, liquidificadores, furadeiras e cafeteiras.
- D) Linha verde: computadores, notebooks, tablets e celulares.

Tendo em mente o mundo em que vivemos e um futuro previamente certo, o mundo precisa de atitudes colaborativas que indiquem para a população como cuidar do

local onde vive, começando pelas pequenas peças e lixos que guarda em casa, que, com um descarte correto, aquele objeto pode ser reutilizado, reciclado ou, se não houver solução descartado corretamente, evitando assim, possíveis doenças, contaminações e poluições.

- **Reciclagem e Descarte**

No Brasil, mais de 1,4 milhão de toneladas de lixo eletrônico são descartados por ano. E, de todo esse lixo, apenas 2% é reciclado, causando, como já dito, um acúmulo enorme de lixo e a contaminação do solo e da água que, conseqüentemente chega a nossa casa nos contaminando. Uma pesquisa da CETESB (Companhia Ambiental do Estado de São Paulo) revelou que só na cidade de São Paulo, há mais de 5 mil áreas contaminadas pelo descarte incorreto de componentes eletrônicos.

O setor de reciclagem brasileiro vem ao longo dos anos se tornando mais necessário e indispensável, levando cada vez mais a obrigatoriedade de descarte correto dos materiais eletrônicos. Em 2010, a Lei nº 12.305 foi sancionada e regulamentada pelo decreto nº 7.404/10 obrigando indústrias de equipamentos eletrônicos a serem responsáveis por dar a destinação correta aos aparelhos após o período de uso. Entretanto, esta lei ainda não está funcionando como deveria, visto que a maioria das empresas de comércio de produtos eletrônicos não possui nenhum posto de coleta.

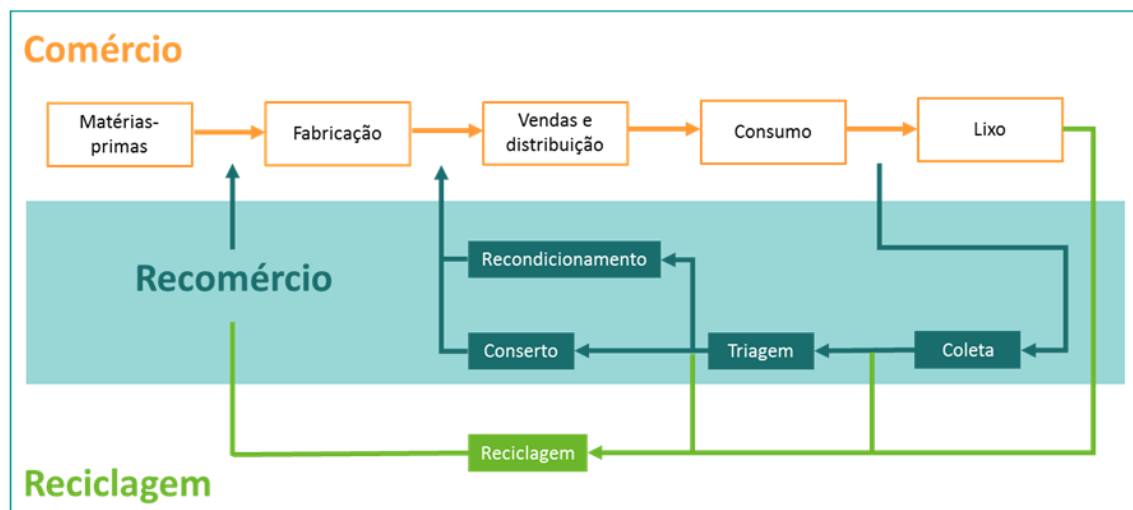
O maior obstáculo para que a reciclagem do e-lixo no Brasil seja mais eficaz é o custo, principalmente para desmontar cada aparelho, separar as partes e etc. contando também com toda a logística que seria envolvida para que esse processo possa acontecer.

Em média, um produto eletrônico ou eletroeletrônico conta com cerca de 40 elementos diferentes, todos misturados dentro de uma única peça. Como por exemplo, as placas de circuito impresso (PCIs) que, na sua composição contam com elementos como chumbo, cobre, níquel, ouro, prata, platina, entre outros. E com a presença dos mesmos, torna a reciclagem deste componente mais cara e menos rentável, tanto que no Brasil, ainda não é feito esse processo. Quando recicladas, estas placas são enviadas para o exterior, aumentando ainda mais os custos.

Portanto, o lixo eletrônico, em sua maior parte, não é tão lucrativa para as empresas de ser reciclado e sim prejudicial as finanças da empresa, para isso, há centros de especialização em descarte eletrônicos. O material restante para reutilização e reaproveitamento é levado para centros de reciclagem especializados para o correto manuseio destes materiais.

Contamos com outras soluções igualmente boas para reduzir os impactos ambientais, além da reciclagem como o “*recommerce*”, que é o comércio inverso ou sistema de recuperação que consiste na troca de um produto usado entre duas partes. O conceito vem da junção entre as palavras “Reusar” e “Comércio”.

Figura 1



(fonte: <http://www.recomercio.com.br/o-que-e-recommerce/>)

Não sendo destinado apenas para pessoas físicas, integrar o *recommerce* ao funcionamento corporativo de uma empresa permite que a mesma possa viabilizar práticas de respeito e preservação ambiental.

Em geral, o *recommerce* permite revender os aparelhos eletrônicos usados, utilizando os mesmos princípios da compra, como por exemplo, um preço de recompra garantido, uma logística reversa, um produto de qualidade boa e em boas condições de uso. Na internet, há diversos sites que proporcionam as pessoas o “Recomércio”.

A expectativa do uso do *recommerce* é um incentivo de descarte e reciclagem,

pois facilita tais processos, além de facilitar a revenda desses materiais. Esse processo também dinamiza a reciclagem/descarte, permitindo que mais e-lixo seja cuidado em um menor espaço de tempo.

Figura 2



(fonte: <http://www.recomercio.com.br/o-que-e-recommerce/>)

- **Substâncias Nocivas**

Com o incentivo ao desenvolvimento e consumo de eletrônicos é gerado o acúmulo de lixo eletrônico. E em seus componentes como já citato diversas vezes antes, encontra-se muitos componentes e substâncias, nocivas ou não. Dentre estas substâncias as mais perigosas são: O mercúrio, Chumbo, Cádmio e o Berílio.

Além de ser prejudicial ao meio em que se encontra, essas substâncias têm um grande efeito negativo no corpo humano, alguns desses efeitos são:

No Mercúrio, o metal invade e deteriora o sistema nervoso e pode causar perturbações motoras e sensitivas, além de tremores e demência, intoxicação aguda: Aspecto cinza escuro na boca e faringe, dor intensa, vômitos, sangramento nas gengivas, sabor amargo na boca, ardência no aparelho digestivo, diarreia grave ou sanguinolenta, inflamação na boca queda ou afrouxamento dos dentes, glossite, tumefação da mucosa grave, necrose nos rins, problemas hepáticos graves. Intoxicação

crônica: Transtornos digestivos e nervosos, caquexia, estomatite, salivação, mau hálito, anemia, hipertensão, afrouxamento dos dentes, problemas no sistema nervoso central, transtornos renais leves, possibilidades de alteração cromossômica.

O Chumbo, ele pode causar alterações genéticas, atacar o sistema nervoso, a medula óssea e os rins, além de causar câncer, intoxicação aguda: fraqueza, irritabilidade, astenia, náusea, dor abdominal com constipação e anemia.

Intoxicação crônica: perda de apetite, perda de peso, apatia, irritabilidade, anemia, danos nos sistemas nervoso, respiratório, digestivo, sanguíneo e aos ossos.

O Cádmio, pode causar dores abdominais, perda de olfato, câncer de pulmão e de próstata, anemia e osteoporose, intoxicação aguda: dores abdominais, náuseas, vômitos, diarreias. Intoxicação crônica: perda de olfato, tosse, falta de ar, perda de peso, irritabilidade, debilitação dos ossos, danos aos sistemas nervoso, respiratório, digestivo, sanguíneo e aos ossos, é cancerígeno para pulmões e rins.

O Berilo, este pode causar calafrios, tosse, acúmulo de fluidos no pulmão e câncer de pulmão, intoxicação aguda: calafrios, febre, tosse dolorosa e acúmulo de fluidos nos pulmões, podendo levar à morte.

Intoxicação crônica: Beriliose ou granulomatose pulmonar crônica, lesões pulmonares, é cancerígeno no pulmão.

4 FUNDAMENTOS DAS TECNOLOGIAS

4.2 Sistemas Distribuídos

Para que haja um sistema distribuído há a necessidade de compartilhamento de recursos como por exemplo de diferentes servidores, com diversos componentes espalhados ou distribuídos, consumidos por clientes. Dentre as consequências para tal definição, há a inexistência de relógio global e falhas independentes.

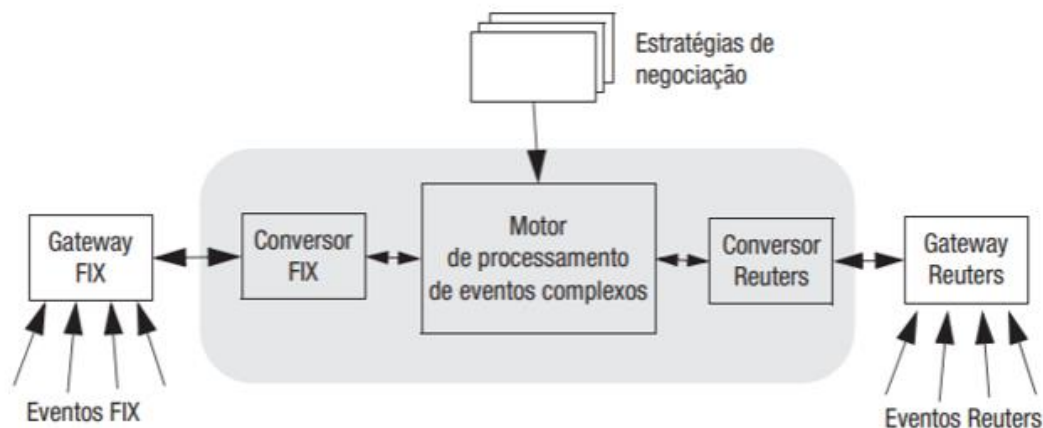
Como indicado por Tanenbaum (2008, p.1), um sistema distribuído é um conjunto de computadores independentes que se apresenta a seus usuários como um sistema único e coerente”.

Cientes da necessidade de um programa eficiente e visando a melhor arquitetura e infraestrutura, deve-se levar em consideração os desafios de SD:

- Segurança;
- Escalabilidade dentre a quantidade de usuários e requisições;
- Desempenho eficaz;
- Comunicação entre ambientes heterogêneos;

Na figura abaixo, em uma instituição financeira com sistema distribuído, é demonstrado um clássico obstáculo referente a heterogeneidade, pois dentre as características presentes, nota-se que os Gateways possuem fonte distintas, mas para adequação há o uso de conversores para uso interno do sistema.

Figura 3



4.3 Microserviços

Microserviços são pequenos serviços independentes que para se comunicar utilizam APIs definidas, facilitando escalabilidade e agilidade com funções básicas que são denominadas serviços. Com isso, em caso de falhas não comprometem o restante dos serviços ativos, pois atuam de maneira independente. Dentre as principais características e vantagens há:

- Serviços implementados com facilidade e rapidez na atualização, pois a integração e a entrega podem ser feitas de maneira contínua;
- Aumento da flexibilidade da infraestrutura, sem necessidade de compartilhar códigos, modelo e *databases* comuns;
- Possibilidade de escalar independentemente garantindo escalabilidade, se tornando dimensionado;
- Estandarização de serviços, para haver padrões de *desing*, comunicação e de integração afim de facilitar e ajudar a lidar com desafios inovadores;
- Redução de custos de escalonamento das aplicações internas;
- Facilidade no modelo operacional entre equipes como infraestrutura e *devops*.

4.4 Web Services

De acordo com o livro Engenharia de Software de Kechi Hirama:

Um serviço pode ser definido como um componente de software reutilizável fracamente acoplado que encapsula funcionalidade discreta e este pode ser distribuído e acessado através de um programa. Existem algumas tecnologias que podem ser usadas para a implementação de SOA: Web Service, Corba, RMI, DCOM, REST etc. No caso de Web Service, um serviço que é acessado usando protocolos padrões da internet e baseados em XML (padrão para troca de informações em ambiente distribuído), o que o torna independente da linguagem usada na sua implementação.

Definido também como um conjunto de métodos acedidos e invocados por outros programas, um *Web Service* possui uma gama de utilidades, levando em consideração a quantidade de protocolos disponíveis para desenvolvimento, tais como REST, e SOAP, é possível utilizar serviços web de diversas formas, através de requisições. Sua utilização trás diversas vantagens, tais como a reutilização de código, por ser utilizável em

diferentes plataformas, redução do tempo de desenvolvimento, segurança, custos baixos e integração de informação e sistemas. Um exemplo de aplicação de *web services* pode ser a utilização de informações de crédito de pessoas disponibilizados a bancos e instituições pela empresa Serasa, validando possibilidades de créditos e dados.

- **APIs**

A API, *Application Programming Interface*, ou Interface de programação de Aplicativos é um padrão de programação para determinadas funcionalidades. Comumente usados para padronizar funções complexas e/ou associar produtos a serviços, funcionalidades essas como acessar uma página de *delivery*, por exemplo, e acompanhar a entrega através do Google *Maps* associado ao aplicativo de entrega. Assim, com as APIs, aplicativos se comunicam entre si sem a necessidade de intervenção do usuário através de diversos códigos de maneira a serem usadas em outras aplicações.

4.5 Container

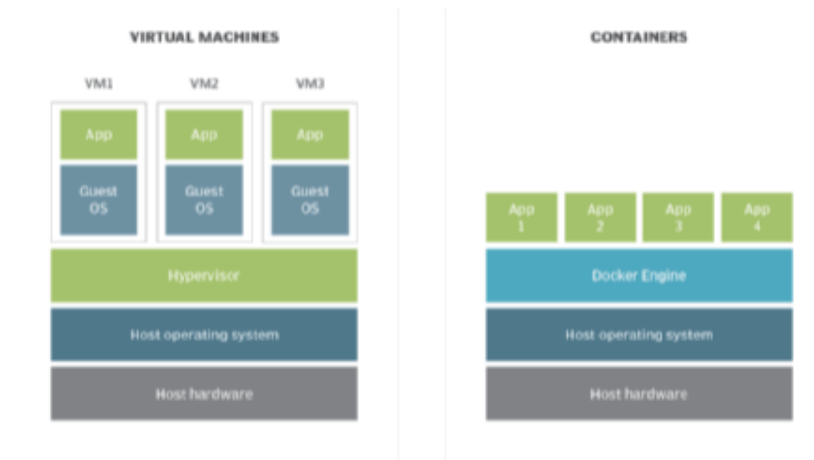
Na conversão para a tecnologia, container ganha o significado de agrupamento, ou seja, a junção de algo e suas dependências. Mais especificadamente, a junção de uma aplicação e suas dependências compartilhando o *kernel* do Sistema Operacional (SO) do *host*.

São imensamente similares a máquinas virtuais, entretanto, com características vantajosas, tais como, ser mais leve, mais fácil, proporcionar melhor desempenho no gerenciamento de recursos e mais integrados ao sistema operacional da máquina hospedeira.

Como demonstrado na imagem, o container armazena a aplicação como um todo, facilitando sua utilização por diversos desenvolvedores, auxiliando equipes de *DevOps* com problemas de compatibilidade e versionamento de *frameworks* dentro de projetos. A portabilidade vem a ser uma importante vantagem na utilização de containers, visto que indiferente ao ambiente em que foi criado, irá funcionar em qualquer outro que possua a tecnologia, ou seja é possível executar as aplicações sem nenhum problema

de dependência, reduzindo problemas internos e discussões, a título de exemplo “Na minha máquina funciona”, situação cotidianamente vivida no mundo empresarial da tecnologia.

Figura 4



De acordo com Jeferson Vitalino e Marcos Castro no livro Descomplicando o Docker,

"O grande passo rumo ao cenário que temos hoje foi a criação, pela Parallels do Virtuozzo, de um painel que permitia o fácil gerenciamento de containers e a disponibilização do core do Virtuozzo como open source com o nome de OpenVZ. O OpenVZ foi uma ferramenta que ganhou bastante destaque no gerenciamento de containers e ajudou e muito na popularização do VPS (Virtual Private Server) e, conseqüentemente, na criação de centenas de empresas de hosting espalhadas pelo mundo. O principal ponto negativo do OpenVZ era a necessidade de aplicar um patch no kernel Linux."

(Jeferson Fernando Noronha Vitalino; Mar..., Descomplicando o Docker)

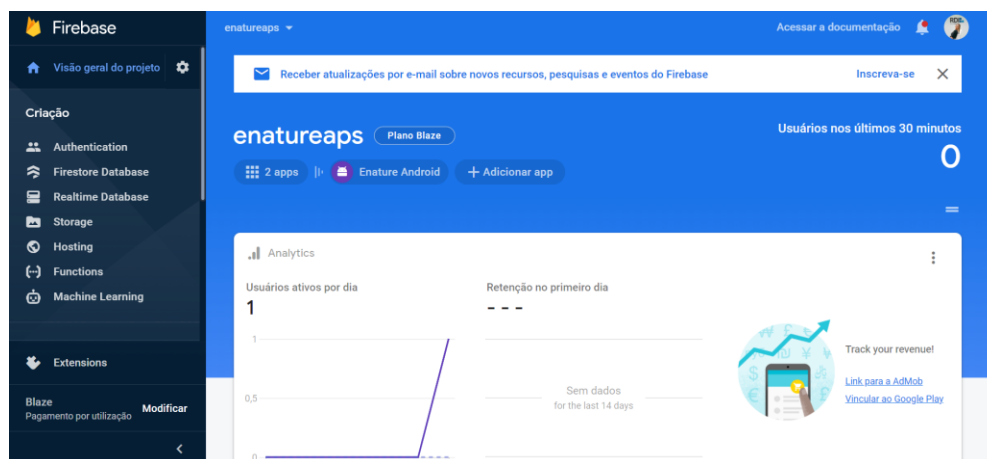
Ou seja, essa tecnologia já é utilizada a muito tempo. Dentre as principais formas de containerização de aplicações, encontramos o Docker e agora o *Firebase*.

4.6 Firebase

A tecnologia de *web service* e container utilizada no projeto foi o *firebase*, que se trata de um container de aplicativos, oferecendo todo o suporte para desenvolvimento e compartilhamento de ferramentas úteis tais como o banco de dados, notificações multiplataformas e o registro de usuários. Seu suporte se estende desde IOS a Android e WEB.

A implantação do *firebase* no projeto configura uma facilidade e unificação de processos e informações. O link entre um aplicativo e o container é feito através da autenticação Auth-2 lincada por uma chave SHA-1 gerada unicamente para cada projeto. Apesar de oferecer uma vasta opção de serviços gratuitos, algumas funcionalidades podem ser pagas dentro da plataforma, que possui uma interface para acesso e controle dos desenvolvedores chamada console.

Figura 5

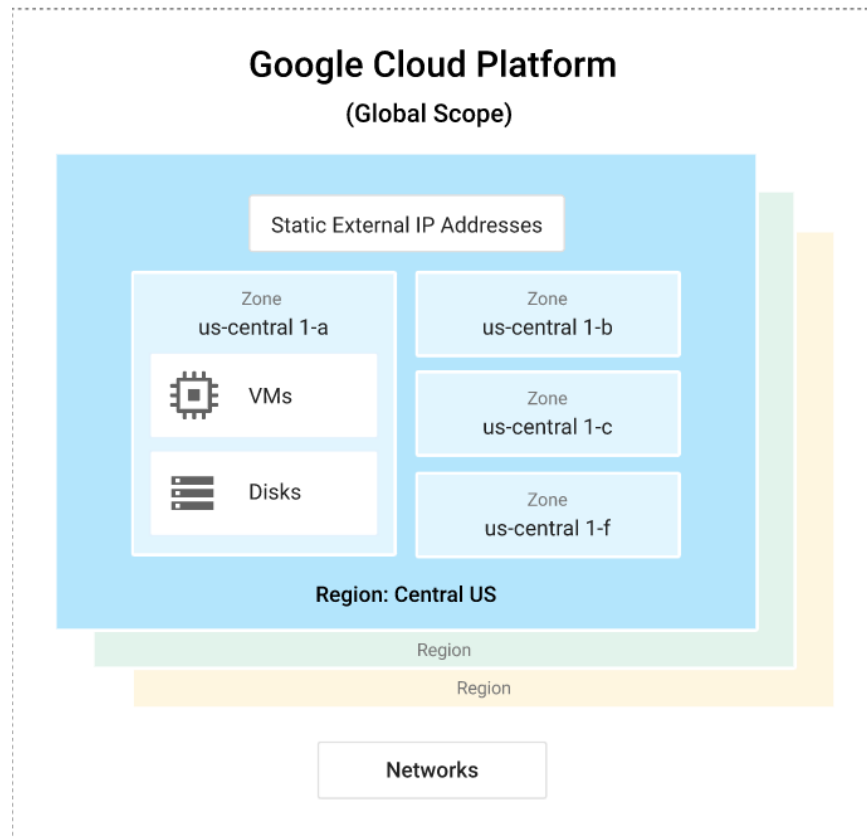


4.7 Google Cloud

A tecnologia do google cloud, de acordo com a plataforma “consiste em um conjunto de recursos físicos (computadores e unidades de disco rígido) e recursos virtuais, como máquinas virtuais (VMs), localizados nos data centers do Google por todo o mundo.” Ou seja, através dele podemos hospedar aplicações e utilizar recursos em um

projeto e ter a segurança de armazenamento e funcionamento em nuvem. Em caso de falhas, o serviço fornece a redundância de serviços.

Figura 6



Fonte: <https://cloud.google.com/docs/overview/?hl=pt-br>

O aplicativo em questão utilizou-se do serviço de *maps* do *google cloud*, adquirindo APIs que fornecem o produto de mapa, usufruindo de um console próprio, tal como o *firebase* citado acima.

Figura 7



4.8 Normas ISO

As normas ISO (*International Organization for Standardization*) tem a função de estabelecer padrões de âmbito global para a entrega de software e a aprimoramento do relacionamento entre o cliente e a empresa. Além de fornecer requisitos que, quando implementados corretamente, garantem o serviço que complete as expectativas dos clientes.

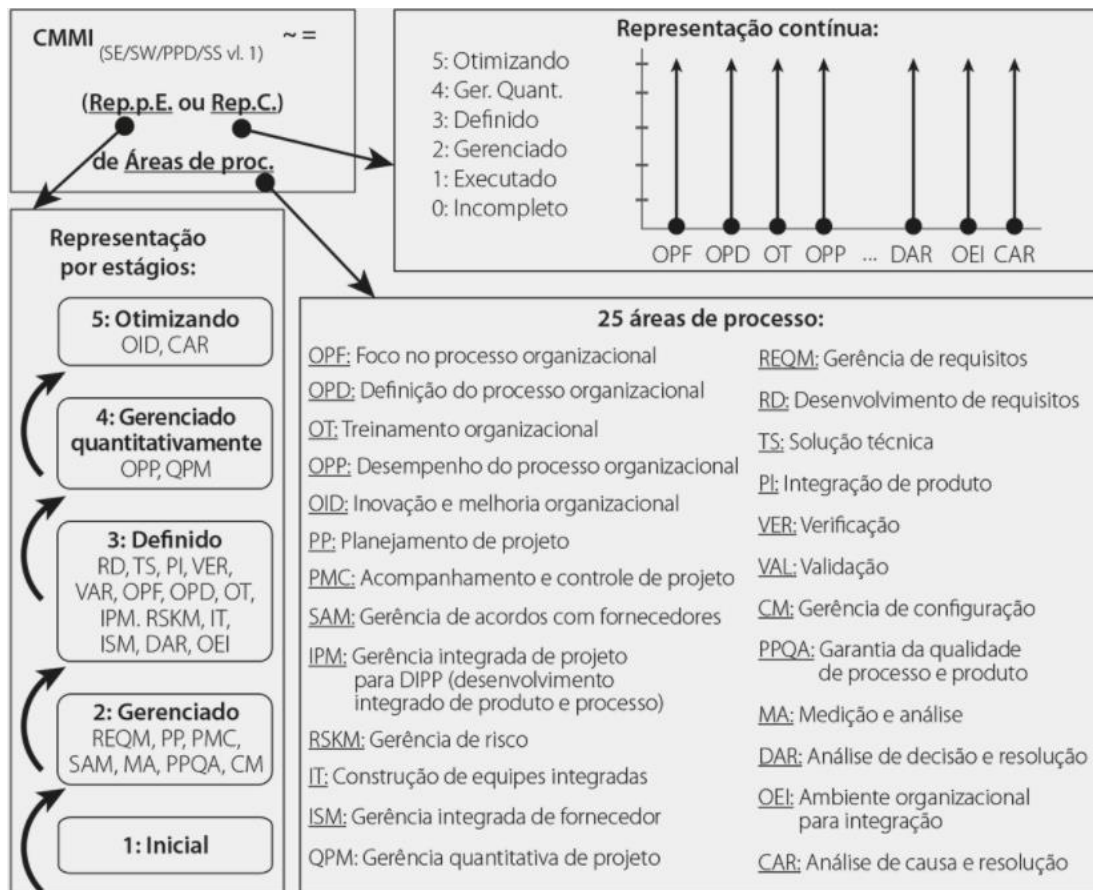
A ISO mais conhecida na questão de qualidade é a 9000, “um padrão internacional que ‘especifica requisitos para um sistema gerencial de qualidade de uma organização’ (VASCONCELOS et al., 2006, p. 88).

- **SEI / CMM**

Software Engineering Institute é o instituto responsável por criar modelos internacionais visando o auxílio em todo o processo de qualidade de software. Dentre os criados, o *Capability Maturity Model for Software* é o mais conhecido, baseado na organização do processo de desenvolvimento, dividindo-o em níveis abstrato e

descritivo, disciplina genérica de projetos, melhoria nos processos do *software*, gerência de qualidade de *software* e garantia de prevenção de erros, porém este modelo foi descontinuado e substituído pelo CMMI (*Capability Maturity Model Integration*).

Figura 8



4.9 DDD

Domain Driven Design (DDD) significa Projeto Orientado a Domínio, Eric Evans, em seu livro *Domain Driven Design* (2004), descreve o DDD como uma maneira de pensar e um conjunto de princípios a fim de auxiliar os desenvolvedores a criar sistemas mais robustos. Não se tratando de uma tecnologia, mas sim uma abordagem de projeto de software que foca no domínio e conhecimento da lógica do sistema.

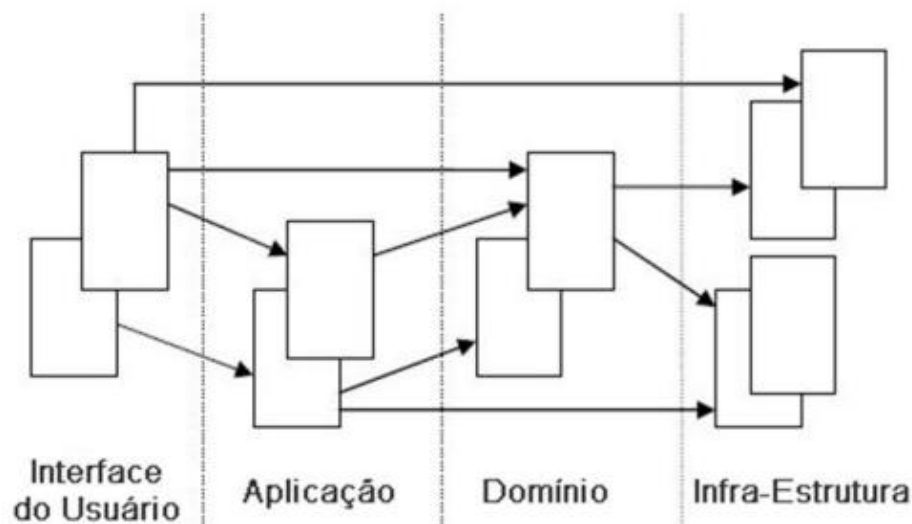
“Com um modelo bem-feito, organizado, as várias partes de um sistema interagem

sem que haja muita dependência entre módulos ou classes de objetos de conceitos distintos.” (CUKIER, s.p,2010).

De acordo com Cukier, o DDD é independente de tecnologia, ele foca em entender as regras de negócio e como elas devem estar refletidas no código e no modelo de domínio.

O DDD depende de um desenvolvimento em camadas, como mostrado na figura abaixo.

Figura 9



*Fonte: AVRAM, MARINESCU, Domain-Driven Design Quickly – p.29.

O âmago do DDD se encontra na camada de domínio, responsável pelas regras de negócio do projeto e de seus casos de uso. A utilização das práticas e princípios do DDD influencia a extensibilidade, usabilidade, testabilidade e manutenibilidade de um sistema, ou seja, cada camada é responsável apenas pelo seu propósito.

Um modelo de domínio pode ser a base de comunicação num projeto de software. Sendo que esta comunicação não está limitada apenas aos diagramas da UML. Os especialistas de um determinado domínio possuem um entendimento limitado do vocabulário técnico, e um vasto vocabulário da sua própria área.

(FRANCO, J.R., et al,2010)

4.10 Segurança do Software

Visando assegurar a satisfação e confiança do usuário houve a adoção de medidas segurança em todo o projeto pois diante dos avanços tecnológicos, em relação aos ambientes virtualizados, notou-se a necessidade de regular e estabelecer as responsabilidades do que é feito com os dados dos usuários tendo em vista as premissas que sustentam a segurança destas informações.

Em sua estrutura, faz-se o uso de técnicas e parâmetros como:

- **Confiabilidade:** Medida preventiva para organização e definição de acessos com autorização e autenticação Auth 2, instituída no login a partir de uma conta google.
- **Integridade:** Garantir que todas as informações estejam em seu formato original. Visando impedir que determinado conteúdo fique exposto e seja manipulado por indivíduos sem autorização.
- **Disponibilidade:** Garantia de acessibilidade a qualquer momento para os usuários, por exemplo: na infraestrutura um *link* temporário de backup para caso o banco de dados caia, realizado frequentemente pela plataforma firebase.

4.10.1 Segurança em Sistemas Distribuídos

Dentre os pilares para que haja segurança em um sistema distribuído, há a necessidade da confiabilidade, integridade, autenticidade, disponibilidade e não repúdio e como ameaças temos a interceptação de dados, indisponibilidade, modificação ou alteração de dados e fabricação maliciosa de acessos.

Dentre os mecanismos de segurança, adotados a autenticação para verificação e identificação das entidades do sistema como por exemplo o usuário para acesso, autorização e auditorias internas visando o bem-estar tecnológico de contribuintes como clientes e fornecedores para aplicação.

Além dos pilares de segurança de um sistema distribuído, visando assegurar a proteção das informações e dados dos clientes, o *software* possui procedimentos internos de segurança, tais como, políticas de gestão de segurança e está em conformidade com a LGPD.

- **Política de gestão de acessos**

Há gestão de controle de acessos na plataforma para melhor organização estrutural e divisão de perfil de acesso:

- Usuário administrador;
- Usuário comum – cliente;
- Usuário comum – fornecedor.

Referente ao usuário com perfil de comum, se tratando de visualização, os indivíduos possuíram visão apenas para áreas previamente delimitadas. Com isso a visão da aplicação como um todo cabe apenas para usuários administradores, como explicado abaixo.

Tipo	Característica	Descrição	Responsabilidade
Usuário administrador.	Administrador da ferramenta.	Usuário com perfil administrador para fazer alterações internas.	Prezar pela funcionalidade da plataforma.
Usuário comum.	Cliente.	Usuário que utiliza a plataforma para descartar de maneira correta os resíduos.	Usar a plataforma com prudência.
Usuário comum.	Fornecedor	Usuário que cadastra seu estabelecimento plataforma o local que recolhe os resíduos.	Usar a plataforma com prudência.

Visando sempre os avanços tecnológicos para promover maior segurança nos sistemas e adequação à LGPD, ao que diz respeito à proteção dos dados dos clientes, o software segue os padrões de políticas e normas de boas praticas seguindo o que rege a lei nº 13.709 que vigorou em 18 de setembro de 2020.

5 DESENVOLVIMENTO DO PROTÓTIPO

5.1 Tecnologias

O protótipo desta atividade prática supervisionada foi desenvolvido em algumas linguagens de programação de acordo com as facilidades, benefícios e recursos disponíveis.

- **Aplicativo**

O desenvolvimento do aplicativo *mobile* foi realizado na linguagem *flutter*. Se trata de uma plataforma de desenvolvimento cruzada que utiliza de uma mesma base de códigos para aplicativos em android e IOS, dando ao desenvolvedor uma facilidade na hora de criar e tornando eficiente o processo de construção do projeto. A linguagem usada no flutter é o Dart, fortemente tipada e criada pelo Google em 2011 com a intenção de substituir o JavaScript no desenvolvimento web, hoje ela já é considerada multi-paradigma e possui algumas variantes.

- **API**

A API (application programming interface) desenvolvida para inserção de dados no banco foi desenvolvida em linguagem ASP .NET (dotnet). É uma plataforma desenvolvida pela Microsoft, para o desenvolvimento de aplicações web é um sucessor das tecnologias ASP. Com uma linguagem integrada no Framework .NET, cria páginas dinâmicas sem a necessidade de uma linguagem de programação PHP ou servidor web como Apache. 26 Seu ambiente de desenvolvimento é o Visual Studio .NET, por já possuir características que auxiliam na facilidade do trabalho por possuir componentes visuais para criação de formulários de página web.

Podendo também reutilizar códigos de outros projetos escritos para a plataforma .NET indiferente a linguagem, diferentemente da tecnologia ASP são compiladas antes da execução, assim gerando um ganho significativo no desempenho. Em geral a diferença clara de .NET para ASP.NET é o fator de que .NET pode desenvolver aplicativos de Windows, web apps e até customizar ferramentas MS como pacote office,

em relação ao ASP.NET pode-se desenvolver programas mobile, como web services , sendo apenas divisões de funcionalidades entre os dois

5.2 Código Fonte

O código fonte completo da aplicação encontra-se no repositório remoto GitHub, no seguinte link: <https://github.com/adriisoliveira/enature>.

- **Autenticação**

Adentrando sobre as partes relevantes do desenvolvimento do projeto, temos a autenticação feita com o usuário google, onde, somente o usuário logado pode ter acesso ao aplicativo se logado. A partir da api do *firebase*, podemos pegar via arquivo de extensão json algumas informações de usuário.

```
class AuthController {
    UserModel? _user;

    UserModel get user => _user!;

    void setUser(BuildContext context, UserModel? user) {
        if (user != null) {
            saveUser(user);
            _user = user;
            Navigator.pushReplacementNamed(context, "/home", arguments:
user);
        } else {
            Navigator.pushReplacementNamed(context, "/login");
        }
    }

    Future<void> saveUser(UserModel user) async {
        final instance = await SharedPreferences.getInstance();
        await instance.setString("user", user.toJson());
    }
}
```

```

        return;
    }

    Future<void> currentUser(BuildContext context) async {
        final instance = await SharedPreferences.getInstance();
        await Future.delayed(Duration(seconds: 2));
        if (instance.containsKey("user")) {
            final json = instance.get("user") as String;
            setUser(context, UserModel.fromJson(json));
            return;
        } else {
            setUser(context, null);
        }
    }
}

```

- **Places Search**

A seguinte classe, através de um arquivo json captura a descrição e o código de identificação (id) do local buscado. Tais dados são fornecidos pela API do google maps utilizada no software. Passível de consulta apenas por chave.

```

class PlaceSearch {
    final String description;
    final String placeId;

    PlaceSearch({required this.description, required this.placeId});

    factory PlaceSearch.fromJson(Map<String, dynamic> json) {
        return PlaceSearch(
            description: json['description'], placeId: json['place_id']);
    }
}

```

```
}
```

- Places Service

A classe *places service* faz a requisição para a api do *google maps*, por questão de segurança, a chave foi removida do código, mas o seguinte trecho insere a chave de acesso junto com o local digitado pelo usuário, a partir do retorno, a classe acima trás as informações necessárias.

```
import 'package:payflow/models/place_search.dart';

class PlacesService {
  final key = '';

  Future<List<PlaceSearch>> getAutocomplete(String search) async {
    final String apiEndpoint =
      'https://maps.googleapis.com/maps/api/place/autocomplete/json?
input=$search&types=(cities)&key=$key';
    final Uri url = Uri.parse(apiEndpoint);
    final response = await http.get(url);
    var json = convert.jsonDecode(response.body);
    var jsonResults = json['predictions'] as List;
    return jsonResults.map((place) =>
PlaceSearch.fromJson(place)).toList();
  }
}
```

- Geolocalização

A classe de geolocalização fica responsável, a partir da API, buscar a localização atual do aparelho em questão.


```
import 'package:geolocator/geolocator.dart';

class GeolocatorService {
  Future<Position> getCurrentLocation() async {
    return await Geolocator.getCurrentPosition(
      desiredAccuracy: LocationAccuracy.high);
  }
}
```

- Login Controller

A classe de *LoginController* é responsável por capturar o usuário e fazer o processo de autenticação já descrito antes, por questão de segurança, das informações fornecidas pelo *Google* no momento da autenticação, é capturado apenas o nome e a foto do usuário.

```
class LoginController {
  final authController = AuthController();
  Future<void> googleSignIn(BuildContext context) async {
    GoogleSignIn _googleSignIn = GoogleSignIn(
      scopes: [
        'email',
      ],
    );
    try {
      final response = await _googleSignIn.signIn();
      final user =
        UserModel(name: response!.displayName!, photoURL:
response.photoUrl);
      authController.setUser(context, user);
      print(response);
    } catch (error) {
```

```

    authController.setUser(context, null);
    print(error);
  }
}
}

```

- **Maps API**

A classe *MapsApi* é responsável pela renderização e atualização do mapa dentro do aplicativo, de maneira a chamar as funcionalidades da *place_search* e *geolocator*.

```

class MapsApi extends StatefulWidget {
  const MapsApi({Key? key}) : super(key: key);

  @override
  _MapsApiState createState() => _MapsApiState();
}

class _MapsApiState extends State<MapsApi> {
  final _logger = Logger("maps_api");
  final controller = HomeController();
  late GoogleMapController mapController;
  void _onMapCreated(GoogleMapController controller) {
    mapController = controller;
  }

  @override
  Widget build(BuildContext context) {
    final applicationBloc = Provider.of<ApplicationBloc>(context);

    return Scaffold(
      appBar: AppBar(
        title: Text('Encontre seu local'),

```

```

        backgroundColor: AppColors.primary),
body: (applicationBloc.currentLocation == null)
? Center(child: CircularProgressIndicator(),)
: ListView(
  children: [
    TextField(
      decoration: InputDecoration(
        hintText: 'Busque o endereço',
        icon: Icon(Icons.search)
      ),
    ),
    Container(
      height: 500.0,
      child: GoogleMap(
        mapType: MapType.normal,
        myLocationEnabled: true,
        initialCameraPosition: CameraPosition(
          target:
            LatLng(applicationBloc.currentLocation.latitude, applic
ationBloc.currentLocation.longitude),
          zoom: 14,
        ),
      ),
    ),
  ],
),
bottomNavigationBar: Container(
  height: 90,
  child: Row(
    mainAxisAlignment: MainAxisAlignment.spaceEvenly,
    children: [
      IconButton(

```

```

        onPressed: () {
          controller.setPage(0);
          setState(() {});
        },
        icon: Icon(Icons.home,
          color: controller.currentPage == 0
            ? AppColors.primary: AppColors.body)),
        GestureDetector(
          onTap: () {
            Navigator.push(context,
              MaterialPageRoute(builder: (context) =>
SearchPage())));
          },
          child: Container(
            width: 56,
            height: 56,
            decoration: BoxDecoration(
              color: AppColors.primary,
              borderRadius: BorderRadius.circular(5)),
            child: Icon(
              Icons.search_sharp,
              color: AppColors.background,
            ),
          ),
        ),
        IconButton(
          onPressed: () {
            Navigator.push(context,
              MaterialPageRoute(builder: (context) =>
MapsApi())));
          },
          icon: Icon(Icons.map_sharp,
            color: controller.currentPage == 1

```

```

        ? AppColors.primary
        : AppColors.body))
    ],
  ),
),
);
}
}

```

- Search Page

A classe *SearchPage* é responsável por renderizar os dados no banco a partir de uma busca do usuário, tais dados inseridos via API.

```

class SearchPage extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    final controller = HomeController();
    late GoogleMapController mapController;

    final LatLng _center = const LatLng(45.521563, -122.677433);

    void _onMapCreated(GoogleMapController controller) {
      mapController = controller;
    }

    final tabela = PontosRepositories.tabela;

    final String aba;
    List<Data> dataList = [];
    DatabaseReference referenceData =
      FirebaseDatabase.instance.reference().child("Data");
    referenceData.once().then((DataSnapshot dataSnapshot) {

```



```

        borderRadius:
            BorderRadius.all(Radius.circular(25.0))
    )
)
),
),
dataList.length == 0
    ? Center(child: Text("Not aval"))
    : ListView.builder(itemBuilder: (context, index) {
        return CardUI(
            dataList[index].imgUrl,
            dataList[index].name,
            dataList[index].material,
            dataList[index].price,
        );
    })
],
),
),
bottomNavigationBar: Container(
    height: 90,
    child: Row(
        mainAxisAlignment: MainAxisAlignment.spaceEvenly,
        children: [
            IconButton(
                onPressed: () {
                    controller.setPage(0);
                    setState(() {});
                },
                icon: Icon(Icons.home,
                    color: controller.currentPage == 0
                        ? AppColors.primary: AppColors.body)),
            GestureDetector(

```

```

        onTap: () {
          Navigator.push(context,
            MaterialPageRoute(builder: (context) =>
SearchPage())));
        },
        child: Container(
          width: 56,
          height: 56,
          decoration: BoxDecoration(
            color: AppColors.primary,
            borderRadius: BorderRadius.circular(5)),
          child: Icon(
            Icons.search_sharp,
            color: AppColors.background,
          ),
        ),
      ),
      IconButton(
        onPressed: () {
          Navigator.push(context,
            MaterialPageRoute(builder: (context) =>
MapsPlace())));
        },
        icon: Icon(Icons.map_sharp,
          color: controller.currentPage == 1
            ? AppColors.primary
            : AppColors.body))
    ],
  ),
),
);
}

```



```

Widget CardUI(String imgUrl, String name, String material, String
prince) {
  return Card(
    margin: EdgeInsets.all(15),
    color: Color(0xfffff2fc3),
    child: Container(
      color: Colors.white,
      margin: EdgeInsets.all(1.5),
      padding: EdgeInsets.all(10),
      child: Column(children: <Widget>[
        Text(
          name,
        ),
        SizedBox(
          height: 1,
        ),
        Container(
          width: double.infinity,
          child: Text(
            prince,
          ),
        ),
        SizedBox(
          height: 1,
        ),
      ]
    ),
  );
}

void setState(Null Function() param0) {}
}

```

- Main

E por fim, a classe *main*, responsável pela execução do código, ela inicia o serviço do firebase, onde estão os processos de autenticação e banco de dados, hospedagem e contenarização do aplicativo, caso algo falhe, é registrado no serviço de log que fica salvo no *backend* do *app*.

```
void main() {
  runApp(AppFirebase());
}

class AppFirebase extends StatefulWidget {
  @override
  _AppFirebaseState createState() => _AppFirebaseState();
}

class _AppFirebaseState extends State<AppFirebase> {
  final Future<FirebaseApp> _initialization =
    Firebase.initializeApp();
  final _logger = Logger('main');

  @override
  Widget build(BuildContext context) {
    return ChangeNotifierProvider(
      create: (context) => ApplicationBloc(),
      child: FutureBuilder(
        future: _initialization,
        builder: (context, snapshot) {
          if (snapshot.hasError) {
            return Material(
              child: Center(
                child: Text(
```

```

        "Não foi possível inicializar o Firebase",
        textDirection: TextDirection.ltr,
    ),
),
);
} else if (snapshot.connectionState == ConnectionState.done)
{
    return AppWidget();
} else {
    return Material(
        child: Center(
            child: CircularProgressIndicator(),
        ),
    );
}
}),
);
}
}

```

- Dependências usadas

A seguir, algumas dependências relevantes que foram utilizadas no desenvolvimento do aplicativo.

```

cupertino_icons: ^1.0.2
firebase_core: ^1.3.0
google_sign_in: ^5.1.1
google_sign_in_web: ^0.10.0+3
shared_preferences: ^2.0.8
google_maps_flutter: ^2.0.0
bloc_pattern: ^3.0.0
rxdart: ^0.27.2

```

```

flutter_polyline_points: ^1.0.0
dio: ^4.0.0
latlng: ^0.1.0
flutter_map: ^0.14.0
cloud_firestore: ^2.5.4
firebase_database: ^8.0.1
provider: ^6.0.1
geolocator: ^7.7.1
http: ^0.13.4
flutter_logs: ^2.1.4
logging: ^1.0.2

```

- **Registros de acesso e Logs**

Como falado acima, sobre os acessos ao aplicativo, os registros ficam salvos no back-end do software, porém, os acessos as APIs são salvas diretamente no cloud do google, algumas informações são criptografadas para a segurança do usuário.

Iniciando pelo tráfego geral pelo aplicativo, temos um gráfico e a tabela responsáveis por indicar o serviço.

Figura 10

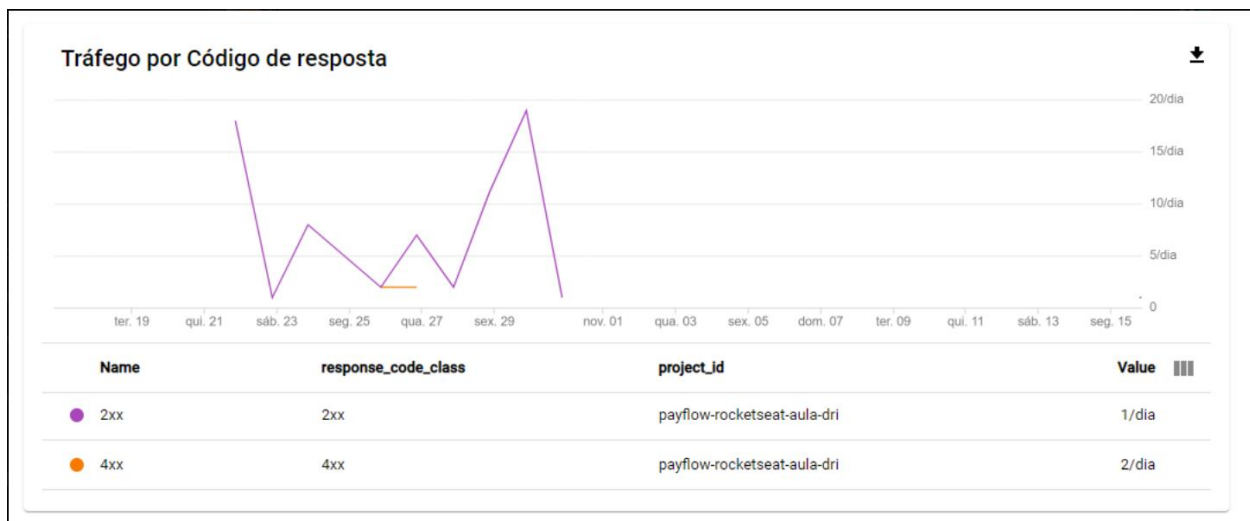


Tabela 1

Start Time	End Time	2xx	4xx
2021-10-21T00:00:00.000Z	2021-10-22T00:00:00.000Z	18	
2021-10-22T00:00:00.000Z	2021-10-23T00:00:00.000Z	1	
2021-10-23T00:00:00.000Z	2021-10-24T00:00:00.000Z	8	
2021-10-25T00:00:00.000Z	2021-10-26T00:00:00.000Z	2	2
2021-10-26T00:00:00.000Z	2021-10-27T00:00:00.000Z	7	2
2021-10-27T00:00:00.000Z	2021-10-28T00:00:00.000Z	2	
2021-10-28T00:00:00.000Z	2021-10-29T00:00:00.000Z	11	
2021-10-29T00:00:00.000Z	2021-10-30T00:00:00.000Z	19	
2021-10-30T00:00:00.000Z	2021-10-31T00:00:00.000Z	1	

Assim como também temos os mesmos registros para erros encontrados durante o acesso.

Figura 11



Tabela 2

Start Time	End Time	4xx
2021-10-25T00:00:00.000Z	2021-10-26T00:00:00.000Z	0.5
2021-10-26T00:00:00.000Z	2021-10-27T00:00:00.000Z	0.222222

A latência do aplicativo durante a utilização:

Figura 12



Tabela 3

Start Time	End Time	Todos os cÃ³digos de resposta
2021-10-25T00:00:00.000Z	2021-10-26T00:00:00.000Z	0.032768
2021-10-26T00:00:00.000Z	2021-10-27T00:00:00.000Z	0.024576
2021-10-27T00:00:00.000Z	2021-10-28T00:00:00.000Z	0.049152
2021-10-29T00:00:00.000Z	2021-10-30T00:00:00.000Z	0.04096

E finalizando, temos o registro de utilização das APIs, sendo elas bem-sucedidas ou falhadas.

Figura 13

APIs				
API ↑	Solicitações	Erros	Latência média (ms)	99% de latência (ms)
Directions API	-	-	-	-
Distance Matrix API	-	-	-	-
Geocoding API	-	-	-	-
Geolocation API	-	-	-	-
Maps Elevation API	-	-	-	-
Maps Embed API	-	-	-	-
Maps JavaScript API	-	-	-	-
Maps SDK for Android	66	-	-	-
Maps SDK for iOS	-	-	-	-
Maps Static API	-	-	-	-
Places API	8	4	37	65
Roads API	-	-	-	-
Street View Static API	-	-	-	-
Time Zone API	-	-	-	-

Linhas por página: 50 ▾ 1 – 14 de 14 < >

5.3 Protótipo

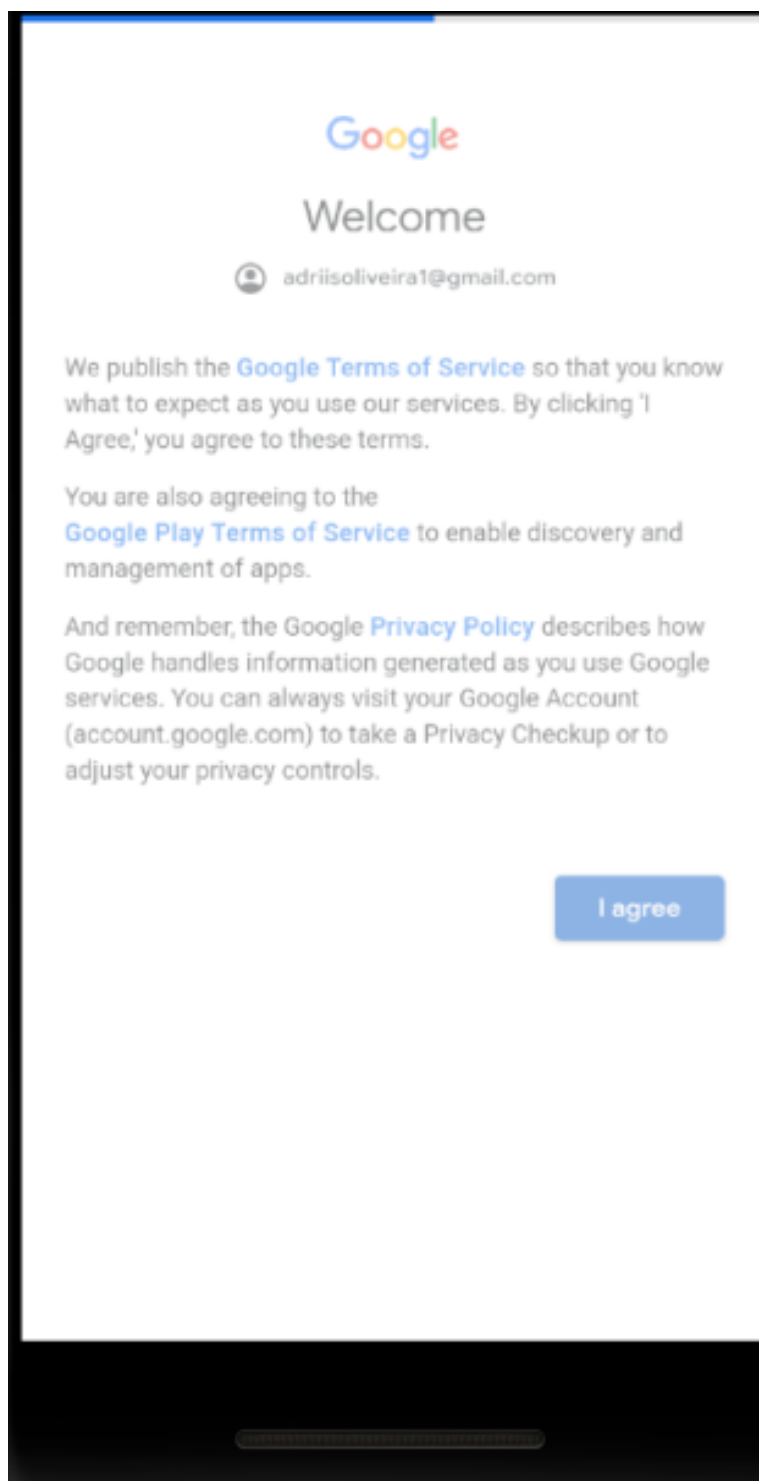
A seguir, a apresentação do protótipo funcionando, inicialmente temos a primeira tela em que é disponível ao usuário o login via conta google.

Figura 14



Logo em seguida, a autenticação sendo feita pelo google, ou seja, a permissão de acesso aos dados do usuário.

Figura 15



Na tela inicial, apresentada nas figuras (16 a 21) a seguir, temos algumas informações sobre o lixo eletrônico e a importância de seu descarte, assim com as consequências do descarte incorreto e algumas informações da empresa fictícia que o presente grupo criou.

Figura 16



Figura 17



Figura 18



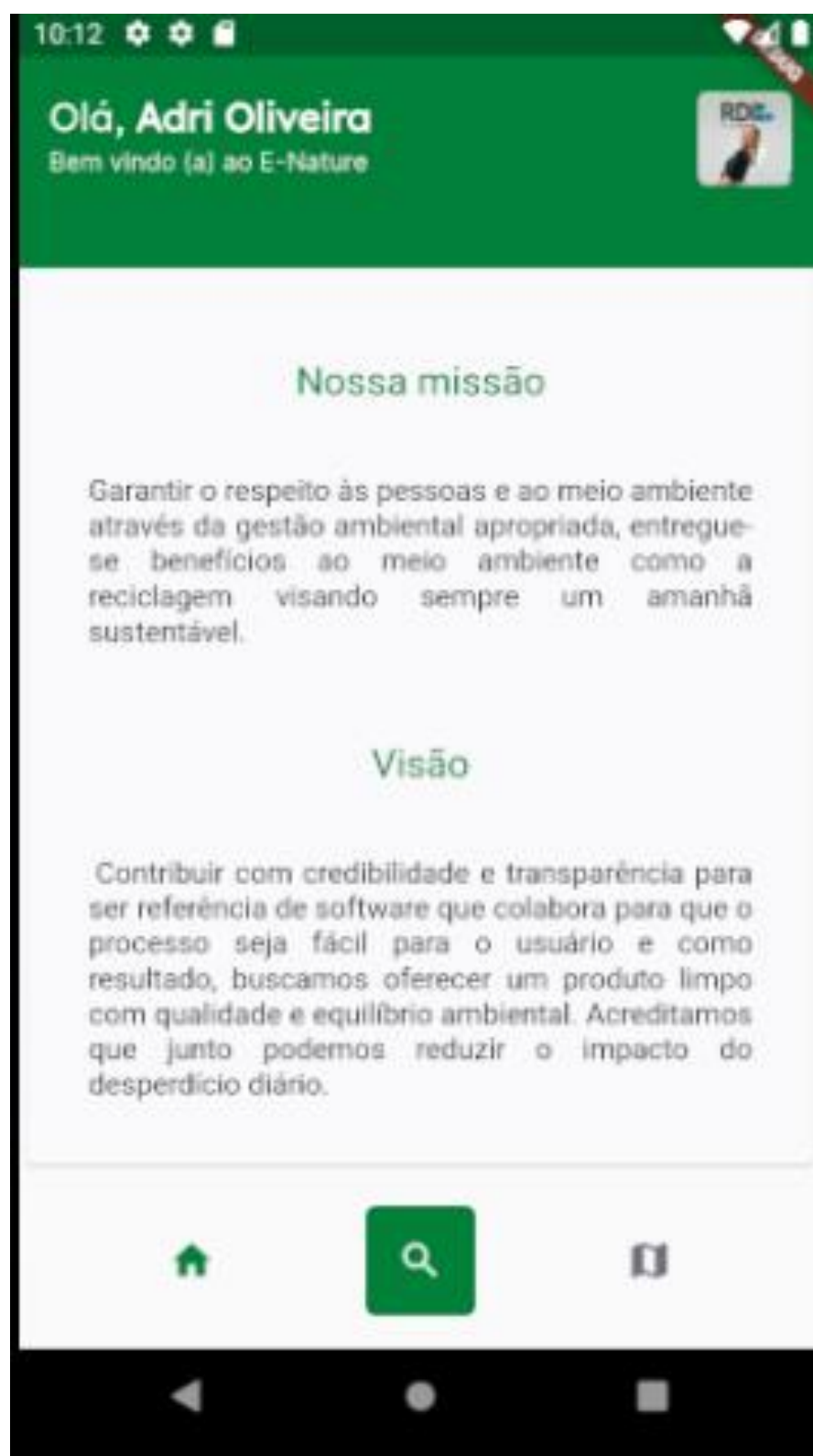
Figura 19



Figura 20

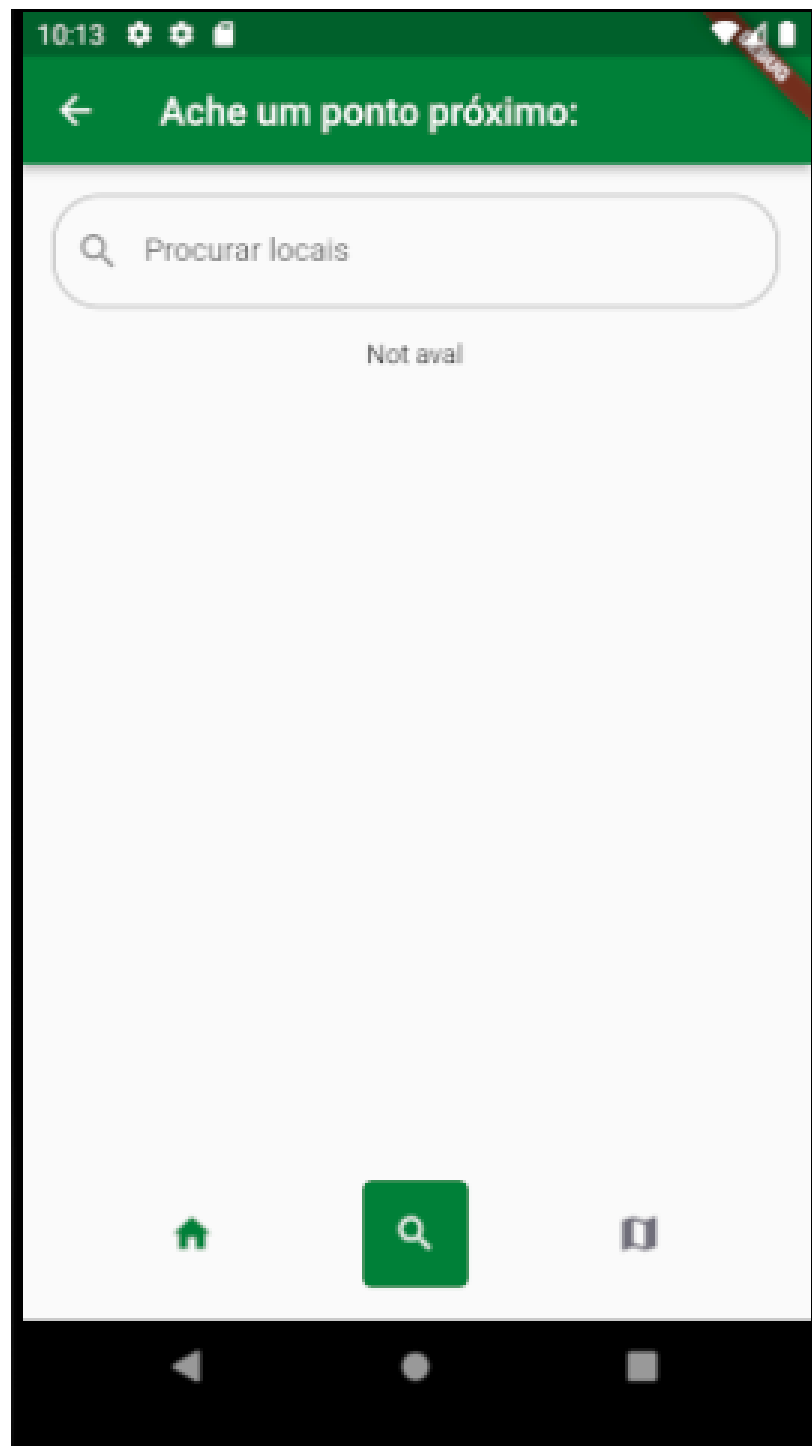


Figura 21



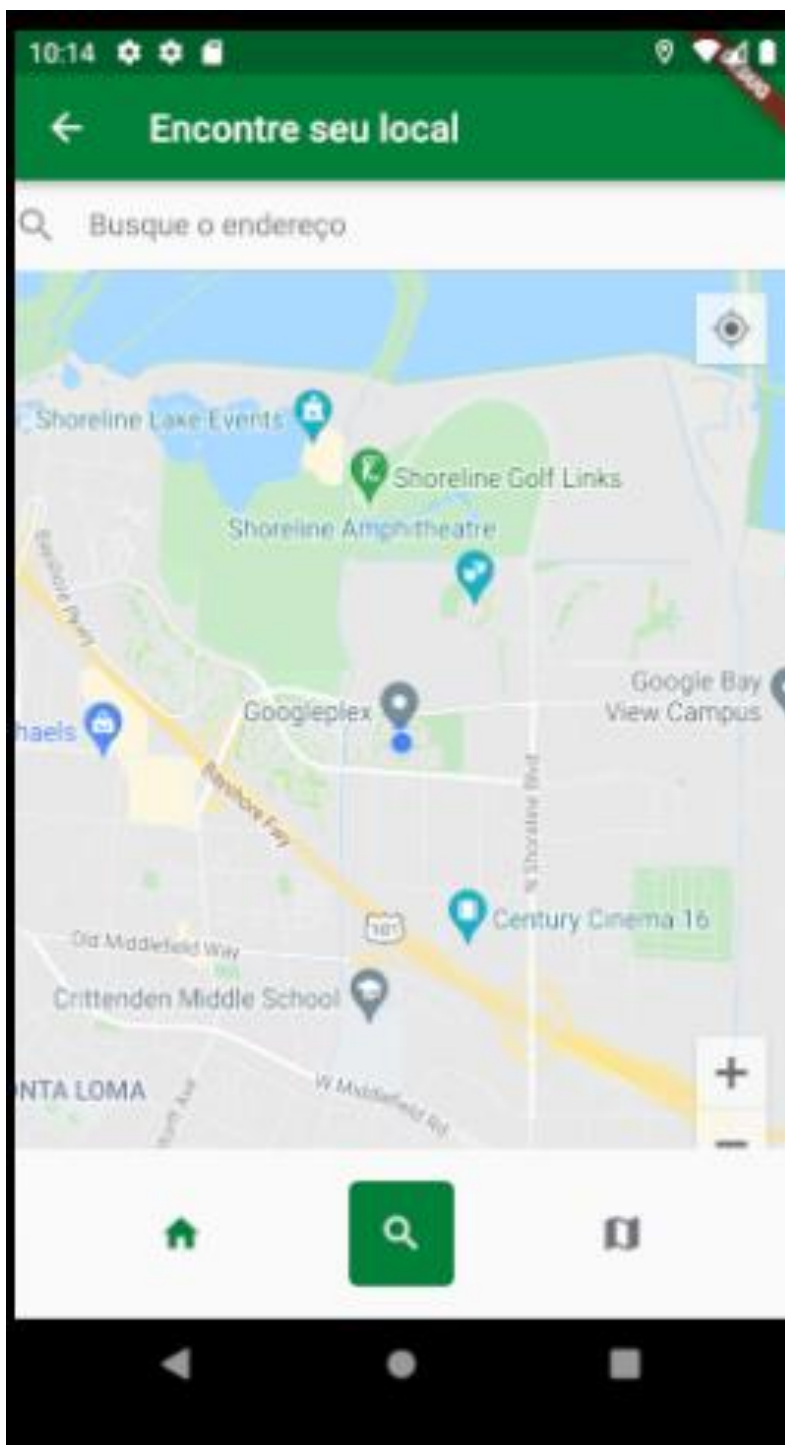
Na próxima figura, temos a página de busca do protótipo, onde, o usuário digita sua cidade e o aplicativo busca no banco de dados os locais cadastrados até o momento.

Figura 22



E finalizando temos a dela de renderização do mapa, apresentando na imagem, a localização atual do dispositivo, por ser um emulador *android* ela apresenta-se na sede do google.

Figura 23



6 CONCLUSÃO

Este documento abrange de forma completa o desenvolvimento do software E-Nature, uma aplicação mobile que consome informações de web services, apresentando importantes pontos do que compõem o sistema.

Desta maneira, atingimos de forma objetiva e clara todos os objetivos esperados e propostos no início do projeto, compreendendo de forma assertiva todos os conceitos apresentados durante as pesquisas construção dos documentos requisitados, acrescentando novos pontos aos conhecimentos já adquiridos durante as aulas.

REFERÊNCIAS

- AUTARQUIA ESPECIAL MUNICIPAL DE LIMPEZA URBANA. **Autarquia Especial Municipal de Limpeza Urbana – Emlur**. Disponível em: <https://www.joaopessoa.pb.gov.br/secretaria/emlur/>. Acesso em: 3 out. 2021.
- BLOGGER . **Segurança em sistemas distribuídos**. Disponível em: <https://dirceuprofessor.blogspot.com/2015/05/seguranca-em-sistemas-distribuidos.html>. Acesso em: 16 out. 2021.
- CLARO RECICLA. **CLARO RECICLA**. Disponível em: <https://www.claro.com.br/institucional/claro-recicla>. Acesso em: 1 set. 2021.
- CONVISO . **6 dicas de segurança no desenvolvimento de softwares**. Disponível em: <https://blog.convisoappsec.com/seguranca-no-desenvolvimento-de-softwares/>. Acesso em: 12 out. 2021.
- COOPREL ALAGOAS. **Cooprel Alagoas**. Disponível em: <http://cooprelalagoas.blogspot.com/>. Acesso em: 9 out. 2021.
- DATUM . **SAIBA MAIS SOBRE SEGURANÇA DO SOFTWARE E CRIPTOGRAFIA**. Disponível em: <https://blog.datum.inf.br/seguranca-do-software/> <https://www.objective.com.br/insights/seguranca-no-desenvolvimento-de-software/>. Acesso em: 16 out. 2021.
- DDD. **Introdução a Domain Driven Design**. Disponível em: <https://pt.slideshare.net/danicuki/ddd-domain-driven-design>. Acesso em: 5 set. 2021.
- DESCARTE INCORRETO DE LIXO ELETRÔNICO PODE CAUSAR DANOS AO MEIO AMBIENTE. **DESCARTE INCORRETO DE LIXO ELETRÔNICO PODE CAUSAR DANOS AO MEIO AMBIENTE**. Disponível em: <http://portal.metodista.br/rpcom/noticias/2013/descarte-incorreto-de-lixo-eletronico-pode-causar-danos-ao-meio-ambiente>. Acesso em: 2 set. 2021.
- ECODESCARTE. **Ecodescarte**. Disponível em: <https://www.ecodescarte.com/>. Acesso em: 2 out. 2021.
- ECODIGITAL. **Ecodigital**. Disponível em: <https://www.ecycle.com.br/postos/cooperativa/pe/ecodigital/20653>. Acesso em: 3 out. 2021.
- ECOLETAS . **Ecoletas** . Disponível em: <http://www.ecoletas.eco.br/>. Acesso em: 3 out. 2021.
- ECYCLE . **PROCESSOS POR TRÁS DA RECICLAGEM**. Disponível em: <https://www.ecycle.com.br/component/content/article/44-guia-da-reciclagem/1823-entenda-os-processos-por-tras-da-reciclagem-de-equipamentos-eletronicos.html>. Acesso em: 5 set. 2021.
- ECYCLE. **GUIA DA RECICLAGEM**. Disponível em: <https://www.ecycle.com.br/component/content/article/44-guia-da-reciclagem/2014-como-descartar-descarte-jogar-fora-residuo-lixo-eletronicos-equipamentos-eletronicos-moveis-eletronicos-velho-antigo-obsoleto-coleta-recolhimento-domicilio-sem-sair-casa-forma-correta-sustentavel-responsavel-se>. Acesso em: 5 set. 2021.

ESAMBIENTAL. **Esambiental**. Disponível em: <http://esambiental.com.br/>. Acesso em: 3 out. 2021.

FIREBASE GOOGLE DOCUMENTATION. **Documentation**. Disponível em: <https://firebase.google.com/support/faq?hl=pt-br>. Acesso em: 23 set. 2021.

FRAGMAQ. **O QUE PODE SER CONSIDERADO LIXO ELETRÔNICO E QUAIS SEUS PRINCIPAIS PROBLEMAS**. Disponível em: <http://www.fragmaq.com.br/blog/o-que-pode-ser-considerado-lixo-eletronico-e-quais-sao-seus-principais-problemas/>. Acesso em: 15 set. 2021.

GALLOTI, A, G. M. **Qualidade de Software**. 1. ed. São Paulo: Pearson, 2016. p. 1-1.

GERENCIAMENTO DE LIXO ELETRÔNICO NO BRASIL. **GERENCIAMENTO DE LIXO ELETRÔNICO NO BRASIL**. Disponível em: <https://techinbrazil.com.br/gerenciamento-de-lixo-eletronico-no-brasil>. Acesso em: 5 set. 2021.

GERVASRECICLAGEMDIGITAL. **Gervasreciclagemdigital**. Disponível em: <http://gervasreciclagemdigital.com.br/>. Acesso em: 2 out. 2021.

GLOBO. **SAIBA ONDE DESCARTAR O LIXO ELETRÔNICO EM SÃO PAULO**. Disponível em: <http://g1.globo.com/sao-paulo/sao-paulo-mais-limpa/noticia/2012/04/saiba-onde-descartar-lixo-eletronico-na-grande-sao-paulo.html>. Acesso em: 19 set. 2021.

GOOGLE DOCUMENTATION. **Google Cloud**. Disponível em: https://www.google.com/search?q=como+funciona+o+google+cloud&rlz=1C1GCEU_enBR979BR979&oq=como+funciona+o+google+cloud&aqs=chrome..69i57j0i22i30l3.5248j0j1&sourceid=chrome&ie=UTF-8. Acesso em: 23 set. 2021.

GUAMÁ. **Pontos de coletas de lixo eletrônico em Belém (PA)**. Disponível em: <http://blog.guamatratamentoderesiduos.com.br/pages/dicas>. Acesso em: 2 out. 2021.

INSTITUTO FEDERAL DE GOIÁS. **Câmpus Goiânia tem ponto de coleta para descarte correto de resíduo eletrônico**. Disponível em: <https://www.ifg.edu.br/servidor/17-ifg/ultimas-noticias/3766-ifg-campus-goiania-tem-ponto-de-coleta-de-residuo-eletronico-2>. Acesso em: 2 out. 2021.

INSTITUTO NET CLARO EMBRATEL. **CARTILHA LIXO ELETRÔNICO**. Disponível em: https://www.institutonetclaroembratel.org.br/cidadania/wp-content/uploads/sites/3/2017/08/Cartilha_01_0_ClaroRecicla_148x210mm.pdf. Acesso em: 12 set. 2021.

IUGU. **O que é flutter**. Disponível em: <https://www.iugu.com/blog/o-que-e-flutter>. Acesso em: 16 set. 2021.

LI XO ELETRÔNICO. **Lixo Eletrônico**. Disponível em: <https://assintec.com.br/produto/lixo-eletronico-2/>. Acesso em: 3 out. 2021.

NATALRECICLAGEM. **Onde descartar Lixo Eletrônico?**. Disponível em: <https://natalreciclagem.com.br/portal/lixoeletronico/>. Acesso em: 3 out. 2021.

PROFº FERNANDO DE SIQUEIRA. **PROFº FERNANDO DE SIQUEIRA**. Disponível em: <https://sites.google.com/site/proffdesiqsistemasdistribuidos/aulas/10-seguranca-em-sistemas-distribuidos>. Acesso em: 10 out. 2021.

RECICLE. **Coletamos seu lixo eletrônico gratuitamente!**. Disponível em: <http://www.reciclelogisticareversa.com.br/>. Acesso em: 2 out. 2021.

RECOMERCIO. **O QUE É RECOMMERCE.** Disponível em: <http://www.recomercio.com.br/o-que-e-recommerce/>. Acesso em: 26 set. 2021.

REVISTAECOLOGICO. **LIXO ELETRÔNICO.** Disponível em: <http://www.revistaecologico.com.br/materia.php?id=110&secao=1930&mat=2232>. Acesso em: 11 set. 2021.

SEBRAE. **DESCARTE CORRETAMENTE OS REJEITOS ELETRÔNICOS.** Disponível em: <http://www.sebrae.com.br/sites/PortalSebrae/artigos/descarte-corretamente-os-rejeitos-eletronicos,0cedd15a9567d410VgnVCM1000003b74010aRCRD>. Acesso em: 19 set. 2021.

SEBRAE. **LIXO QUE NÃO É LIXO.** Disponível em: <http://www.mg.agenciasebrae.com.br/sites/asn/uf/MG/o-lixo-que-nao-e-lixo,a59090a0d2816410VgnVCM1000003b74010aRCRD>. Acesso em: 19 set. 2021.

SETEAMBIENTAL. **Seteambiental.** Disponível em: <https://www.seteambiental.com.br/>. Acesso em: 3 out. 2021.

SIGNIFICADOS. **LIXO ELETRÔNICO.** Disponível em: <https://www.significados.com.br/lixo-eletronico/>. Acesso em: 15 set. 2021.

SOARES; S, A. K. M. D. **Qualidade de Software:** Aprenda as metodologias e técnicas mais modernas para o desenvolvimento de software. 2. ed. São Paulo: Novatec, 2006. p. 25-41.

STACK DESENVOLVEDOR. **O que é Docker?.** Disponível em: <https://stack.desenvolvedor.expert/appendix/docker/oquee.html>. Acesso em: 5 set. 2021.

SUAPESQUISA. **O QUE É LIXO ELETRÔNICO.** Disponível em: https://www.suapesquisa.com/o_que_e/lixo_eletronico.htm. Acesso em: 15 set. 2021.

SUCATA DIGITAL. **SUCATA DIGITAL.** Disponível em: <http://www.sucatadigital.com.br/quem-somos/>. Acesso em: 12 set. 2021.

TECMUNDO. **COMO OS PRINCIPAIS COMPONENTES ELETRÔNICOS SÃO RECICLADOS.** Disponível em: <https://www.tecmundo.com.br/produtos/37275-como-os-principais-componentes-de-eletronicos-sao-recicladados-.htm>. Acesso em: 19 set. 2021.

TODAMATERIA. **LIXO ELETRÔNICO.** Disponível em: <https://www.todamateria.com.br/lixo-eletronico/>. Acesso em: 15 set. 2021.

TREINA WEB. **O que é dart.** Disponível em: <https://www.treinaweb.com.br/blog/o-que-e-dart>. Acesso em: 16 set. 2021.

VENDERMEUCELULAR. **RECICLAGEM DE LIXO ELETRÔNICO NO BRASIL.** Disponível em: <http://www.vendermeucelular.com.br/reciclagem-de-lixo-eletronico-brasil/>. Acesso em: 26 set. 2021.

VITALINO; CASTRO, Jeferson; MARCUS. **Descomplicando o Docker.** 1. ed. São Paulo: Editora Brasport, 2016. p. 1-120.

7 FICHAS

7.1 Ficha Adrienne



FICHA DAS ATIVIDADES PRÁTICAS SUPERVISIONADAS - APS

NOME: Adrienne Sousa Oliveira
CURSO: Ciência da Computação
DIGO DA ATIVIDADE: 53A6

TURMA: CC8P13
CAMPUS: Marquês
SEMESTRE: Oitavo

R.A.: D7048D1
SEMESTRE: 8º Semestre
ANO GRADE: 2021

TURNO: Noturno

DATA DA ATIVIDADE	DESCRIÇÃO DA ATIVIDADE	TOTAL DE HORAS	ASSINATURA DO ALUNO	HORAS ATRIBUÍDAS (1)	ASSINATURA DO PROFESSOR
25 de agosto	Escolha do tema e divisão de tarefas	3h			
01 de setembro	Início das pesquisas teóricas	6h45			
12 de setembro	Pesquisas de conteúdo teórico	5h40			
22 de setembro	Alinhamento de projeto	3h40			
23 de setembro	Documentação - pesquisas e formatação	4h20			
24 de setembro	Finalização da documentação geral	3h30			
25 de setembro	Planejamento de desenvolvimento	3h30			
28 de setembro	Pesquisa de documentação LOPD	4h			
02 de outubro	Produção de templates do software	6h40			
13 de outubro	Codificação da base do software	9h35			
15 de outubro	Codificação do aplicativo	4h30			
17 de outubro	Implementação da API do Maps	5h30			
19 de outubro	Finalização da documentação	3h			
20 de outubro	Documentação do software	8h30			
21 de outubro	Implementação do Firebase	5h			
25 de outubro	Chamada do firebase, mapa e banco de dados	8h			

(1) Horas atribuídas de acordo com o regulamento das Atividades Práticas Supervisionadas do curso.

TOTAL DE HORAS ATRIBUÍDAS: 85h e 10 min AVALIAÇÃO:

Aprovado ou Reprovado

NOTA:

DATA: / /



FICHA DAS ATIVIDADES PRÁTICAS SUPERVISIONADAS - APS

NOME: Alan da Silva Queiroz
CURSO: Ciência da Computação
DIGO DA ATIVIDADE: 53A6

TURMA: CC8P13
CAMPUS: Marquês
SEMESTRE: Oitavo

R.A.: N228GJ6
SEMESTRE: 8º Semestre
ANO GRADE: 2021

TURNO: Noturno

DATA DA ATIVIDADE	DESCRIÇÃO DA ATIVIDADE	TOTAL DE HORAS	ASSINATURA DO ALUNO	HORAS ATRIBUÍDAS (1)	ASSINATURA DO PROFESSOR
25 de agosto	Escolha do tema e divisão de tarefas	3h			
01 de setembro	Início das pesquisas teóricas	6h45			
12 de setembro	Pesquisas de conteúdo teórico	5h40			
22 de setembro	Alinhamento de projeto	3h40			
23 de setembro	Documentação - pesquisas e formatação	4h20			
24 de setembro	Finalização da documentação geral	3h30			
25 de setembro	Planejamento de desenvolvimento	3h30			
28 de setembro	Pesquisa de documentação LGPD	4h			
02 de outubro	Produção de templates do software	6h40			
13 de outubro	Codificação da base do software	9h35			
15 de outubro	Codificação do aplicativo	4h30			
17 de outubro	Implementação de API de Maps	5h30			
19 de outubro	Finalização da documentação	3h			
20 de outubro	Documentação do software	8h30			
21 de outubro	Implementação do Firebase	5h			
25 de outubro	Chamada do Firebase, mapa e banco de dados	8h			

(1) Horas atribuídas de acordo com o regulamento das Atividades Práticas Supervisionadas do curso.

TOTAL DE HORAS ATRIBUÍDAS: 85h e 10 min AVALIAÇÃO:

Aprovado ou Reprovado

NOTA: _____

DATA: ____/____/____

7.2 Ficha Alan Queiroz

7.3 Ficha Ana Simal



FICHA DAS ATIVIDADES PRÁTICAS SUPERVISIONADAS - APS

NOME: Ana Carolina Martins Simal
CURSO: Ciência da Computação
DIGO DA ATIVIDADE: 53A6

TURMA: CC8P13
CAMPUS: Marquês
SEMESTRE: Oitavo

R.A.: N2734E2
SEMESTRE: 8º Semestre
ANO GRADE: 2021

TURNO: Noturno

DATA DA ATIVIDADE	DESCRIÇÃO DA ATIVIDADE	TOTAL DE HORAS	ASSINATURA DO ALUNO	HORAS ATRIBUÍDAS (1)	ASSINATURA DO PROFESSOR
25 de agosto	Escolha do tema e divisão de tarefas	3h			
01 de setembro	Início das pesquisas teóricas	6h45			
12 de setembro	Pesquisas de conteúdo teórico	5h40			
22 de setembro	Alinhamento de projeto	3h40			
23 de setembro	Documentação - pesquisas e formatação	4h20			
24 de setembro	Finalização da documentação geral	3h30			
25 de setembro	Planejamento de desenvolvimento	3h30			
28 de setembro	Pesquisa de documentação LGPD	4h			
02 de outubro	Produção de templates do software	6h40			
13 de outubro	Codificação da base do software	9h35			
15 de outubro	Codificação do aplicativo	4h30			
17 de outubro	Implementação da API de Maps	5h30			
19 de outubro	Finalização da documentação	3h			
20 de outubro	Documentação do software	8h30			
21 de outubro	Implementação do Firebase	5h			
25 de outubro	Chamada do firebase, mapa e banco de dados	8h			

(1) Horas atribuídas de acordo com o regulamento das Atividades Práticas Supervisionadas do curso.

TOTAL DE HORAS ATRIBUÍDAS: 85h e 10 min AVALIAÇÃO:

Aprovado ou Reprovado

NOTA: _____

DATA: ____/____/____