

**UNIVERSIDADE DE SÃO PAULO**  
**ESCOLA DE ENGENHARIA DE LORENA**

**EDUARDA WILTINER REIS SANTANA**

*Silent Sprint: Uma plataforma web ágil bem guiada para realização de projetos de*  
**Internet das Coisas**

**Lorena**

**2021**

**EDUARDA WILTINER REIS SANTANA**

***Silent Sprint: Uma plataforma web ágil bem guiada para realização de projetos de  
Internet das Coisas***

**Trabalho de Graduação apresentado à  
Escola de Engenharia de Lorena da  
Universidade de São Paulo como requisito  
para obtenção do título de Engenheira  
Física**

**Orientador: Prof. Dr. Carlos Yujiro Shigue**

**Lorena  
2021**

**AUTORIZO A REPRODUÇÃO E DIVULGAÇÃO TOTAL OU PARCIAL DESTE TRABALHO, POR QUALQUER MEIO CONVENCIONAL OU ELETRÔNICO, PARA FINS DE ESTUDO E PESQUISA, DESDE QUE CITADA A FONTE.**

**Ficha Catalográfica**

Elaborada pela Biblioteca Especializada em Engenharia de Materiais  
EEL USP

Eduarda Wiltiner Reis Santana

*Silent Sprint: Uma plataforma web ágil bem guiada para realização de projetos de Internet das coisas.* / Eduarda Wiltiner Reis Santana; orientador Carlos Yujiro Shigue. --Lorena, 2021.

76 f.: il.

Monografia (Trabalho de Graduação em Engenharia Física) - Escola de Engenharia de Lorena - Universidade de São Paulo.

1. Internet das Coisas 2. Plataforma Web 3. Indústria 4.0 4. Empreendedorismo Tecnológico I.

*Silent Sprint: Uma plataforma web ágil bem guiada para realização de projetos IoT.*

Dedico este trabalho às mulheres que um dia acharam que tecnologia não é um “mundo” para elas e já sofreram de desmotivação e síndrome da impostora ao atuarem na área. A tecnologia evolui com mais diversidade e pode evoluir muito com mais mulheres.

## AGRADECIMENTOS

Agradeço aos meus pais, Paulo e Ana, que sempre me apoiaram e contribuíram para a minha formação e educação. Eles sempre foram pacientes nos momentos em que eu escolhia estar estudando e adiantando projetos ao invés de ficar o final de semana com eles. Eles são pessoas dedicadas e muito batalhadoras, uma coisa eles me ensinaram, de que esforço nunca é demais e que com ele vamos longe.

Agradeço a minha irmã, Paula, por estar sempre ao meu lado e me ajudando com todas as dificuldades. A opinião dela sempre foi muito importante para mim e não foi diferente para este Trabalho de Graduação. Eu aprendi com a minha irmã que desistir nunca é uma opção, por mais difícil que seja atingir um objetivo, não devemos parar de tentar e sempre aprender com os erros.

Agradeço à Escola de Engenharia de Lorena, à todos os funcionários que trabalham lá e a todos os professores que tive aula por me proporcionar muitos aprendizados durante o meu curso de Engenharia Física e por terem me ajudado a compreender mais sobre como a tecnologia funciona e como ela pode ajudar a humanidade resolvendo problemas através da ciência. Porém, agradeço mais ainda por me ensinarem a aprender e a desenvolver um pensamento crítico.

Agradeço à Trybe e a todos os profissionais que nela trabalham por ter me dado a chance de aprender muitas ferramentas tecnológicas das quais tenho certeza que vão me ajudar na minha carreira em tecnologia e me ajudaram muito a tornar esse trabalho num projeto que possa ser de grande utilidade para a sociedade na construção de projetos tecnológicos de Internet das Coisas.

Agradeço ao meu orientador Carlos Shigue por ter me ajudado neste Trabalho e por ter me orientado não só neste projeto final de graduação, mas em outros durante o curso de Engenharia Física, em que conquistei aprendizados decisivos que me fizeram seguir para uma carreira em tecnologia.

Finalizo os meus agradecimentos para todas as pessoas que de alguma forma contribuíram diretamente ou indiretamente na minha vida, formação e neste trabalho.

“A tecnologia não é nada. O que é importante é a nossa fé nas pessoas, que basicamente são boas e inteligentes, e se lhes dermos ferramentas, elas vão fazer coisas maravilhosas com elas.”

Steve Jobs

## RESUMO

SANTANA, E. W. R. ***Silent Sprint: Uma plataforma web ágil bem guiada para projetos de Internet das Coisas.*** 2021. 76 f. Monografia (Trabalho de Graduação em Engenharia Física) – Escola de Engenharia de Lorena, Universidade de São Paulo, Lorena, 2021.

A Internet das Coisas vem se tornando um assunto cada vez mais comentado e estudado, objetos inteligentes que se comunicam entre si e enviam dados do ambiente através da Internet, já estão tendo aplicações no Brasil e fato é que atualmente, empreender é muito difícil e com uso de tecnologias complexas como essa, o processo fica ainda mais desafiador. Após algumas tentativas que realizei para empreender com o uso da IoT(*Internet of Things*) durante o curso de Engenharia Física, as quais eu falhei, e estudando sobre uma visão de negócio focada na tecnologia IoT, percebi que o que havia de errado com a tentativa de se criar um produto inteligente competitivo junto a uma equipe de engenheiros da USP, era o uso de uma metodologia ágil generalista como o *Sprint*, que, por sua vez, não atendia a necessidade de se ter questionamentos voltados para explorar as reais vantagens de implementar a IoT para alavancar um negócio e garantir o sucesso da ideia. Por isso, o objetivo do presente trabalho foi construir uma plataforma web criada com uso de ferramentas tecnológicas modernas como, por exemplo, MongoDB, React, Node.js, Heroku e Vercel para servir como uma ferramenta para ser usada na realização de projetos de IoT através de um processo ágil criado especialmente para garantir o sucesso da ideia. A *Silent Sprint* nasceu para atender uma demanda crescente por tecnologias emergentes, especialmente a Internet das Coisas, através do fornecimento de uma metodologia dedicada e bem guiada para essa área com um total de seis etapas a serem seguidas. A plataforma já se encontra hospedada e pode ser acessada por qualquer usuário cadastrado no banco de dados que possua um notebook e conexão à internet. Após este trabalho, a *Silent Sprint* irá entrar em fase de teste para validação do processo e da experiência do usuário com a plataforma, aplicada, inicialmente, em ambiente universitário.

Palavras-chave: Internet das Coisas, processo ágil, React, Node.js, plataforma web.

## ABSTRACT

SANTANA, E. W. R. *Silent Sprint: A well-guided web platform for IoT projects*. 2021. 76 f. Monograph (Undergraduate dissertation in Engineering Physics) – Escola de Engenharia de Lorena, Universidade de São Paulo, Lorena, 2021.

The Internet of Things is becoming an increasingly discussed and studied subject, intelligent objects that communicate with each other and send data from the environment through the Internet, are already having applications in Brazil and the fact is that currently, entrepreneurship is very difficult and with use of complex technologies like this one makes the process even more challenging. After some attempts I made to undertake with the use of IoT (Internet of Things) during the Physical Engineering course, which I failed, and studying about a business vision focused on IoT technology, I realized that what was wrong with the attempt to create a competitive intelligent product with a team of engineers from USP was the use of a generalist agile methodology such as Sprint, which in turn did not meet the need to have questions aimed at exploring the real advantages of implementing IoT for leverage a business and ensure the idea success. Therefore, the objective of this work was to build a web platform created using modern technological tools such as MongoDB, React, Node.js, Heroku and Vercel to serve as a tool to be used in the realization of IoT projects through an agile process created especially to guarantee the success of the idea. Silent Sprint was born to meet a growing demand for emerging technologies, especially the Internet of Things, by providing a dedicated and well-guided methodology for this area with a total of six steps to be followed. The platform is already hosted and can be accessed by any user registered in the database who has a notebook and internet connection. After this work, Silent Sprint will enter the testing phase to validate the process and the user experience with the platform, initially applied in a university environment.

Keywords: Internet of Things, agile process, React, Node.js, web platform.

## LISTA DE FIGURAS

<b>Figura 1.</b> Comparativo do número de dispositivos conectados e o número de pessoas ao longo dos últimos anos com uma projeção feita para até 2020	20
<b>Figura 2.</b> Volume de pesquisas no Google sobre Rede de Sensores Sem Fio e Internet das Coisas	21
<b>Figura 3.</b> Tecnologias emergentes	22
<b>Figura 4.</b> Classificação e exemplos de sistemas de IoT	23
<b>Figura 5.</b> Representação das Revoluções Industriais	24
<b>Figura 6.</b> Estrutura da Indústria 4.0	25
<b>Figura 7.</b> Plataforma web de aprendizagem de construção de objetos inteligentes	27
<b>Figura 8.</b> Esquema dos passos seguidos para realização do projeto	28
<b>Figura 9.</b> Fluxograma do processo de realização de um projeto IoT	30
<b>Figura 10.</b> Arquitetura de três camadas	36
<b>Figura 11.</b> Divisão tecnológica através de uma visão de negócio	37
<b>Figura 12.</b> Etapa “O desafio” - Caso do ar-condicionado inteligente fictício	40
<b>Figura 13.</b> Etapa “2Pipelines” - Caso do ar-condicionado inteligente fictício	41
<b>Figura 14.</b> Etapa “Sketch” - Caso do ar-condicionado inteligente fictício	42
<b>Figura 15.</b> Esquema de funcionamento do hardware	44
<b>Figura 16.</b> Etapa “Protótipo” - Caso do ar-condicionado inteligente fictício	45
<b>Figura 17.</b> Etapa “Teste” - Caso do ar-condicionado inteligente fictício	46
<b>Figura 18.</b> Etapa “Starting” - Caso do ar-condicionado inteligente fictício	47
<b>Figura 19.</b> Versões do logotipo	48
<b>Figura 20.</b> Login na plataforma e Requisitos	49

<b>Figura 21.</b> Componente de progresso da página Home	50
<b>Figura 22.</b> Componente de trilha do projeto da página Home	51
<b>Figura 23.</b> Componente de recomendações da página Home	51
<b>Figura 24.</b> Página de detalhes	52
<b>Figura 25.</b> Conteúdo na plataforma	53
<b>Figura 26.</b> Sistema de coleta de feedback na plataforma	54
<b>Figura 27.</b> Página de Loading na plataforma	54
<b>Figura 28.</b> Erro 404 na plataforma	55
<b>Figura 29.</b> Etapas para a modelagem do banco de dados	56
<b>Figura 30.</b> Diagrama entidade-relacionamento	58
<b>Figura 31.</b> Serviço no Atlas de gerenciamento do banco de dados	60
<b>Figura 32.</b> Esquema de criação da API	61
<b>Figura 33.</b> Repositório da API Silent Sprint no GitHub	62
<b>Figura 34.</b> Repositório da API Silent Sprint no Heroku	65
<b>Figura 35.</b> Esquema de criação da plataforma Silent Sprint	66
<b>Figura 36.</b> Repositório da aplicação Silent Sprint no GitHub	67
<b>Figura 37.</b> Plataforma web Silent Sprint hospedado no Vercel	69
<b>Figura 38.</b> A página que aparece após o login do usuário	70



## LISTA DE TABELAS

**Tabela 1.** Crescimento e progresso da Internet

19

## LISTA DE QUADROS

**Quadro 1.** Criação de valor - Caso do ar-condicionado inteligente fictício

43

## LISTA DE SIGLAS

API	<i>Application Programming Interface</i> ou Interface de Programação de Aplicação.
AWS	<i>Amazon Web Service</i> ou Serviço Web da Amazon
CLI	<i>Command Line Interface</i> ou Interface de Linha de Comando
CORS	<i>Cross-Origin Resource Sharing</i> ou Compartilhamento de Recursos de Origem Cruzada
CSS	<i>Cascading Style Sheets</i> ou Folhas de Estilo em Cascatas
DBaaS	<i>Database-as-a-Service</i> ou Banco de Dados como um Serviço
EEL	Escola de Engenharia de Lorena
EF	Engenharia Física
GCP	<i>Google Cloud Platform</i> ou Plataforma na Nuvem do Google
HTML	<i>HyperText Markup Language</i> ou Linguagem de Marcação de Hipertexto
HTTP	<i>HyperText Transfer Protocol</i> ou Protocolo de Transferência de Hipertexto
IIoT	<i>Industrial IoT</i> ou IoT Industrial
IP	<i>Internet Protocol</i> ou Protocolo Internet
IoT	<i>Internet of Things</i> ou Internet das Coisas
JWT	<i>Json Web Token</i> ou Token Web em Json
NPM	<i>Node Package Manager</i> ou Gestor de Pacotes Node
NoSQL	<i>No Structured Query Language</i> ou Sem Linguagem de Consulta Estruturada.
PaaS	<i>Platform-as-a-Service</i> ou Plataforma como um Serviço
REST	<i>Representational State Transfer</i> ou Transferência de Estado Representacional
SGBD	Sistemas de Gerenciamento de Banco de Dados
SQL	<i>No Structured Query Language</i> ou Sem Linguagem de Consulta Estruturada
TG	Trabalho de graduação
UI	<i>User Interface</i> ou Interface do Usuário
USP	Universidade de São Paulo
UX	<i>User Experience</i> ou Experiência do Usuário
VSC	<i>Visual Studio Code</i> ou <i>VSCode</i>



## SUMÁRIO

1	INTRODUÇÃO	16
2	OBJETIVO	18
3	REVISÃO BIBLIOGRÁFICA	19
3.1	A história e definição da Internet das Coisas	19
3.2	A Indústria 4.0 e a Internet das Coisas	22
3.3	A Internet das Coisas no Brasil	24
3.4	O uso das plataformas digitais no aprendizado	25
4	MATERIAIS E MÉTODOS	28
4.1	A criação do processo Silent Sprint	29
4.1.1	O desafio	30
4.1.2	2Pipelines	32
4.1.3	Sketch	34
4.1.4	Protótipo	35
4.1.5	Teste	38
4.1.6	Starting	39
4.1.7	Caso de uso de um ar-condicionado inteligente	39
4.2	A construção da identidade visual	47
4.3	O design do Mockup	48
4.4	Modelagem do banco de dados	55
4.5	Criação e hospedagem da API	60
4.6	A programação e hospedagem da plataforma	65
6	RESULTADOS E DISCUSSÃO	70
7	CONCLUSÃO	71
	REFERÊNCIAS	72

## 1 INTRODUÇÃO

Bom, eu realizo projetos de tecnologia desde o meu primeiro ano de faculdade em 2017, projetos que envolvem tanto a parte de *hardware* quanto a parte de *software* e foi daí que nasceu o meu interesse pelo tema de Internet das Coisas, essa tecnologia une esses dois “universos” diferentes para gerar aplicações incríveis. A IoT nasceu através da evolução de outras tecnologias e começou a sua existência sendo um título de uma apresentação.

Com a utilização de pequenas redes interconectadas e aplicações dedicadas sobre a Internet, surge o conceito de computação em nuvem, que pode ser definido como um conjunto de recursos como capacidade de processamento, armazenamento, conectividade e serviços disponibilizados na Internet (TAURION, 2009). Nesse contexto, o Kevin Ashton cunhou o termo *Internet of Things* (IoT), em 1999, com base na ideia de “coisas” que usam a conexão via internet para serem rastreadas (STEVAN, 2018).

Em um artigo com mais de 5400 citações intitulado “*That ‘Internet of Things’ thing*”, Ashton conta como nasceu o termo IoT, em que esse termo começou sendo um título de uma apresentação para a empresa Procter & Gamble nos Estados Unidos, que tinha como o objetivo chamar a atenção para as vantagens do uso da tecnologia de Identificação via Radiofrequência (RFID) e se resumiu num *insight* importante que 10 anos depois iria se tornar um título de basicamente tudo, desde de um artigo na *Scientific American* até o nome de uma conferência da União Europeia (ASHTON, 2009).

Após o investimento de grandes empresas, como Amazon, Google, IBM, Microsoft e Cisco, acredita-se que a Internet das Coisas nasceu de fato entre 2008 e 2009, quando o número de dispositivos conectados à internet ultrapassou o número de pessoas conectadas. Esse momento foi constatado pelas medições estatísticas realizadas pelo Grupo de Soluções de Negócio de Internet da Cisco (*Cisco Internet Business Solutions Group*), que contabilizam a quantidade de dispositivos como smartphones, tablets e microcomputadores conectados à internet (STEVAN, 2018).

Cada vez mais estão sendo criados dispositivos conectados e inteligentes, é uma tendência mundial e pode trazer muitas vantagens competitivas para inúmeras empresas que procuram adotar um modelo de negócio de resultados possibilitado pelo uso da tecnologia IoT. Sabendo disso, ao longo do tempo, comecei a realizar mais projetos que envolviam o tema de Internet das Coisas, até que houve uma época onde eu pude realizar alguns ainda mais específicos para o tema de empreendedorismo com o uso dessa tecnologia.

Então, sim, eu tentei empreender com a IoT, porém após algumas tentativas e erros somados a algumas frustrações, juntamente com a equipe, decidi que o melhor seria me afastar da ideia de empreender por um tempo com o uso dessa tecnologia. Apenas com um tempo e estudando sobre como desenvolver uma visão sistêmica e de negócio para o uso da tecnologia IoT com o objetivo de empreender é que identifiquei que o que havia de errado era a metodologia ágil que estava sendo usada, antes. Não que a metodologia Sprint, criada por profissionais da Google e a metodologia *Scrum* empregada em larga escala no mercado atualmente, estejam erradas. Esses são ótimos exemplos de metodologias que quando

implementadas conseguem criar projetos muito bem desenvolvidos. Porém, como foram criadas para serem aplicadas de maneira geral em qualquer projeto, em trabalhos mais específicos e com o uso de tecnologias emergentes, como a Internet das Coisas, percebe-se que há uma certa carência de estratégias ou questões que possam beneficiar e facilitar o processo de criação de um verdadeiro produto IoT que consiga sobreviver ao mercado. Há a possibilidade de se adaptar alguma dessas metodologias ágeis e gerais existentes para serem aplicadas num projeto específico, mas isso pode ser desgastante e um tanto incerto quanto a eficácia da adaptação em relação ao que se pode criar com a IoT.

Foi a partir desse acontecimento que tive a ideia de criar um processo chamado *Silent Sprint* para resolver essa dor. Então, com base nas minhas experiências e analisando as tendências atuais, acredito que seja importante que se tenha um processo ágil dedicado à Internet das Coisas para auxiliar no desenvolvimento de projetos de IoT. E não somente isso, quando se tem um acompanhamento bem guiado e transparente, possibilitado por uma plataforma web que disponibiliza os conteúdos numa linguagem mais acessível de forma personalizada para o usuário e interativa torna o processo de criação de um produto IoT muito mais fácil e proveitoso. Essa plataforma será a *Silent Sprint* que deve se tornar uma ferramenta útil para o desenvolvimento de um projeto empreendedor e de Internet das Coisas.

## 2 OBJETIVO

O objetivo do presente trabalho de conclusão do curso de Engenharia Física é fornecer uma ferramenta para realização de um projeto focado no uso da tecnologia IoT que fornece um processo ágil e bem guiado. Uma ferramenta que possa ajudar as pessoas, que querem empreender com a IoT, a criar produtos tecnológicos incríveis que possam impactar a sociedade trazendo inovação, facilidades, resultados conectados e personalizados para as pessoas, e, ao mesmo tempo, incentivando o desenvolvimento de uma verdadeira Cultura Movida por Dados (*Data-driven culture*) no Brasil. Então, a *Silent Sprint* além de ser uma ferramenta para guiar projetos empreendedores ela possui um diferencial de ser dedicada a Internet das Coisas e com isso trazer questionamentos importantes para a validação de um produto IoT que algumas metodologias ágeis generalistas não trazem nas etapas de realização do projeto.

### 3 REVISÃO BIBLIOGRÁFICA

#### 3.1 A história e definição da Internet das Coisas

Atualmente, existem muitas definições sobre o que é a Internet das Coisas, uma delas é que ela é a chamada *Silent Intelligence* (Inteligência Silenciosa), pois ela conseguiu fazer parte da vida humana quase que instantaneamente (KELLMEREIT; OBODOVSKI, 2013). Para isso existem diversas suposições e explicações sobre isso, uma delas é que a Internet das Coisas representa um avanço de algo que já estava acontecendo.

O primeiro ponto que é essencial para a Internet das Coisas, sem o qual ela mesma não existiria, é a Internet, que começou a ser criada muito antes de ser inventado o termo IoT. Analisando os fatos históricos nos seus respectivos anos tem-se a Tabela 1, logo abaixo, que mostra o crescimento e progresso do mundo até chegar na tecnologia IoT, em 1999.

Tabela 1 - Crescimento e progresso da Internet

Ano	Descrição
1969	conceito da ARPANET
1974	protocolos TCP/IP
1977	ARPANET implementada com TCP/IP
1990	Internet
1991	Tim Berners-Lee inventou o WWW
1997	Wi-Fi
1999	IoT foi criada
1999	Primeiro dispositivo designado como RFID usado em IoT

Fonte: Adaptada de (KEERTIKUMAR; SHUBHAM; BANAKAR, 2015)

Os primeiros conceitos do que seria a Internet surgiu em 1957 durante a Guerra Fria entre os Estados Unidos e a União Soviética. Nessa época, os americanos criaram a Agência de Projetos Avançados de Defesa (*Advanced Research Projects Agency* - ARPA), que tinha o objetivo de desenvolver uma rede de computadores capaz de trocar informações por meio de tecnologias de comunicação para evitar a perda de informações, caso parte da rede física ficasse inativa devido a destruição parcial, provocada, por exemplo, por um ataque nuclear. Surgiu, então, a primeira versão da rede, a Rede de Agências de Projetos de Pesquisa Avançada (*Advanced Research Projects Agency Network* - ARPANET) em 1969. Embora o

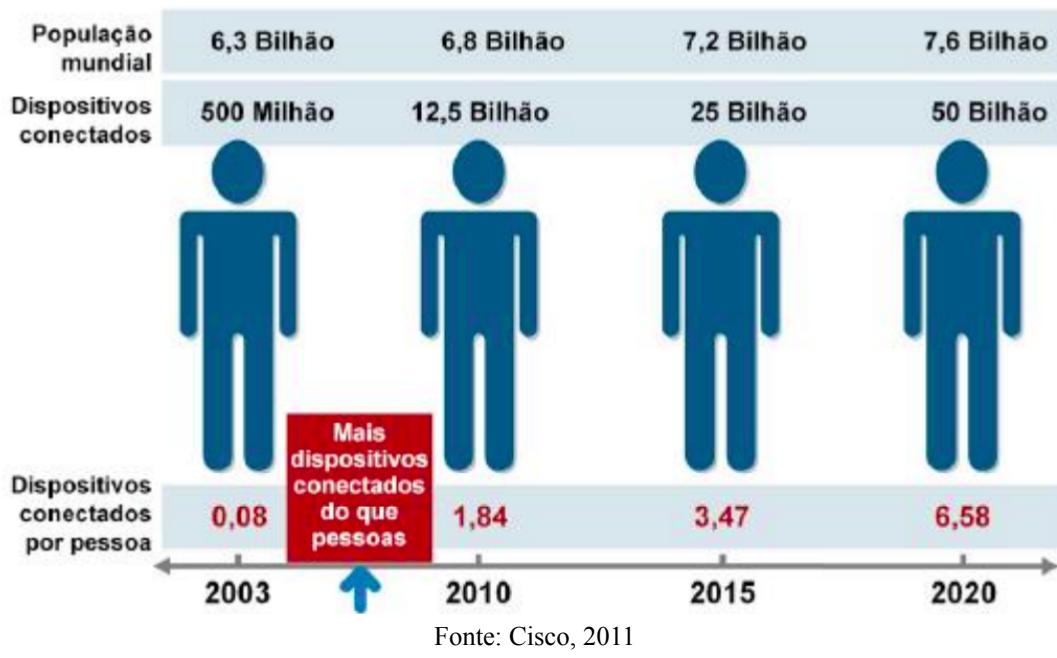
conceito de ARPANET estivesse evoluindo, demorou cerca de 8 anos para implementar o conceito comercialmente (KEERTIKUMAR; SHUBHAM; BANAKAR, 2015).

O protocolo TCP/IP (Protocolo de Comunicação e Transmissão/Protocolo de Internet) foi desenvolvido em 1974, sendo desenvolvido o pacote Arpanet com a comutação de protocolos pelo Robert E. Kahn e Vint Cerf no mesmo ano, teve seu primeiro uso comercial na ARPANET em 1974. Tim Berners-Lee no ano de 1991 introduziu a Rede Mundial de Computadores (*World Wide Web - WWW*), que é recebida como um avanço em serviços de Internet. Assim que o Wi-Fi foi introduzido em 1997, muitas aplicações começaram a surgir usando a internet. Em 1999, quando a palavra IoT foi cunhada, o primeiro dispositivo foi projetado usando o modo de comunicação Identificação via radiofrequência (RFID), que tinha um alcance de 10 cm - 200 m (KEERTIKUMAR; SHUBHAM; BANAKAR, 2015).

Equipamentos elétricos e eletrônicos ("Coisas") controlados remotamente são tecnologias bem conhecidas desde o início de 1990, quase 10 anos antes de ter sido criado o termo Internet das Coisas (SURESH, 2014). Com esses conceitos e estudos de tecnologias sendo controladas remotamente, o avanço de *smartphones*, provocando a criação e a inovação de diversos sensores e a computação em nuvem ficando cada vez mais popular, o que estava faltando era conectar tudo e integrar sistemas e ecossistemas compostos de dispositivos de diferentes aplicações para juntos entregar resultados completos a sociedade, e dessa necessidade nasceu a Internet das Coisas.

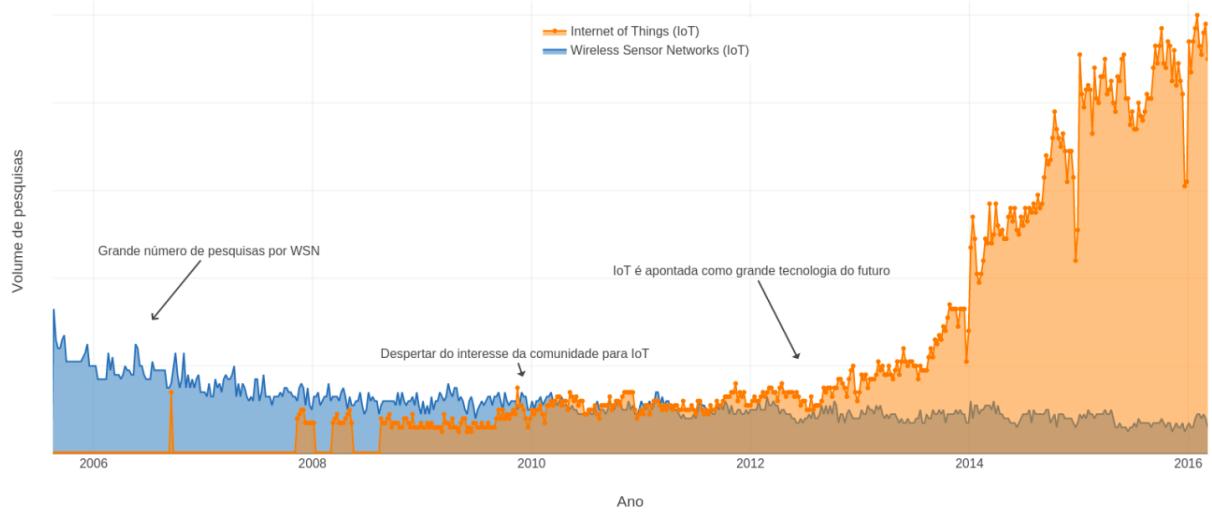
Hoje, temos uma quantidade de dispositivos conectados muito maior que a população mundial. Após o surgimento do conceito da Internet das Coisas em 1999, em 2015 foi constatado que a quantidade de dispositivos conectados era cerca de três vezes a população mundial, ou seja, cerca de 25 bilhões (STEVAN, 2018). A comparação da quantidade de dispositivos com o crescimento populacional pode ser vista na Figura 1, logo abaixo.

Figura 1 - Comparativo do número de dispositivos conectados e o número de pessoas ao longo dos últimos anos com uma projeção feita para até 2020.



Realizando uma breve pesquisa no Google Académico de *Internet of Things*, atualmente, consegue-se encontrar mais de 3 milhões de artigos e livros publicados sobre o tema, alguns com mais de 2000 citações. Então é inegável que o tema vem tendo cada vez mais importância na sociedade. Um estudo interessante afirma que por volta de 2005, o termo bastante procurado, tanto pelo meio acadêmico quanto pela indústria, e que apresenta relação com a IoT foi Redes de Sensores Sem Fio (RSSF) (SANTOS; SILVA; CELES; BORGES; NETO; VIEIRA; GOUSSEVSKAIA; LOUREIRO, 2016).

Figura 2 - Volume de pesquisas no Google sobre Rede de Sensores Sem Fio e Internet das Coisas



Fonte: (SANTOS; SILVA; CELES; BORGES; NETO; VIEIRA; GOUSSEVSKAIA; LOUREIRO, 2016)

A mesma pesquisa informa que nos anos seguintes (entre 2008 e 2010), o termo Internet das Coisas ganhou uma popularidade surpreendente. Isto se deve ao amadurecimento e a evolução das RSSFs e ao crescimento das expectativas de sucesso sobre a tecnologia IoT. Em 2010, as buscas para IoT dispararam, chegando a ultrapassar as pesquisas sobre RSSFs, como pode-se observar na Figura 2, logo abaixo.

Figura 3 - Tecnologias emergentes

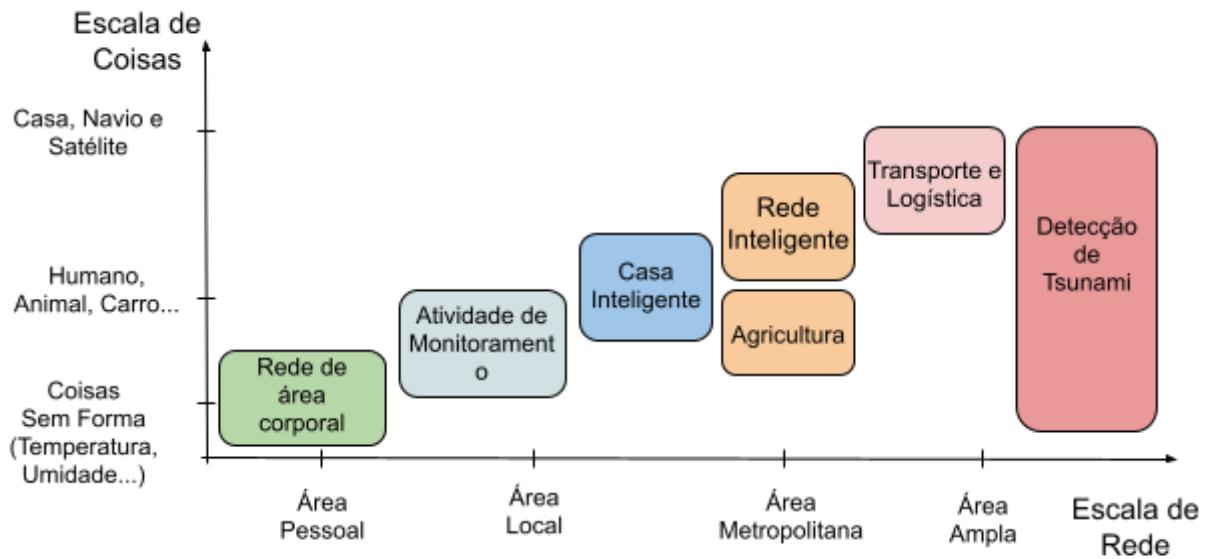


Fonte: (SANTOS; SILVA; CELES; BORGES; NETO; VIEIRA; GOUSSEVSKAIA; LOUREIRO, 2016)

Por fim, uma das definições mais aceitas, atualmente para a Internet das Coisas é que ela pode ser vista como uma representante de qualquer dispositivo ligado à internet que pode ser identificado através de endereçamentos universais (chamados de endereços IP, que identificam as máquinas disponíveis para conexão) e com isso qualquer “coisa” que possa prover informações desse ambiente e possa se comunicar com qualquer outra “coisa” que nela esteja conectada (STEVAN, 2018).

Sistemas de IoT como Casas Inteligentes, Cidades Inteligentes, Agricultura Automatizada e Monitoramento de Atividades são exemplos de aplicações que estão sendo já utilizadas, na atual conjuntura mundial. É interessante analisar a escala de *network* que cada um desses sistemas pertence, ou seja, observar qual a área que a rede de tráfego de dados deverá englobar para atender a essas aplicações de Internet das Coisas, como por exemplo, o sistema de aplicação de uma casa inteligente deve necessitar, numa escala de rede, de uma cobertura entre o tamanho de uma área local e metropolitana, aproximadamente. Isso demonstra o quanto a Internet das Coisas possui efeitos diretos nos lugares em que as aplicações inteligentes são implementadas, carregando inovação e facilidades para a vida humana. Logo abaixo, na Figura 4, há a classificação de sistemas de IoT de acordo com as suas respectivas escalas da rede e escalas de “coisas”, que podem ser interpretadas como as categorias dos sistemas exemplificados (KAWAMOTO; HIROKI; NEI; NAOKO; SHINICHI, 2014).

Figura 4 - Classificação e exemplos de sistemas de IoT



Fonte: Adaptada de (KAWAMOTO; HIROKI; NEI; NAOKO; SHINICHI, 2014)

### 3.2 A Indústria 4.0 e a Internet das Coisas

O conceito de Indústria 4.0 foi discutido pela primeira vez durante a feira de Hannover na Alemanha e começou a ser implantado a partir de 2013. Um projeto do governo alemão voltado para novas estratégias que aliam tecnologia e meios de produção que começou a introduzir a Indústria 4.0 no país. O fundamento básico da Indústria 4.0 diz que as fábricas que conectam máquinas e sistemas possuem a capacidade e autonomia para controlar e agendar manutenções, prever falhas em processos através de análises preditivas e se adaptar a mudanças inesperadas que ocorrem nas etapas de produção através do monitoramento constante. Isso implica em uma nova era no âmbito das revoluções industriais (FREITAS, 2017).

Historicamente a indústria faz parte da economia que produz bens materiais que são altamente mecanizados e automatizados com a evolução da tecnologia ao longo do tempo. Desde o início da industrialização, saltos tecnológicos levaram a mudanças de paradigma que hoje são chamados de “revoluções industriais” (LASI; FETTKE; KEMPER; FELD; HOFFMANN, 2014). A Figura 5 abaixo, ilustra as características e marcos de cada revolução.

Figura 5 - Representação das Revoluções Industriais



Fonte: (FREITAS, 2017)

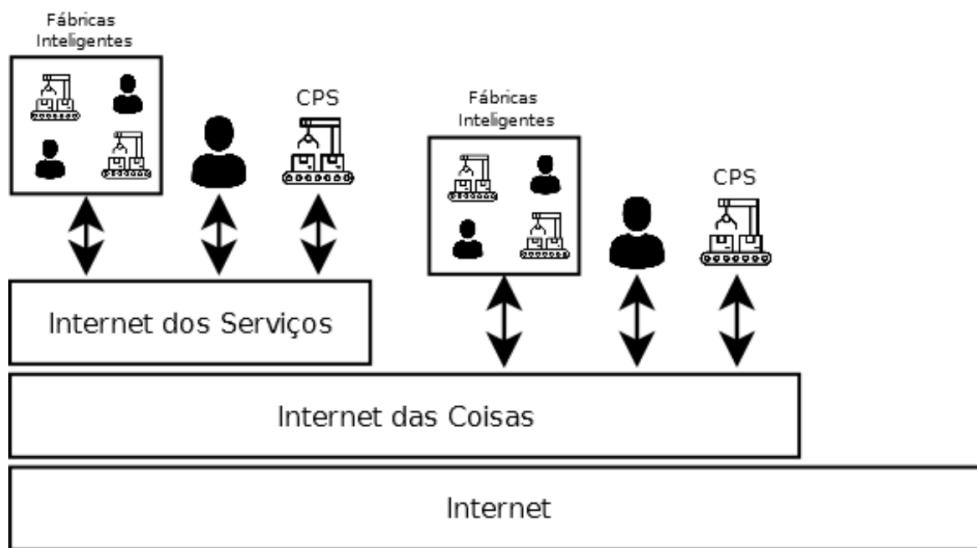
Após o conceito formado pelo governo alemão ter se tornado popular, a Quarta Revolução Industrial vem sendo chamada de Indústria 4.0 e, pela primeira vez, não está sendo estudada após ocorrer como aconteceu com as revoluções anteriores a esta. Trata-se de uma revolução dos processos de manufatura, tendo por base, entre outras tecnologias, os Sistemas Ciber Físico (CPS), são equipamentos com capacidade de integrar seu corpo físico ao mundo virtual e de representar seu estado a partir da coleta de informações em tempo real, podendo, desta forma, tomar decisões autônomas, e a Internet das Coisas, como já comentada, faz uso da disponibilidade cada vez maior de infraestrutura de rede e transmissão, conectando os mais diversos objetos de nosso cotidiano, como, por exemplo, os próprios CPS. A partir da Indústria 4.0, essas maiores complexidades poderão ser tratadas em processos produtivos, obtendo produtos personalizados a preços competitivos interferindo diretamente na economia do país que a aplica e a desenvolve (PEREIRA; SIMONETTO, 2018).

A IoT é de extrema importância na atual revolução industrial, ela pode ser vista como a principal ponte entre as aplicações físicas e digitais originadas pela quarta revolução industrial. Atualmente os sensores e outros meios de conectar objetos às redes virtuais estão se espalhando em ritmo acelerado, isso se deve principalmente a popularização do *smartphone*. Os sensores são menores, mais baratos e inteligentes. Então, é fácil poder imaginar casas, roupas, acessórios, cidades, redes de transporte e energia, e indústria tendo sensores acoplados para fornecer dados preciosos de acordo com o objetivo da aplicação de Internet das Coisas.

A expectativa, como já comentada, é que o número desses dispositivos que fazem tudo acontecer, aumente drasticamente com o passar dos anos, podendo chegar a mais de um trilhão de dispositivos. Isso pode alterar a maneira que se gerencia as cadeias de fornecimento, pois permitirá monitorar e otimizar as várias atividades de forma precisa por

meio do sensoriamento e análise de dados em tempo real. É certo que todos os setores serão impactados. Pode-se pensar no monitoramento remoto, uma aplicação generalizada da IoT. Por exemplo, qualquer pacote, palete ou contêiner, transportados de um ponto a outro, pode ter uma etiqueta RFID, permitindo serem rastreados pelas empresas que saberão a localização de cada objeto a qualquer momento. Os clientes poderão acompanhar seus pedidos de forma contínua, praticamente em tempo real (FREITAS, 2017). Assim pode-se esquematizar a estrutura da Indústria 4.0 como mostrada na Figura 6.

Figura 6 - Estrutura da Indústria 4.0



Fonte: (PEREIRA; SIMONETTO, 2018)

### 3.3 A Internet das Coisas no Brasil

Com o avanço da tecnologia, existem muitas “coisas” conectadas à internet, com capacidade para compartilhar, processar, armazenar e analisar um volume enorme de dados entre si, realizando o chamado *Big Data*, o imenso volume de dados provenientes de diversas fontes que trafegam entre máquinas através da internet. No âmbito nacional a implementação da Internet das Coisas vem sendo cada vez mais incentivada. No Brasil, o Governo Federal instituiu o Plano Nacional de Internet das Coisas, Decreto nº 9.854, de 25 de junho de 2019. De acordo com o art. 1º, tem o objetivo “de implementar e desenvolver a Internet das Coisas no País e, com base na livre concorrência e na livre circulação de dados, observadas as diretrizes de segurança da informação e de proteção de dados pessoais”.

Segundo o Plano Nacional de Internet das Coisas, foi considerado que a Internet das Coisas é “a infraestrutura que integra a prestação de serviços de valor adicionado com capacidade de conexão física ou virtual de coisas com dispositivos baseados em tecnologias da informação e comunicação existentes e nas suas evoluções, com interoperabilidade” (BRASIL, 2019). Isso pode ser considerado um grande passo para o Brasil, pois uma pesquisa realizada para saber o cenário da implementação de projetos de IoT em governos em todo o mundo, mais notáveis como EUA, Europa e a China revelou que estes colocaram a IoT como uma das principais prioridades de sua agenda de pesquisa, influenciando diretamente na criação de novas aplicações tecnológicas (GAZIS, 2015).

Existem muitos estudos sendo realizados no Brasil, inclusive na área de medicina, que mostram o impacto e os benefícios que a Internet das Coisas pode trazer. Um estudo sobre *big data* em saúde no Brasil, afirma que “a quantidade de dados gerados pela internet das coisas será imensamente útil aos epidemiologistas, já que permitirá identificar todos os passos imediatos e distantes que levaram ao aparecimento das doenças ou ao óbito” (CHIAVEGATTO, 2015). Um conjunto de sensores em sistemas embarcados podem estar realizando coleta de dados em tempo real de pacientes e enviando esses dados via internet para os respectivos médicos e pesquisadores responsáveis pelo acompanhamento de pacientes, permitindo identificar com mais precisão os efeitos de uma doença e seus perigos com mais assertividade e rapidez.

Para o apoio e representatividade de âmbito nacional e assim, permitir a atuação brasileira em todas as frentes do setor de Internet das Coisas foi criada a Associação Brasileira de Internet das Coisas (ABINC) em 2015. Uma organização sem fins lucrativos, formada por executivos e empreendedores do mercado de TI e Telecom. Essa associação tem como objetivo incentivar a troca de informações e fomentar a atividade comercial entre associados, para mover o mercado de Internet das Coisas e promover diversas atividades de pesquisa e desenvolvimento, atuando junto às autoridades governamentais envolvidas e fazendo as parcerias internacionais com entidades do setor (ABINC). Possuir essa associação motiva o Brasil a investir e a desenvolver cada vez mais projetos de IoT, o que pode trazer uma evolução na economia, na qualidade de vida dos brasileiros, no posicionamento mundial de desenvolvimento tecnológico do país dentre outros.

Contudo, ainda há barreiras para se vencer para conseguir desenvolver grandes aplicações de IoT no Brasil. Cerca de 50% dos lares não têm acesso à internet, de acordo com a pesquisa TIC Domicílio (TIC DOMICÍLIO). Os números têm diminuído gradualmente, mas a democratização do acesso à internet é um passo necessário para se alcançar uma rede verdadeiramente nacional. Mas isso não impede as transformações promissoras que as tecnologias digitais podem trazer, principalmente a Internet das Coisas (MAGRANI, 2018).

### 3.4 O uso das plataformas digitais no aprendizado

Numa pesquisa realizada em Taiwan, na China, com um total de 116 alunos de 4 classes diferentes, foram projetadas atividades de ensino para uma aprendizagem em plataformas digitais, os resultados da pesquisa foram as seguintes:

1. A aprendizagem digital apresenta melhores efeitos positivos na motivação de aprendizagem do que o ensino tradicional;
2. A aprendizagem digital mostra melhores efeitos positivos no resultado da aprendizagem do que o ensino tradicional;
3. A motivação de aprendizagem revela efeitos significativamente positivos sobre o efeito de aprendizagem na aprendizagem resultado;
4. A motivação de aprendizagem parece ter efeitos notavelmente positivos sobre o ganho de aprendizagem no resultado de aprendizagem;

Os resultados demonstram que o aprendizado nas plataformas digitais provoca uma motivação maior nos alunos em aprender do que em um ensino tradicional, e que, consequentemente, gera efeitos positivos na aprendizagem (LIN; CHEN; LIU, 2017). As plataformas web de ensino vêm ficando cada vez mais conhecidas e utilizadas inclusive por conta da atual Pandemia. Muitas instituições e até empresas estão migrando para a modalidade dupla de trabalho, presencial e remoto, e no desenvolvimento de atividades e projetos, as plataformas web têm sido as principais ferramentas utilizadas para dar o suporte de acompanhamento, comunicação, compartilhamento de conteúdo e orientação.

No Brasil, foi realizada uma pesquisa, em que se obteve o impulso registrado pela Educação a Distância (EAD) junto ao desenvolvimento de novas Tecnologias de Informação e Comunicação (TIC) nos últimos anos, gerou um aperfeiçoamento tecnológico dos ambientes virtuais de aprendizagem, ou seja, nas plataformas web de ensino. O fato é que essa necessidade de se aperfeiçoar o uso de novas ferramentas tecnológicas, propõe uma nova visão pedagógica à modalidade, que é o de contribuir para o aprendizado autônomo do aluno, porém de forma mais colaborativa e ativa (GABARDO, 2010).

Então, criar conteúdo e processos para serem seguidos e aprendidos não apenas em textos, mas também de forma interativa e mais motivacional em plataformas digitais de qualidade e responsivas, demonstra ser uma boa estratégia para a qualidade de ensino de diversos assuntos, inclusive no ensino de desenvolvimento de projetos de tecnologia, o que, também, incentiva a capacitação e o interesse em assuntos de base tecnológica para a comunidade. Um exemplo de uma plataforma digital de ensino e muito conhecida desenvolvida para o aprendizado de construção de objetos inteligentes, investida pela empresa Samsung, é a Code IoT, que disponibiliza conteúdos técnicos sobre como criar um sistema embarcado de Internet das Coisas, e que está representada na Figura 7, logo abaixo.

Figura 7 - Plataforma web de aprendizagem de construção de objetos inteligentes - Code IoT

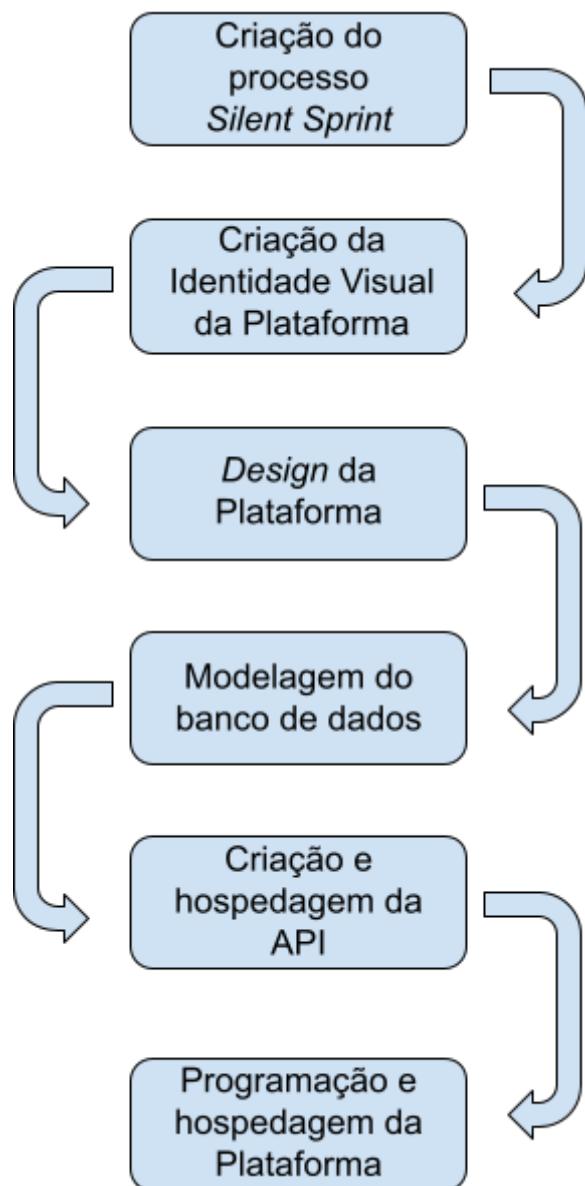


Fonte: (CODEIOT)

#### 4 MATERIAIS E MÉTODOS

A realização do trabalho foi dividida em seis partes: a criação do processo ágil dedicado à realização de um projeto IoT, a construção da identidade visual da plataforma web que vai armazenar o conteúdo do processo, o *design* da plataforma, a modelagem do banco de dados para armazenar dados dos usuários, a criação e hospedagem da Interface de Programação de Aplicação (API) que vai conectar o banco de dados à plataforma web e, por fim, a programação e hospedagem da plataforma. Abaixo, na Figura 8, encontra-se um esquema dos passos.

Figura 8 - Esquema dos passos seguidos para realização do projeto



Fonte: autoria própria

#### 4.1 A criação do processo *Silent Sprint*

Antes de se criar a estrutura da plataforma web que vai servir para capacitar e guiar as pessoas para realizarem projetos incríveis de tecnologia, deve-se criar toda a parte de conteúdo primeiro, pois essa é a parte mais importante de quando se trata de disponibilizar uma ferramenta útil para a sociedade. Se os passos a serem seguidos durante o processo não estiverem bem elaborados e claros para quem estiver lendo e seguindo, não vai causar o efeito que se busca com a criação de um processo ágil e bem guiado.

As metodologias ágeis seguem o estilo iterativo e incremental de desenvolvimento de projetos que se ajusta de forma dinâmica às mudanças e permite um melhor gerenciamento de risco. Os quatro princípios básicos de ágil conforme definido pelo Manifesto Ágil são (HODA; NOBLE; MARSHALL, 2008, p.219):

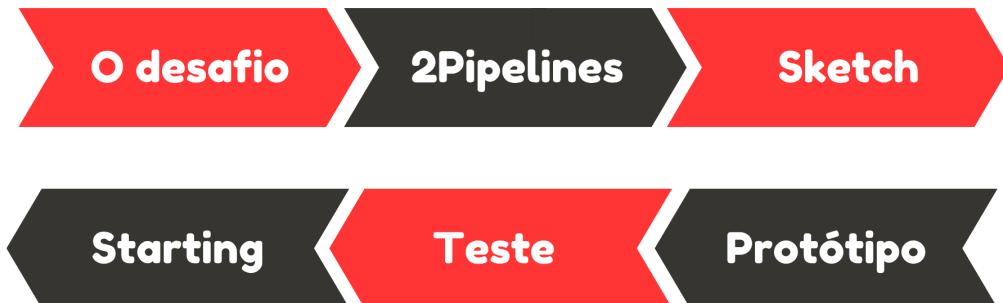
- Indivíduos e a interação entre eles, mais que processos e ferramentas;
- *Software* em funcionamento, mais que documentação abrangente;
- Colaboração com o cliente, mais que negociação de contratos;
- Responder a mudanças, mais que seguir um plano;

De acordo com o Manifesto Ágil, mesmo havendo valor nos itens à direita (processos e ferramentas, documentação abrangente, negociação de contratos e plano), os itens à esquerda (indivíduos e interações, *software* em funcionamento, colaboração e resposta a mudanças) são mais valorizados. Isso combina com o modelo de negócios impulsionado pela Internet das Coisas, um modelo de negócio de resultados que tem como principal foco o cliente.

O processo de realização do projeto de IoT foi criado para ser implementado como uma metodologia ágil, inspirado em sua grande parte no livro “Sprint” (KNAP; ZARETSKY; KOWITZ, 2017) escrito por profissionais da empresa Google Ventures para realização de projetos empreendedores e no livro “Como usar a Internet das Coisas para alavancar o seu negócio?” do autor Bruce Sinclair (SINCLAIR, 2018) que fornece alguns questionamentos importantes sobre como criar um produto de IoT de sucesso e competidor.

O processo foi dividido em seis partes fundamentais, como pode-se perceber na Figura 9, logo abaixo, para a realização de um projeto empreendedor tradicional, desde a criação da ideia até o primeiro contato com os investidores (*stakeholders*), mas adaptado para o uso de uma tecnologia emergente, no caso a Internet das Coisas, para sanar algum problema existente na sociedade.

Figura 9. Fluxograma do processo de realização de um projeto IoT



Fonte: autoria própria

Para que o processo não seja tão cansativo e consiga captar a atenção do usuário, que vai estar consumindo o conteúdo através da plataforma, por um breve tempo, foram criados alguns passos em cada uma das etapas criadas. Isso vai ajudar a dividir o conteúdo de forma que o usuário consiga ter uma leitura fluida e ainda assim possa conectar cada parte do conteúdo lido, tendo ao final de cada passo sempre atividades práticas a serem realizadas.

Diferente do “*Design Sprint*” da Google Ventures não haverá um tempo determinado para realização das seis etapas e elas não precisam ser feitas uma em cada dia, muito menos tudo em uma semana, o usuário vai poder adaptar a sua gestão de tempo do seu projeto conforme a sua rotina sendo guiado pela plataforma durante o desenvolvimento. A seguir cada etapa será explicada com mais detalhes.

- **4.1.1 O desafio**

Esta primeira etapa serve para entrar numa linha de pesquisa que possibilite ter mais segurança sobre os passos seguintes do projeto. Alguns projetos podem terminar apenas nessa primeira etapa, fazendo com que as pessoas economizem tempo e energia para validar outras ideias mais promissoras. Por isso o nome dessa etapa é “O desafio”.

O desafio possui ao todo três passos que englobam métodos e questionamentos importantes para criação e validação de uma ideia.

#### **Passo 1 - A criação de uma ideia**

Se o usuário não tiver uma ideia, ele terá a possibilidade de criá-la através de um dos dois métodos chamados Funil de Ideias “A partir da sua vivência” e Funil de Ideias “A partir do mercado” (NUCCIO; WOLLHEIM, 2016):

O Funil de Ideias “A partir da sua vivência” nada mais é do que um processo ou linha de pensamento baseado na vivência, em que engloba algumas perguntas que permitem encontrar um negócio alinhado com o perfil empreendedor de quem a aplica. Já um Funil de Ideias “A partir do mercado” é um método de geração de ideias que serve para avaliar ideias existentes que estejam em outras localidades, que possuam uma deficiência ou uma tendência, e passar por uma série de perguntas que vai ajudar a filtrá-las até chegar na grande ideia.

## Passo 2 - Validação da ideia

A validação da ideia é também uma parte fundamental no início de um projeto e para isso é necessário:

1. Verificar se a ideia realmente soluciona alguma dor na sociedade e se é possível que a ideia seja resolvida com uso da IoT, já que se trata de um projeto de Internet das Coisas. Com isso, é importante saber que um objeto inteligente é normalmente composto por quatro unidades básicas: processamento de memória, comunicação, geralmente feita via Internet, energia para alimentar o dispositivo, sensores e/ou atuadores.
2. Identificar qual o setor ou área da solução de IoT, pois quando se trabalha com Internet das Coisas através de uma visão de negócio, é necessário saber qual a categoria do produto de IoT a ser criado.

Então, o usuário aprenderá as quatro categorias de IoT numa visão de negócio. É importante se tratar de uma categoria com uma divisão de negócio e não de uma divisão técnica da IoT já que se trata de um projeto empreendedor e essa visão também deve ajudar nas próximas etapas do processo. Logo, as categorias são (SINCLAIR, 2018):

- **IoT de Consumo:** São produtos inteligentes de consumo pessoal que possuem dispositivos conectados entre si, realizando medidas do ambiente em que estão inseridos. Exemplos: casas inteligentes com produtos individuais conectados como escova de dentes, eletrodomésticos e camas, carros inteligentes que estão se tornando cada vez mais autônomos e dispositivos de consumo gerais.
- **IoT Comercial:** São serviços oferecidos pela aplicação da Internet das Coisas em transportes, assistência médica e seguradoras como por exemplo. Em geral, os equipamentos, nessa categoria, estão sendo instrumentados para transmitir dados aos proprietários e usuários, a fim de melhorar o negócio e as relações com os clientes.
- **IoT Industrial:** A Internet das Coisas cria condições para que os fabricantes façam “produtos melhores” e lancem no mercado mais máquinas e mais produtos a custo baixo. Nos setores de petróleo e mineração analisam dados coletados por sensores para se tornarem mais eficientes em seus processos.
- **IoT de Infraestrutura:** Com o avanço das cidades inteligentes que conectam os equipamentos urbanos com habitantes e com os meios de transporte, a infraestrutura está ficando cada vez mais inteligente. Empresas de serviços de utilidade pública estão distribuindo eletricidade com mais eficiência e com mais credibilidade utilizando IoT, por exemplo.

### **Passo 3 - Alguém ou empresa já solucionou o problema?**

Verificar se alguém ou alguma empresa já solucionou o problema encontrado no início do processo é uma forma de tentar garantir que o produto de IoT desenvolvido irá sobreviver ao mercado e que o esforço para tanto será justificado, trazendo mais confiança no projeto durante todo o processo.

Bom, se o problema ainda não foi resolvido, pode significar que o esforço para tanto não compense economicamente, ou a tecnologia atual não possibilita uma solução viável para resolver a dor, ou o estado atual é resistente a mudanças. Para esses três exemplos existem soluções que, se bem desenhadas, podem levar ao sucesso (TOYOFUKU). As soluções vêm através do estudo da concorrência.

Então, é importante que o usuário estude a concorrência que vai enfrentar na tentativa de implementação da sua ideia no mercado. Existem três categorias de concorrentes que o usuário vai poder analisar, que aparecem quando se trabalha com Internet das Coisas (SINCLAIR, 2018):

- **Concorrentes Nativos:** São concorrentes que possuem um negócio parecido com o da empresa que está lançando o produto IoT, porém eles não possuem vantagens competitivas de IoT intrínsecas.
- **Concorrentes Exóticos:** São considerados como futuros concorrentes de ecossistemas. São empresas de outros setores que normalmente não são consideradas competitivas.
- **Novos Entrantes:** São os concorrentes mais difíceis de se prever. São concorrentes que apesar da pouca experiência com toda a parte técnica de IoT podem ter uma grande ideia de um produto que possa ser muito competitivo.

Neste passo, então, a pessoa, que desenvolve o projeto de IoT, deverá descobrir e mapear quais concorrentes existem para acompanhar a evolução de comportamentos competitivos e tomadas de decisão. Nessa pesquisa o usuário será orientado a ficar atento para possíveis parcerias com outras empresas e a possibilidade de criar uma rede de conexão entre diferentes produtos IoT para gerar valor incremental.

É interessante, neste passo, ajudar o usuário, que consome o conteúdo na plataforma web, a enxergar que com a ampla quantidade de dispositivos conectados coletando uma grande quantidade de dados por segundo, mesmo a Internet das Coisas atuando nos bastidores da aplicação, ela ainda consegue, através de produtos inteligentes diferentes trabalhando em conjunto, entregar o resultado que o cliente deseja, até de maneira personalizada e muitas vezes, atuando antecipadamente através de uma análise preventiva e prescritiva de dados para corrigir erros. Isso se trata do deslocamento de importantes tendências, como os negócios de economia baseados em produtos e serviços, para uma Economia de Resultado (*Outcome Economy*) (SINCLAIR, 2018). Isso quer dizer que alguns concorrentes podem ser, na verdade, parceiros que possam ajudar na entrega do resultado esperado pelo cliente.

- **4.1.2 2*Pipelines***

Construir *pipelines*, nada mais é do que criar uma sequência de passos bem definidos e conectados entre si posicionados numa linha horizontal para conseguir uma visão do processo inteiro do empreendimento de Internet das Coisas e assim conseguir atuar de maneira estratégica para que cada etapa funcione da melhor forma e evite possíveis erros.

Nos dois primeiros passos o usuário irá aprender conceitos e coletar informações que o ajudará na montagem de dois *pipelines* importantes no processo de desenvolvimento do projeto.

### **Passo 1 - A descoberta do cliente**

Aqui é importante entender quais pessoas podem ser os possíveis clientes e entender o quanto importante é o problema a ser solucionado na vida deles de forma rápida e eficiente usando as 5 perguntas elaboradas pelo John Wilcox em sua “Descoberta do Cliente” .

O método de descoberta do cliente criado pelo John Wilcox é um processo fácil e prático, em que o objetivo é identificar a viabilidade da ideia do projeto e descobrir se realmente existe um grupo de clientes que comprariam o produto. Para isso é necessário realizar entrevistas com o uso das seguintes perguntas em sequência (WILCOX, 2017).

1. “Me conta uma história sobre a última vez que você teve problemas em...(problema a ser resolvida pela IoT)”
2. “O que foi mais difícil?”
3. “Por que foi difícil?”
4. “Como você resolve esse problema agora?”
5. “Por que isso não é incrível?”

Com essas perguntas o entrevistador vai identificar se a solução para o problema resolvido pela ideia criada já é mitigada pelas soluções já existentes encontradas pelo possível cliente, se ela realmente gera um ótimo resultado para o cliente a ponto de não ter oportunidade de competir com essa solução no mercado, se realmente vale a pena investir tempo e estudo na realização do projeto baseado nessa ideia ou se seria interessante *pivatar*, ou seja, voltar todo o processo para o início.

### **Passo 2 - Modelagem de Valor**

Neste passo o usuário irá identificar o que o produto de IoT irá fazer e qual o impacto no mercado ele terá, isso vai ajudar a construir o *pipeline* de vendas e de resultados, pois o seu objetivo já estará claro. Para isso existem quatro formas de gerar valor que vão ajudar a determinar isso (SINCLAIR, 2018):

- **Tornar produtos melhores:** cria valor por meio da inovação. A maneira mais eficaz de fazer os “produtos melhores” é por meio da invenção e inovação. Uma maneira nova, melhor e mais eficiente de fazer alguma coisa que os clientes consideram valiosa é o que torna ótimo os produtos, e, no processo, vence a competição, conquistar mercados e contribui gerando lucro para a empresa.
- **Operar melhor os produtos:** cria valor por meio do aumento da eficiência operacional. Quando se trabalha com a IoT, aumentar a eficiência operacional é um dos principais benefícios da IoT amplamente reconhecido no mundo, atualmente. Esse é, em geral, o melhor ponto de partida para o desenvolvimento de um produto IoT segundo o Bruce Sinclair.
- **Suporrtar melhor os produtos:** cria valor por meio do aumento da utilização do ativo. Um dos primeiros benefícios que vem à cabeça de qualquer pessoa que conhece a Internet das Coisas Industrial (IIoT - *Industrial IoT*) é a manutenção preventiva. A manutenção preventiva é importante para suportar melhor os produtos.
- **Criar melhor os novos produtos:** cria valor por meio da invenção. Embora a inovação com IoT melhore significativamente os produtos existentes, a invenção com IoT cria novos produtos totalmente inéditos. Por definição, esse tipo de criação de valor não é somente incremental, ou seja, ele tem o potencial de exercer grande impacto em relação às perspectivas de sucesso da empresa que aplica esse processo de modelagem de valor.

### **Passo 3 - A criação das trilhas ou dos funis do processo de venda e do caminho para o resultado**

É importante lembrar ao usuário, neste passo, que o que faz a Internet das Coisas ter uma verdadeira fama no mundo dos negócios e da tecnologia é pelo que ela proporciona. O aplicativo matador (*killer app*) da IoT, ou seja, o que torna a Internet das Coisas uma tecnologia competitiva e superior aos produtos concorrentes existentes são os resultados, o que, na verdade, é o que os clientes querem. Os clientes não querem saber sobre o produto, eles querem saber o que os produtos fazem para eles e, mais importante, que problema eles resolvem (SINCLAIR, 2018).

Então, para aplicar todas as informações adquiridas nos passos anteriores e conseguir elaborar uma estratégia para atingir o resultado esperado pelo cliente, neste passo o usuário irá construir dois *pipelines*:

1. **Pipeline de vendas:** O objetivo é criar uma sequência de eventos que devem ocorrer para atingir a venda do produto, desde a prospecção de maneira resumida.
2. **Pipeline de resultado:** O objetivo é se aproximar do *killer app* da Internet das Coisas e para isso criar uma sequência de eventos que devem ocorrer para atingir os resultados.

- **4.1.3 Sketch**

Nessa etapa, quem realiza o projeto vai desenvolver o esboço do produto que irá servir de guia para a construção do protótipo.

### **Passo 1: Trabalhando na solução de IoT**

O usuário será orientado, primeiramente, a fazer uma lista de soluções que já existem ou uma lista de produtos, ou serviços a serem analisados em busca de soluções inspiradoras, pois isso é importante para deixar o produto ainda mais competitivo e favorecer a criatividade. Não se deve restringir a procura nas soluções de outras empresas, podem ser projetos ou ideias anteriores já criadas em projetos de IoT antigos.

E depois para visualizar como a solução irá funcionar, o usuário irá registrar em tópicos como a solução irá funcionar, depois ele terá que fazer um esboço dos procedimentos desta solução num papel ou em algum *software* de desenho, usando um procedimento até hoje muito utilizado pela empresa Pixar na criação de seus roteiros e animações, o *Storyboard*. O *Storyboard* consiste numa matriz de quadrados com desenhos esboçados que contam uma história.

Na criação de um produto, o *Storyboard* é formado por uma sequência lógica de acontecimentos desenhados que simulam o funcionamento do produto e a sua interação com o cliente. Essa técnica é uma prática muito comum na produção cinematográfica para prever como o produto, no caso um filme ou série, irá acontecer e quais alterações são necessárias para melhorar antes de começarem as gravações (KNAP; ZARETSKY; KOWITZ, 2017).

### **Passo 2 - Criar o valor do produto de IoT**

Para deixar ainda mais claro de como a solução irá funcionar e esboçar os passos de funcionamento do produto para gerar valor, o usuário irá desenvolver os seguintes tópicos: definir, qual a proposta de valor do produto, ou seja, o que o produto vai fazer para atender o resultado esperado pelo cliente. Qual o modelo que representa o centro de estudo ou função que será aplicada e avaliada para tomadas de medidas? Qual o aplicativo que será feito para manipular o produto IoT de acordo com o modelo criado? Qual a análise de dados que será aplicada para descobrir a relação causal entre os dados sobre as variáveis medidas para o modelo e os eventos que estão acontecendo? Através dessas perguntas será possível definir o valor incremental gerado pela aplicação (SINCLAIR, 2018).

### **Passo 3 - Desenhar a interface gráfica.**

Neste passo, será importante o usuário começar a pensar em como o cliente vai interagir com a aplicação IoT, como visualizar os dados e quais comandos estão disponíveis para que ele tenha a possibilidade fazer as suas próprias escolhas de maneira consciente sobre o uso do produto. As ferramentas para o esboço da interface gráfica serão compartilhadas na plataforma para ajudar quem está nesse passo para desenvolvê-lo da melhor forma.

- **4.1.4 Protótipo**

Nesta etapa o usuário vai definir quais materiais e componentes que irá utilizar para montar o seu primeiro objeto inteligente que representa o seu produto e montar, de fato, um protótipo suficiente para se conseguir testá-lo e validá-lo nas entrevistas a serem feitas na próxima etapa. Então, ele deverá seguir os seguintes passos.

**Passo 1: Definição dos materiais**

Primeiramente, deve-se seguir uma arquitetura de referência para definir os materiais a serem utilizados. Usar uma arquitetura de referência básica é uma escolha prudente, pois um dos grandes pilares da tecnologia IoT é a construção de uma rede de dispositivos conectados entre si para gerar aplicações ainda maiores e mais eficientes. Então, segui-la vai ajudar a fazer com que o produto possa ser integrado a outros serviços e/ou produtos compatíveis. A arquitetura de três camadas (*three-layer architecture*) é encorajada a ser seguida no processo, pois é uma das mais simples e define a ideia principal da Internet das Coisas (PALLAVI; SMRUTI, 2017).

Figura 10. Arquitetura de três camadas

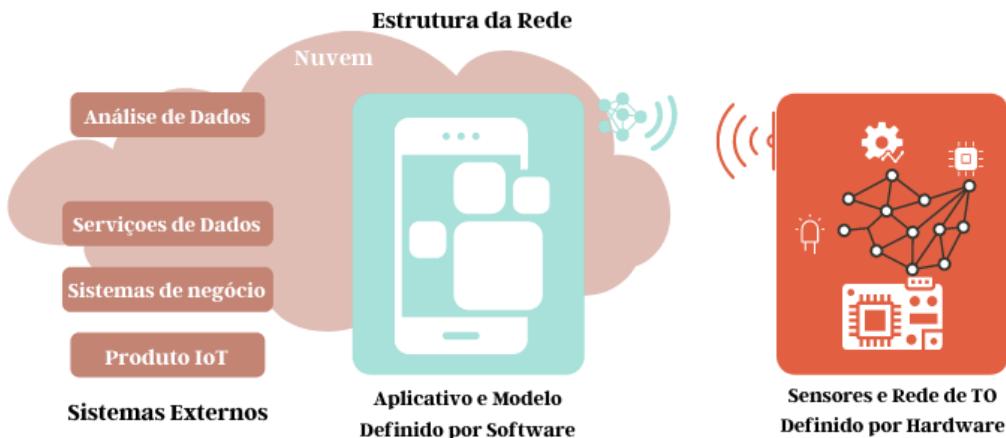


Fonte: autoria própria

- **Componente Percepção/Atuação:** Essa camada representa as “coisas” da IoT. Se refere às partes do sistema que fazem interação com o mundo físico e que recolhe informações sobre o mundo real. Ela detecta alguns parâmetros físicos ou identifica outros objetos inteligentes no ambiente.
- **Componente de Rede:** Essa é a camada responsável por fazer conexões no sistema de IoT, conexões entre objetos inteligentes ou computadores. Seus recursos também são usados para transmissão e processamento dados do sensor.
- **Componente de Aplicação:** É a parte que entrega o serviço para as pessoas. Ele utiliza os dois componentes anteriores para fazer algo útil. Define vários aplicativos em que a Internet das Coisas pode ser implantada, por exemplo, casas inteligentes, cidades e saúde inteligente.

É necessário, também, logo em seguida separar os materiais que serão utilizados a partir de uma visão de negócio, isso possibilita que o usuário explore as possibilidades de geração de valor através de cada parte tecnológica do produto, um aspecto importante de se analisar pois muitos produtos de IoT fracassam neste ponto (SINCLAIR, 2018). Para isso, a pessoa que desenvolve o projeto deve identificar partes específicas que vão compor o protótipo na divisão ilustrada, logo abaixo, na Figura 11.

Figura 11. Divisão tecnológica através de uma visão de negócio



Fonte: Adaptado de (SINCLAIR, 2018)

- **Produto definido por Software:** É a parte do sistema IoT que irá gerar valor para aplicação. O *software* descreve a funcionalidade do produto do ponto de vista de IoT, ele engloba o modelo junto ao aplicativo e a análise de dados desenvolvidos no passo 2 da etapa “*Sketch*”. Essa parte pode ser associada à camada de aplicação da arquitetura de três camadas.
- **Produto definido por Hardware:** É a parte física do produto IoT. Essa parte consiste em sensores, acionadores e sistemas embarcados que irão interagir diretamente com o meio em que estão inseridos. Uma parte que se assemelha à camada de percepção e/ou atuação da arquitetura de três camadas.
- **Estrutura da rede:** a estrutura da rede envolve tudo. Inclui as redes de TO (tecnologia das operações) e de TI (tecnologia da informação). A rede de TO é a rede instalada dentro do produto de IoT, já a de TI é a rede externa do mesmo produto.
- **Sistemas externos:** a integração do produto IoT com sistemas externos, pela internet, aumenta a funcionalidade da aplicação. Os sistemas externos fornecem dados externos para complementar os dados internos coletados por sensores do dispositivo.

## Passo 2: Criação do algoritmo

O algoritmo é uma parte importante no desenvolvimento da inteligência do dispositivo, pois vai interferir em como ele vai se comportar conforme o modelo, aplicativo e a maneira de se analisar dados determinados no passo 2 na etapa “*Sketch*”. O processo *baby steps* é encorajado a ser utilizado na plataforma por se tratar de um método simples e fácil na

criação de um algoritmo, em que basicamente há uma quebra da problemática a ser resolvida em pequenos passos traduzidos em comandos que serão implementados na lógica de programação, independe da linguagem a ser usada para se comunicar com o sistema embarcado.

### **Passo 3: Criação do protótipo**

A criação do protótipo será onde deve-se seguir todos os esboços e sequências de passos anteriormente criados com base na arquitetura de referência e divisão de negócio para geração de valor de um produto de IoT, para criar um dispositivo que possa ser testado.

Então, quem estiver desenvolvendo o projeto deverá:

1. Montar o dispositivo conforme a arquitetura de referência.
2. Testar os algoritmos desenvolvidos no dispositivo.
3. Criar a interface gráfica em que o cliente irá visualizar os dados e interagir com o *hardware*.

- **3.1.5 Teste**

Nessa etapa o usuário irá realizar encontros com os clientes e analisar o seu desempenho, fazendo algumas perguntas que comprovem algumas hipóteses e outras que ajudem a identificar possíveis falhas. Para isso, primeiramente se deve marcar as entrevistas para então, elaborar as perguntas a serem feitas, e realizá-las nas datas marcadas, para depois tomar a decisão se vai ou não divulgar o projeto na tentativa de se conseguir um investimento.

### **Passo 1: Marcar as entrevistas**

É neste passo que o usuário será instruído a marcar as entrevistas para testar o seu protótipo com pelo menos 5 possíveis clientes. Esse cliente poderá ser as pessoas que ele já se comunicou durante a “Descoberta do Cliente” na etapa “O desfaio”. Haverá um guia que estará descrito na plataforma como uma sequência de tentativas de comunicação e estratégias de persuasão para se conseguir marcar as entrevistas da melhor forma levando em conta algumas dicas compartilhadas no livro “Sprint” da Google Ventures, como a estratégia de se oferecer um presente ou um cupom para incentivar a participação nas entrevistas de teste de protótipos (KNAP; ZARETSKY; KOWITZ, 2017).

### **Passo 2: Entrevistas e Decisão**

Para ajudar na realização das entrevistas, a plataforma irá fornecer modelos de perguntas para que o entrevistador possa se inspirar na criação de suas próprias perguntas serem feitas na hora do teste, além de dar dicas de como realizar essas entrevistas e quais ferramentas usar para gerar uma experiência agradável para o entrevistado, pois ele deve ser o foco da experiência (KNAP; ZARETSKY; KOWITZ, 2017). Pode-se citar algumas dicas:

1. Deve-se dar as boas-vindas ao cliente e fazê-lo se sentir confortável;

2. Começar a dar uma breve explicação de como funciona o protótipo, mas ter o cuidado de não fornecer detalhes que interfiram na real experiência do entrevistado.
3. Antes de permitir a interação com o produto, é importante informar ao entrevistado que algumas coisas podem não estar funcionando por se tratar de um protótipo.
4. Pedir para o entrevistado, à medida que for interagindo com o objeto inteligente, falar em voz alta o que está tentando fazer e o que está achando.
5. Realizar as perguntas, elaboradas no passo anterior, ao final do teste do protótipo;

- **4.1.6 Starting**

Essa é a etapa onde o usuário poderá expor o seu projeto para possíveis investidores e vender a sua ideia. Além disso, ele poderá organizar a estrutura do empreendimento e conseguir aproveitar todas as oportunidades que a Internet das Coisas consegue gerar e conseguir alinhar os objetivos com todos os setores.

### **Passo 1: Networking e Pitch**

Uma boa rede de contatos ajuda muito na divulgação do projeto, ao encontrar parceiros e investidores fundamentais para começar a funcionar o empreendimento. O *Pitch* é a forma mais utilizada, atualmente, para apresentar um projeto empreendedor, mostrando de forma clara e objetiva o que se pretende alcançar através do produto desenvolvido.

O *Pitch* é uma apresentação curta e direta, normalmente de 3 a 15 minutos de duração e tem o objetivo de vender a ideia de uma *startup* para um possível investidor. No livro “A Arte do Pitch” (*The Art of the Pitch*) escrito pelo Peter Coughter, é compartilhada uma visão diferente sobre o *Pitch*, no livro pode-se concluir que na visão do autor esse tipo de apresentação não se trata de um trabalho com o foco na criatividade, mas em estabelecer uma confiança com o cliente. Eles precisam confiar no que está sendo vendido. As pessoas que assistem o *Pitch* podem até se esquecer do trabalho apresentado, mas se lembrarão de como se sentiram a respeito dele (COUGHTER, 2012). Essa é uma ótima perspectiva que pode ser passada para quem está no final do seu projeto de IoT.

### **Passo 2: Departamento por departamento**

Existem boas práticas que uma empresa de IoT deve implementar no seu dia a dia e que cada departamento deve aplicar, mesmo não sendo uma área ligada à parte técnica da tecnologia. Nesse passo o usuário receberá um PDF com algumas dicas e conselhos sobre boas práticas a serem seguidas após a entrada no mercado para manter a empresa operando da melhor forma e conseguir aproveitar das vantagens de se trabalhar com tecnologia IoT.

- **4.1.7 Caso de uso de um ar-condicionado inteligente**

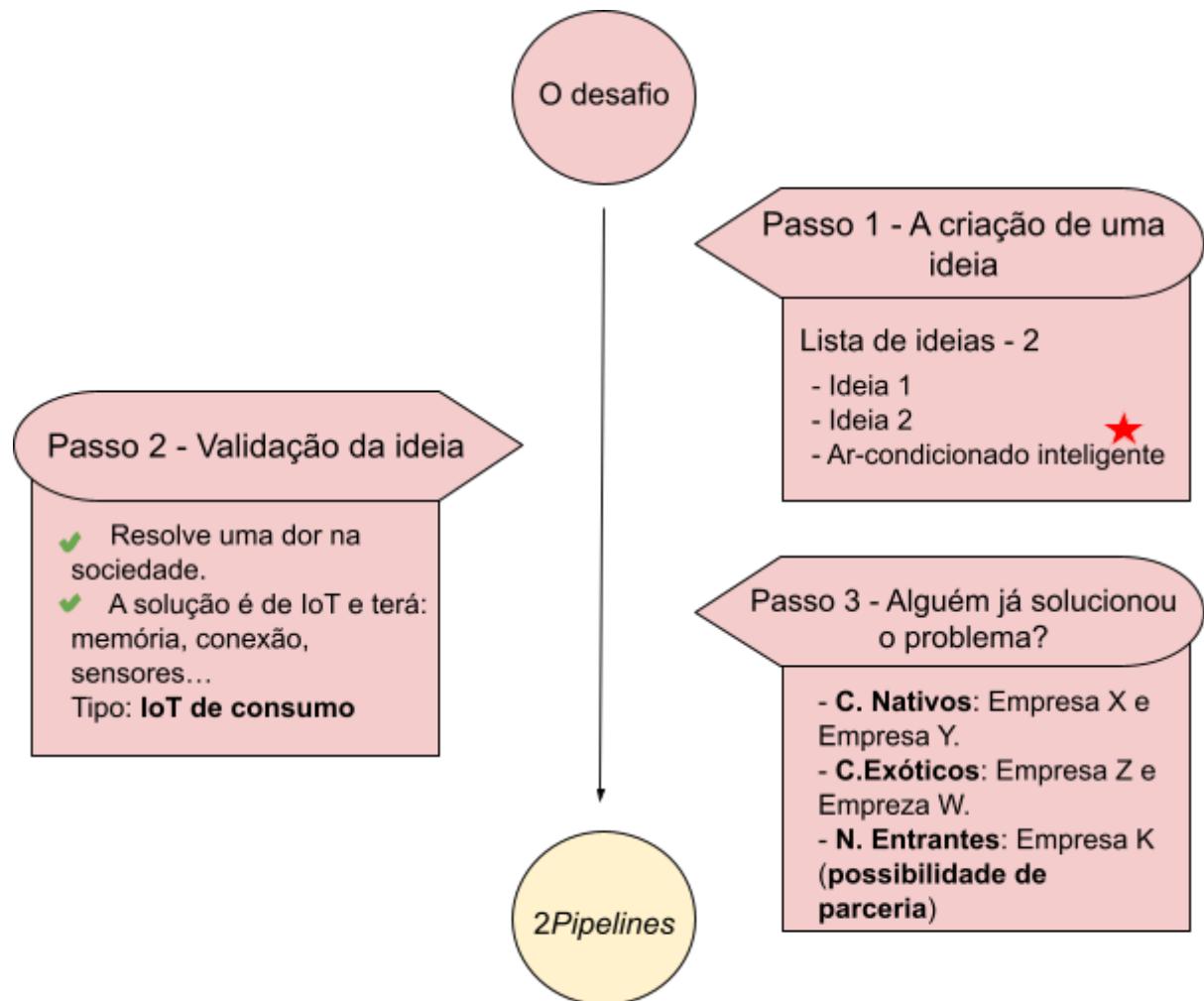
Será explicado um caso de ar-condicionado inteligente para esclarecer como o processo funcionaria na prática. Esse caso será um projeto fictício, pois como se trata de um

processo de realização de projeto IoT recém-criado não houve testes para comprovar a viabilidade, logo não existem casos reais para serem exemplificados.

Então, o projeto fictício utilizando o processo *Silent Sprint* envolve a construção de um ar-condicionado inteligente que deve possuir um comportamento independente do usuário para climatizar o ambiente em que está instalado conforme a temperatura ótima determinada pelo usuário e também gerar economia de energia.

Bom, o processo se inicia com “O desafio”, como descrito no processo. No esquema abaixo tem-se a primeira etapa do projeto com os seus respectivos resultados de cada passo desenvolvido na parte prática, simulando como o projeto deveria ter sido feito seguindo as etapas do *Silent Sprint* para a transformação de um ar-condicionado normal, numa máquina inteligente.

Figura 12. Etapa “O desafio” - Caso do ar-condicionado inteligente fictício



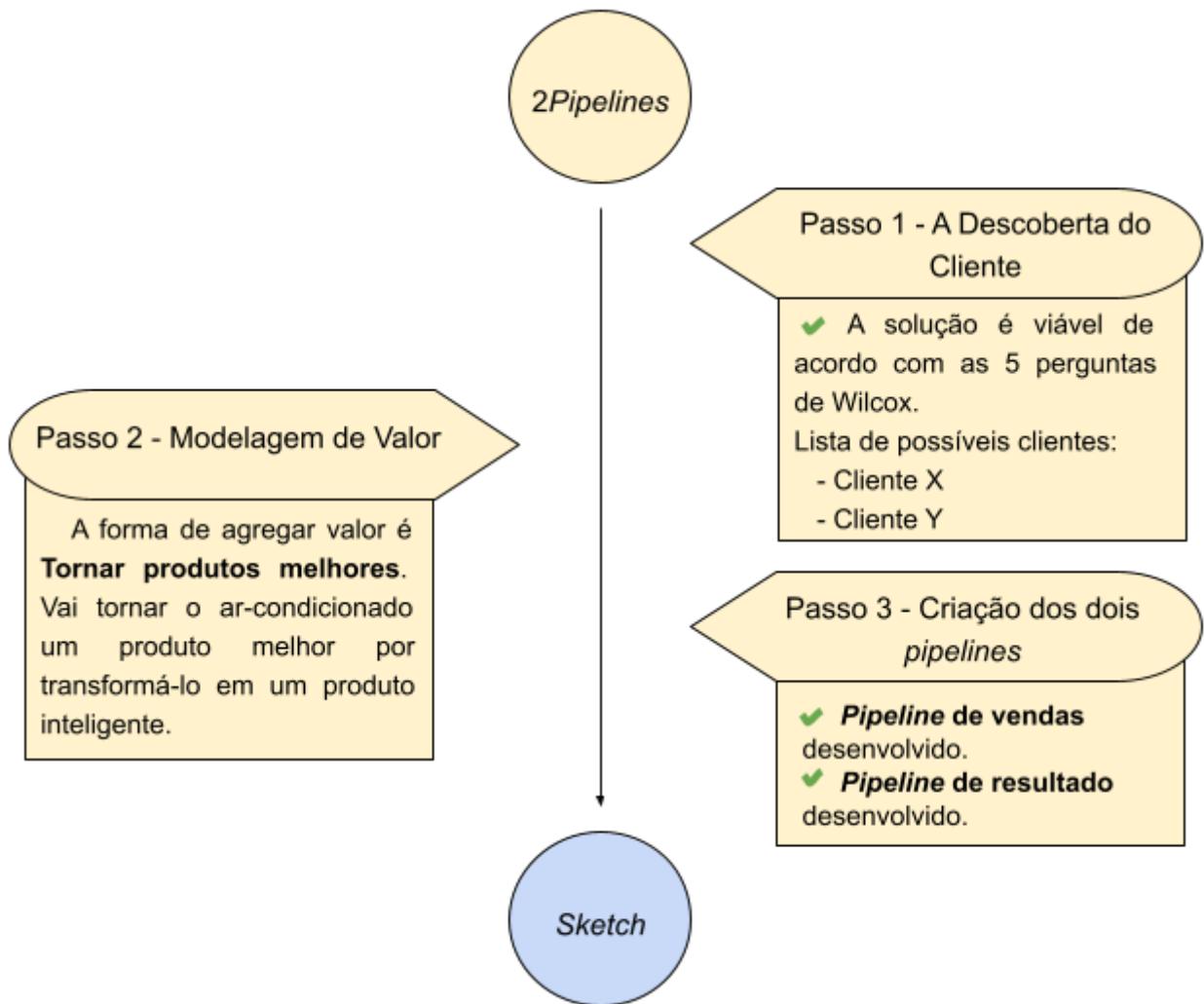
Fonte: autoria própria

O projeto, de forma fictícia, conseguiu passar pela primeira etapa, encontrando a sua grande ideia, o ar-condicionado inteligente, passando pelo Funil de Ideias “A partir da sua

vivência” e Funil de Ideias “A partir do mercado”, no primeiro passo. Logo depois, a ideia foi validada, no passo dois, verificando se a solução criada consegue resolver um problema na sociedade e se realmente ela é uma ideia que usa IoT. Por fim, no final da etapa foi listado os concorrentes para conseguir verificar se já existe alguma ideia parecida com a de um ar-condicionado inteligente e se já consegue sanar o problema encontrado a partir de uma visão de negócio. Com isso, foi possível, também, listar empresas que possam se tornar parceiras ao invés de concorrentes, como a Empresa K que possui um serviço de manutenção preventiva para ar-condicionado que pode ser integrado ao sistema de análise de dados do dispositivo IoT a ser desenvolvido.

Após ter passado pela etapa “O desafio”, o projeto segue para etapa “2Pipelines”, em que vai ser possível validar um pouco mais a ideia com a técnica “Descoberta do Cliente” do John Wilcox e saber qual a maneira de se criar valor com a tecnologia IoT conforme a solução elaborada. Com esses passos anteriores feitos, o último passo dessa etapa deve ser a construção do *pipeline* de vendas e do *pipeline* de resultado como explicado e guiado na plataforma. Na Figura 13, logo abaixo, pode-se visualizar o esquema dessa etapa.

Figura 13. Etapa “2Pipelines” - Caso do ar-condicionado inteligente fictício

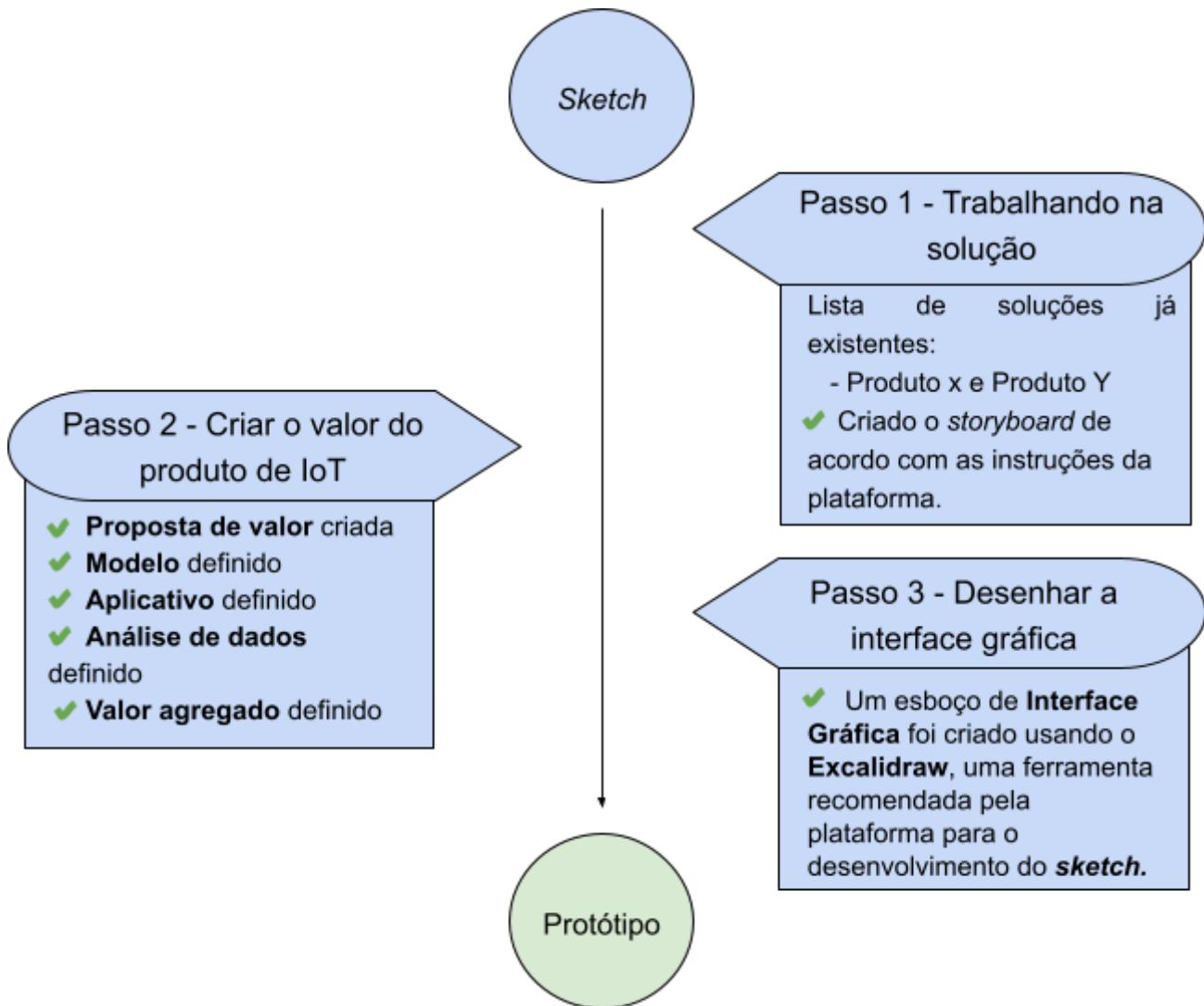


Fonte: autoria própria

Com essa etapa feita, consegue-se visualizar como a ar-condicionado vai estar agregando valor através da inovação aplicada com a criação de um dispositivo IoT que torne o ar-condicionado num produto melhor. Isso ajuda a organizar esquematicamente as duas linhas horizontais de passos a serem cumpridos para se chegar na venda do produto e no resultado para o cliente, sendo importante para o desenvolvimento de um negócio IoT.

Seguindo para o “*Sketch*”, é inicialmente pesquisado sobre as soluções existentes para tê-las como inspiração de sistemas inteligentes econômicos para a climatização de um ambiente, antes mesmo de se fazer o esboço, uma técnica compartilhada pelo livro “*Sprint*” da Google Ventures que ajuda nesse processo de criação do *sketch* (KNAP; ZARETSKY; KOWITZ, 2017). O esquema dessa etapa é apresentado na Figura 14, logo abaixo.

Figura 14. Etapa “*Sketch*” - Caso do ar-condicionado inteligente fictício



Fonte: autoria própria

Uma das atividades mais importantes a serem elaboradas no processo de *Silent Sprint* é a criação de valor do produto de Internet das Coisas, abordada no passo 2 da etapa “*Sketch*”. Para o caso do ar-condicionado inteligente fictício, a criação de valor pode ser representada pelos tópicos no Quadro 1.

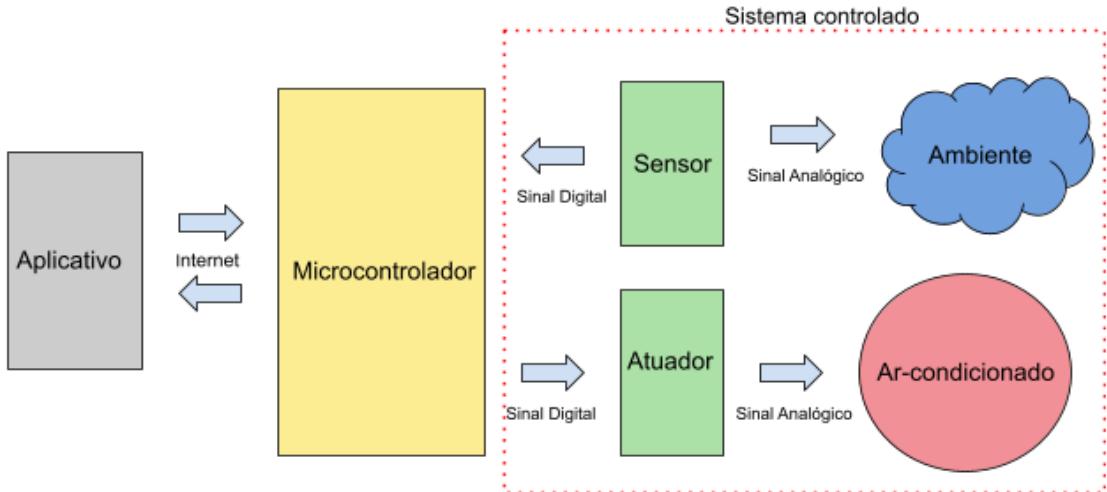
Quadro 1 - Criação de valor - Caso do ar-condicionado inteligente fictício

Proposta de valor	Controlar a temperatura ambiente de forma inteligente e econômica
Modelo ( $tempo = f(E, T)$ )	De acordo com o modelo, o tempo de funcionamento do ar-condicionado provavelmente vai mudar com o consumo de energia e a temperatura ambiente.
Aplicativo	Um <i>software</i> que deve possibilitar o controle do dispositivo pelo usuário e transparência sobre a economia de energia.
Análise de Dados	<ul style="list-style-type: none"> <li>● Avaliar o tempo de ligado.</li> <li>● Avaliar o consumo de energia.</li> <li>● Descobrir a tendência de mudança da temperatura</li> </ul>

Fonte: autoria própria

Bom, finalizada a etapa “*Sketch*” com o esboço da interface gráfica que os possíveis clientes vão interagir durante a entrevista realizada na etapa “*Teste*”, pode-se seguir para elaboração do protótipo. O “*Protótipo*” para o ar-condicionado inteligente fictício, para não ter que envolver altas despesas na construção de algo que seja real o suficiente para que as pessoas no teste possam julgar com sinceridade e ao mesmo tempo ter uma experiência agradável ao testá-lo, tem-se que fazer o mínimo para tornar a aplicação viável para o cliente, mas ainda assim barata e atender uma arquitetura de referência de IoT para assegurar que ele terá um comportamento de um dispositivo inteligente.

Então, foi decidido que o dispositivo IoT a ser criado deverá ser um dispositivo não invasivo, ou seja, não será instalado no ar-condicionado, mas ainda assim terá todos os componentes necessários para interagir com o ar-condicionado e transformá-lo numa máquina inteligente. Abaixo, na Figura 15, encontra-se um esquema do que será desenvolvido como produto definido por *hardware* com os componentes básicos de atuação e percepção seguindo a arquitetura de três camadas e também o produto definido por *software* representado pelo aplicativo.

Figura 15. Esquema de funcionamento do *hardware*

Fonte: autoria própria

Então, para a construção do protótipo será necessário um aplicativo para possibilitar que o cliente interaja com o dispositivo IoT e receba alertas importantes, como consumo em tempo real de energia e o estado do ar-condicionado para saber se ele está ligado ou não. É importante o uso de um microcontrolador para ter o armazenamento de memória e processamento necessário para coletar os dados dos sensores, analisá-los de acordo com o modelo criado, no passo 2 da etapa “*Sketch*” e enviá-los para o aplicativo através da Internet. Para a percepção e atuação, deve-se ter um sensor que consiga medir variáveis de ambiente, como umidade e temperatura, e um atuador que consiga se comunicar com o ar-condicionado, que não torne o dispositivo invasivo, como, por exemplo, atuadores que emitem infravermelho, implementados nos controles de ar-condicionado remoto, atualmente.

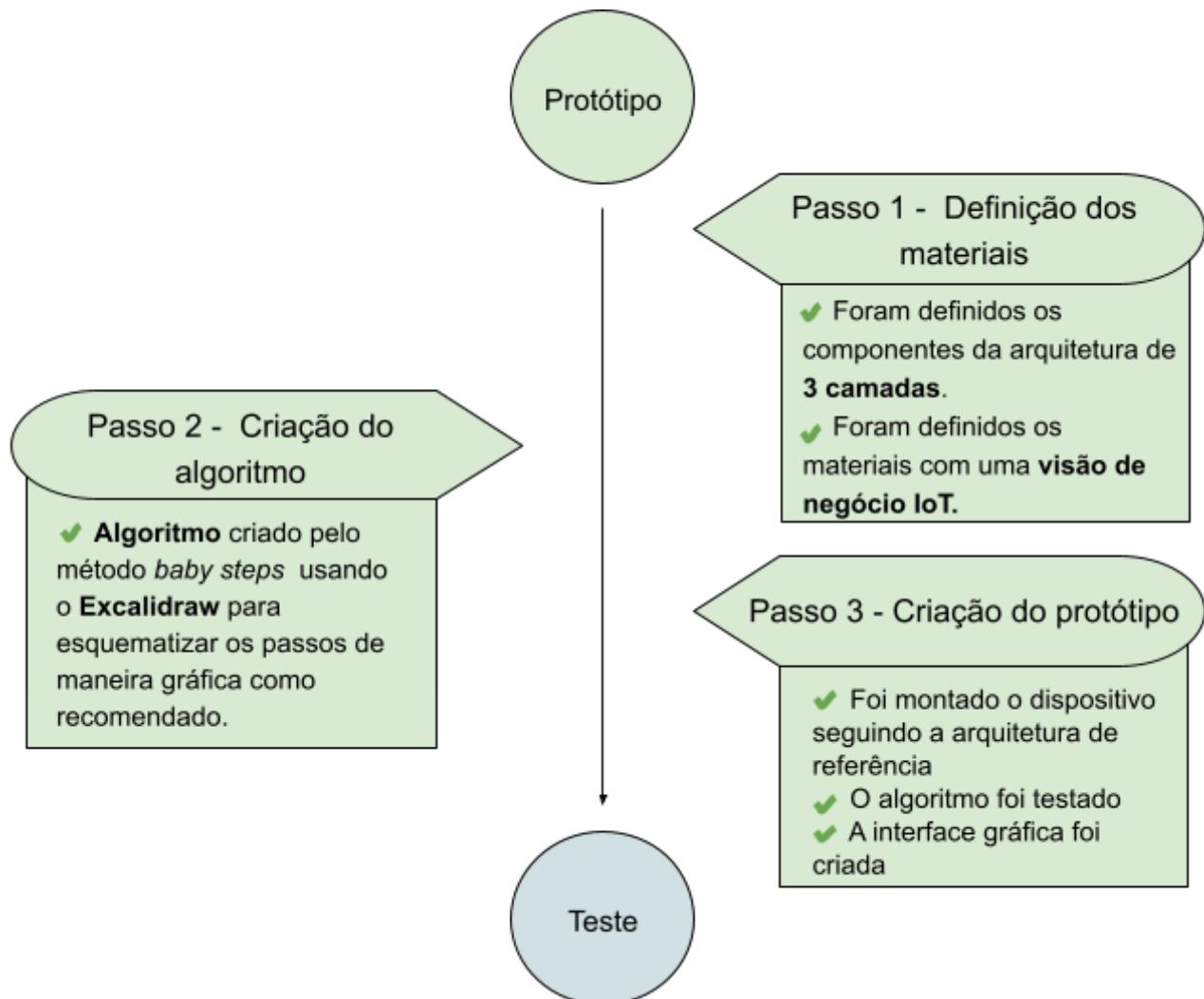
Antes de partir para a parte prática de construção do protótipo, é necessário se criar o algoritmo para o desenvolvimento da parte inteligente do produto, independentemente da linguagem escolhida. E para isso, seguindo o que processo orienta, deve-se criar passos objetivos e pequenos (algoritmo do tipo *baby steps*) do que deverá ser feito para atingir o objetivo de solução do problema, que no caso do ar-condicionado inteligente fictício, é ter um controle da temperatura e umidade com base numa temperatura ótima pré-determinada pelo cliente no aplicativo e ao mesmo tempo proporcionar a economia de energia. Então, o algoritmo para esse caso deve ser:

1. O ar-condicionado deve decidir se vai ligar ou desligar com base na temperatura ambiente.
2. Ao ligar deve esperar chegar até a temperatura ótima para decidir se vai poupar energia desligando-o ou não.
3. O ar-condicionado deve identificar que chegou na temperatura ótima e desligar. Ainda assim deve-se analisar se o tempo de desligado é maior que 40 min, tempo recomendado para deixar o ar-condicionado desligado para compensar a energia consumida ao ligá-lo, até chegar numa temperatura diferente da ótima.

4. O ar-condicionado deve identificar que chegou na temperatura ótima e desligar. Ainda assim, deve -se analisar se o tempo de desligado é maior que 40 min até se chegar a uma temperatura diferente da ótima.
5. Caso a resposta seja "sim". Quer dizer que o tempo de ligação pode ser subtraído do tempo que ele deve estar ligado.
6. Caso a resposta seja "não". Quer dizer que o tempo de ligado deve ser somado ao tempo que ele deve estar ligado.

Após definido os materiais e o algoritmo, o protótipo deve ser montado seguindo o esquema de funcionamento já definido anteriormente e testado localmente para conseguir garantir as funcionalidades necessárias para poder ser utilizado por possíveis clientes nos testes da próxima etapa. O esquema que resume o que hipotéticamente deveria ter sido feito na etapa de protótipo está na Figura 16, logo abaixo.

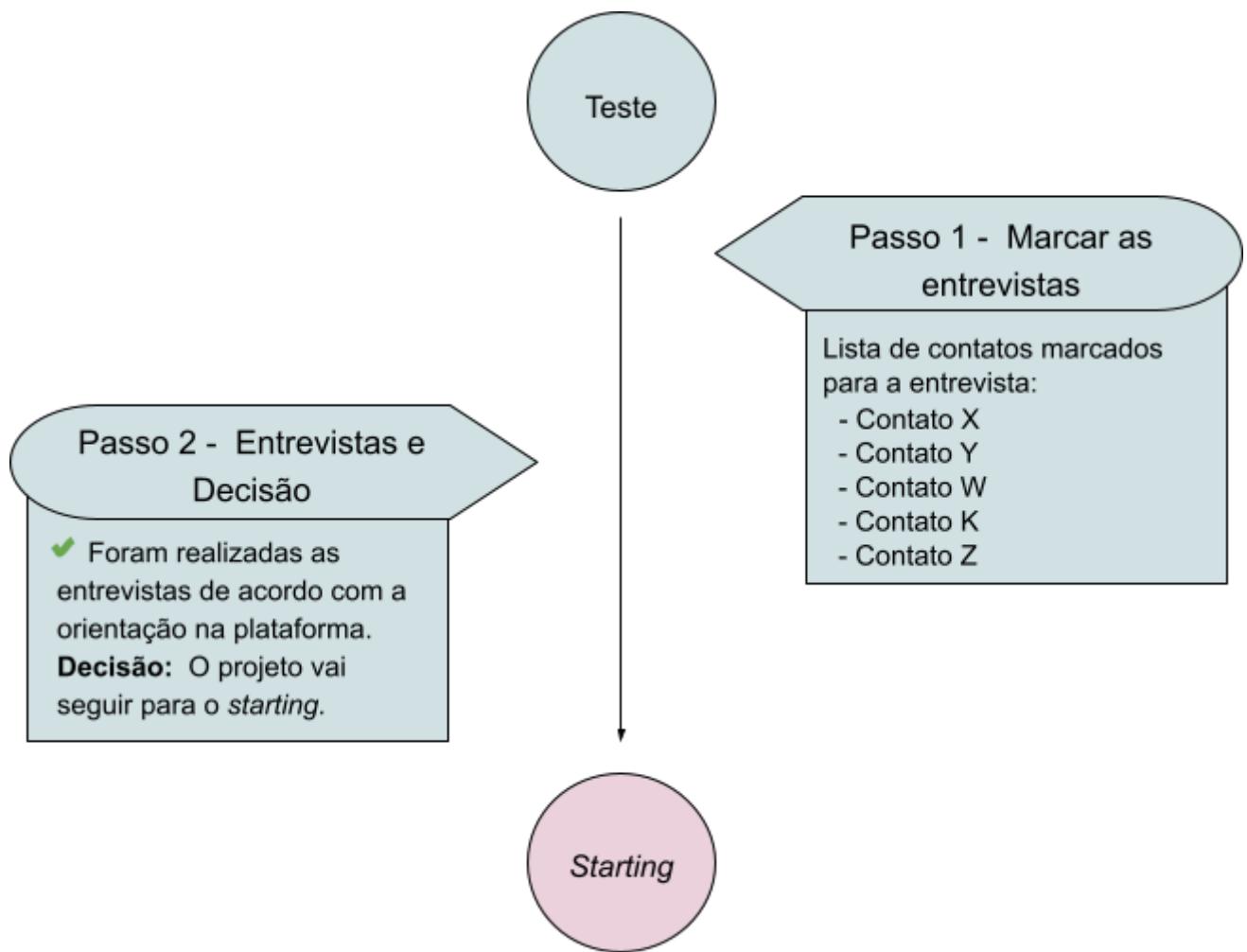
Figura 16. Etapa “Protótipo” - Caso do ar-condicionado inteligente fictício



Fonte: autoria própria

Na próxima etapa, é escolhido um lugar que tenha um ar-condicionado para realizar as entrevistas e poder mostrar ao cliente o que o projeto conseguiu criar e quais resultados o dispositivo IoT pode trazer para a vida dele. Logo em seguida, são marcados os encontros com os entrevistados e elaboradas as perguntas guiadas pela plataforma, que vão ajudar a validar a hipótese da viabilidade da ideia de IoT e assim decidir se vai ou não finalizar o projeto com a etapa “Starting”. O esquema abaixo, na Figura 17, mostra como a etapa de “Teste” do ar-condicionado inteligente foi feita.

Figura 17. Etapa “Teste” - Caso do ar-condicionado inteligente fictício

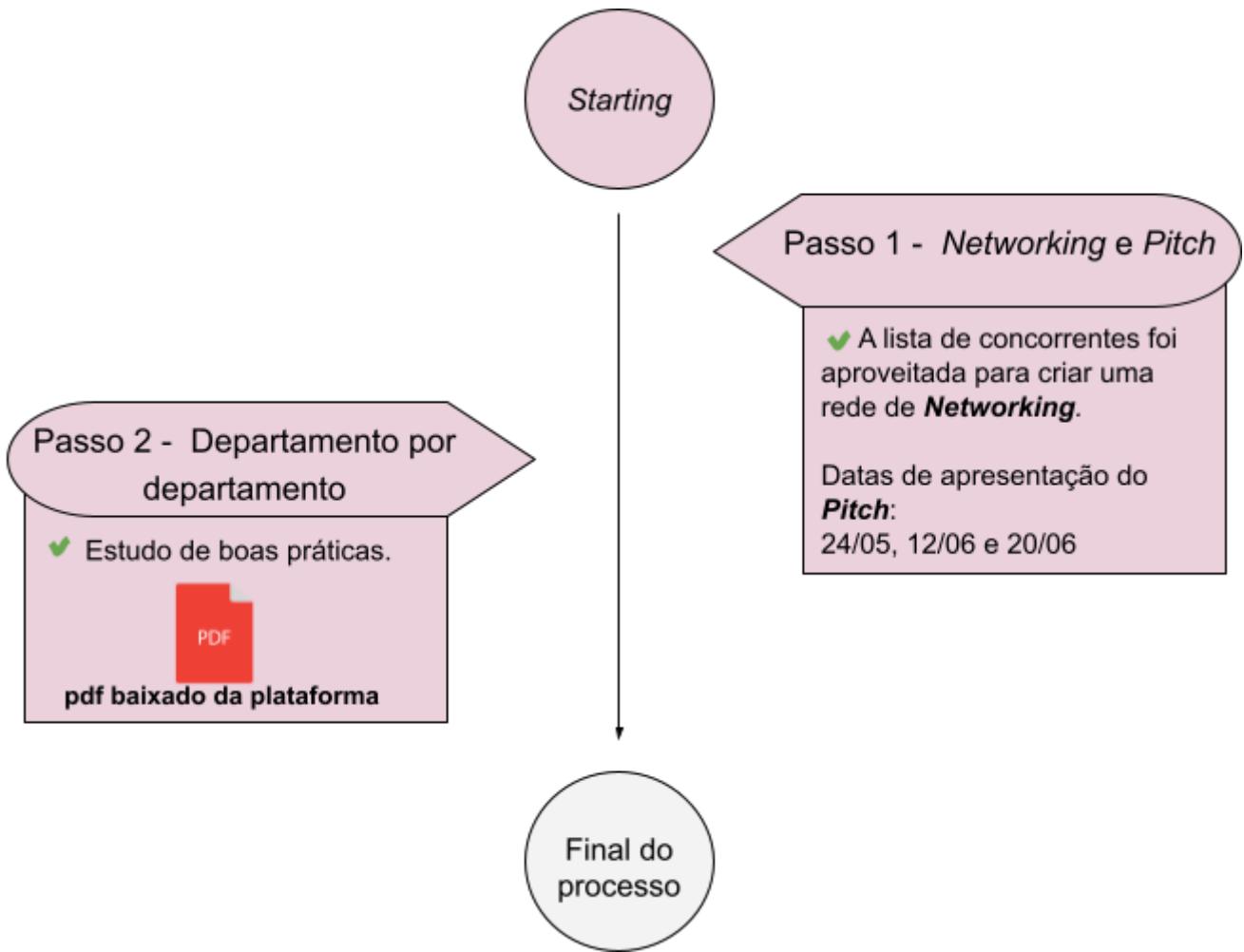


Fonte: autoria própria

Então, supondo que os testes foram um sucesso e com base nos dados coletados, foi possível concluir que o protótipo está pronto para ser divulgado e apresentado para os investidores. Um projeto incrível de tecnologia que atende aos critérios de um produto de IoT de sucesso será apresentado. Só falta, então, preparar o *Pitch* e apresentá-lo. Para isso é necessário criar uma rede de comunicação com pessoas interessadas no seu produto e marcar as apresentações, tendo sempre em mente que o protótipo criado vai atender os resultados esperados pelo cliente quando tiver o apoio e integração de outros serviços e produtos no seu

ecossistema de atuação através de parcerias. No final do processo, o projeto vai ter um apoio da plataforma de boas práticas que uma empresa de IoT deve seguir para mantê-la atualizada às oportunidades que a evolução da tecnologia consegue trazer e aproveitar dos benefícios de se trabalhar com uma tecnologia emergente, no caso, a Internet das Coisas. Logo abaixo pode-se encontrar o esquema da etapa “Starting”, na Figura 18, finalizando o processo *Silent Sprint* do caso do ar-condicionado inteligente fictício.

Figura 18. Etapa “Starting” - Caso do ar-condicionado inteligente fictício



Fonte: autoria própria

#### 4.2 A construção da identidade visual da plataforma

Assim como o fluxograma do processo, a identidade visual da plataforma *Silent Sprint* foi criada usando a ferramenta Canva, uma plataforma para *design* intuitiva e acessível para a criação dos mais diversos projetos de *design*, e foi útil no desenvolvimento das versões do logotipo e na escolha das cores que caracterizam a plataforma (CANVA).

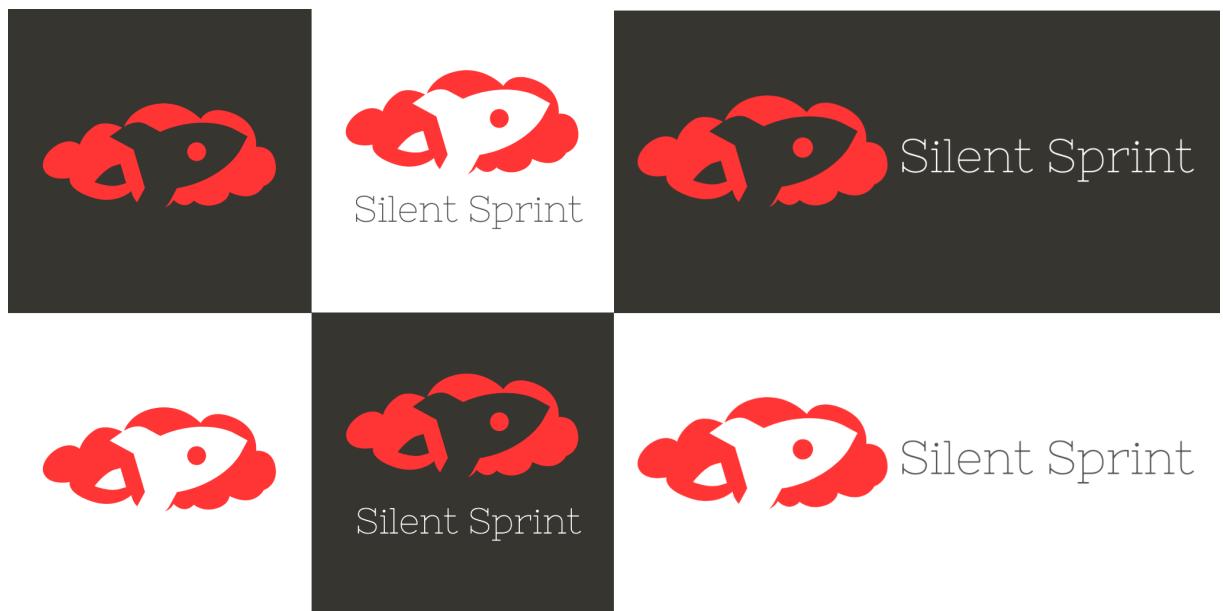
Considerando a psicologia das cores e a experiência do usuário (UX - *user experience*) sabe-se que cada componente, imagem e texto que se cria e que se mostra em um site tem que

gerar algum sentido para quem está vendo e interagindo com ele e por vezes provocar uma ação, como, por exemplo um "click" em um botão para gerar engajamento.

Na escolha de uma cor, deve-se evitar que a mensagem que ela passa seja mal interpretada da forma em que se chega para quem está vendo, se o objetivo é chamar a atenção, provocar uma ação e passar uma mensagem de forma rápida, a melhor opção seria a cor vermelha. A luz vermelha é uma luz de menor frequência, logo de maior comprimento de onda no espectro visível, então ela é captada rapidamente pelo ser humano comparada às outras frequências de luzes (LOPES, 2013). Experiências mostram que o vermelho consegue estimular o corpo humano, fazendo com que ocorra um aumento na pressão sanguínea e do número de batimentos cardíacos, por exemplo. Desta forma, essa cor se torna importante para estimular a agilidade que é um fator importante para o processo de projeto IoT desenvolvido.

Então, optou-se para a plataforma, uma identidade visual que contenha as cores vermelho, branco e cinza. O logotipo contém ilustrações que representam o ato de realizar um projeto empreendedor, um foguete, e um símbolo muito utilizado para representar a Internet das Coisas, tecnologia central a ser utilizada no processo, a nuvem.

Figura 19. Versões do logotipo



Fonte: autoria própria

#### 4.3 O *design* da plataforma

Antes de começar toda a parte de desenvolvimento web em um projeto é importante fazer um *design* que sirva de referência na hora de criar as linhas de código que quando interpretadas pelo *browser* possam gerar a interface gráfica que vemos na parte do *front-end* da aplicação. Para isso, temos o *Wireframe* e o *Mockup* que são dois estilos diferentes de *design*, mas que o segundo depende do primeiro para ser construído de forma clara e organizada.

*Wireframe* é um desenho básico da estrutura de uma determinada interface, em que não envolve a preocupação com critérios estéticos, mas que demonstra de forma simplificada como o produto final deverá funcionar (TEIXEIRA, 2017).

Já o *Mockup* é o *design* que mais se aproxima de forma estática da interface final. Mesmo essa modalidade não possuindo interatividade e testes de usabilidade, ela vai traduzir a identidade visual da marca em páginas web criadas por meio de um *software* de criações gráficas para atingir o objetivo do projeto.

Após desenhado o *wireframe* da plataforma em um papel para servir de guia sobre a estrutura que a aplicação deve ter, seguiu-se para o *Mockup*. Foi utilizado o Canva como ferramenta de design novamente e as cores já estabelecidas para a identidade visual da *Silent Sprint*. O *Mockup* foi dividido em seis páginas fundamentais para o desenvolvimento do site e a garantia de uma boa usabilidade para os usuários. Optou-se por criar apenas a versão para computador e tablet da plataforma web para garantir uma ótima experiência do usuário e a realização do projeto sem empecilhos devido a alguns momentos necessitar de um notebook.

### Página 1: *Login* na plataforma

Antes do usuário conseguir acessar a *Home* e aproveitar a experiência na plataforma, ele deverá ser autenticado pelo sistema na página de *login*. A autenticação vai ser realizada pelo *back-end* desenvolvido, através de uma API com a geração de um token para permitir que o usuário não tenha que fazer o *login* novamente na plataforma caso a página seja fechada. As informações a serem verificadas pelo *back-end* vão ser enviadas pelo formulário do *front-end* da página de *login*, como mostrado, logo abaixo, na Figura 20.

Figura 20. *Login* na plataforma e Requisitos

Silent Sprint



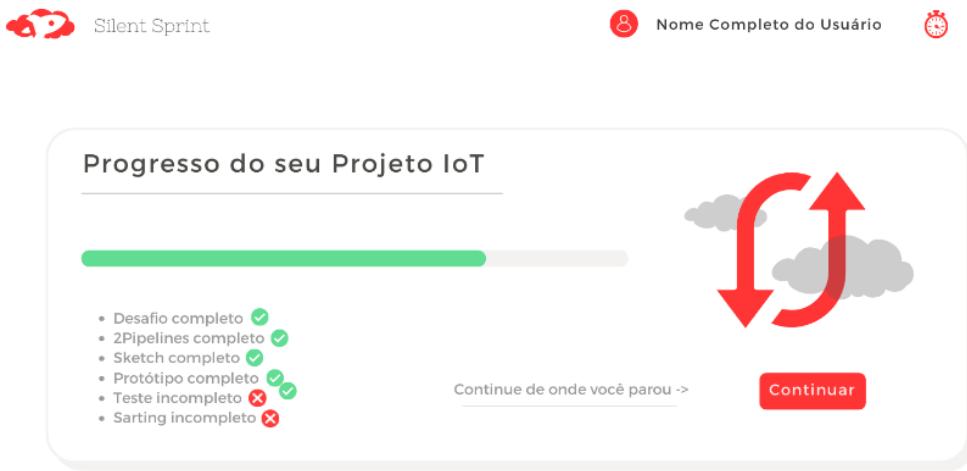
Fonte: autoria própria

## Página 2: Home

A *Home* é a parte em que o usuário terá uma visão geral do que a plataforma tem a oferecer e quais as etapas a serem seguidas para se realizar o projeto IoT e ela pode ser dividida em três partes principais.

A primeira parte desta página deve conter um acompanhamento do progresso do usuário, como pode-se observar na Figura 21 para que ele tenha visibilidade do quanto falta para ele terminar o projeto, já que se trata de um método bem guiado para realização de um projeto de IoT.

Figura 21. Componente de progresso da página *Home*.

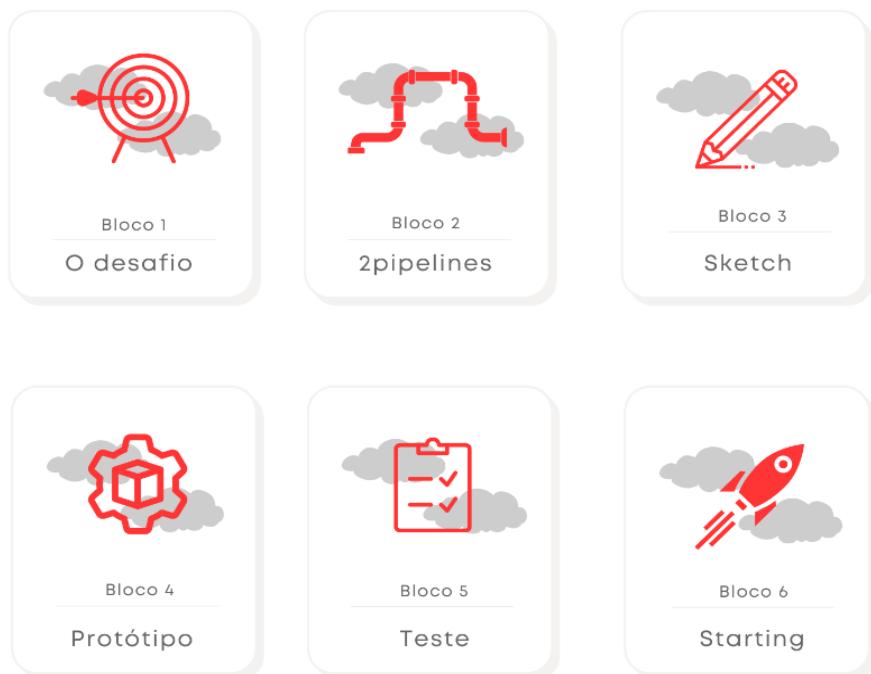


Fonte: autoria própria

A segunda parte deve mostrar o total de etapas que o usuário deverá cumprir para realizar o projeto e proporcionar uma transparência sobre o processo que ele vai aprender e acompanhar na plataforma, como pode-se observar na Figura 22, logo abaixo.

Figura 22. Componente de trilha do projeto da página *Home*.

### Trilha do Projeto

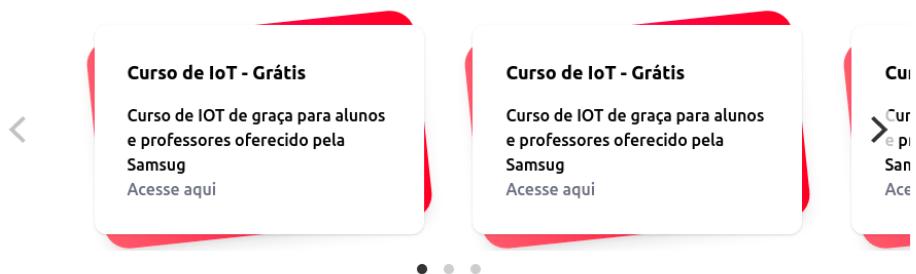


Fonte: autoria própria

A terceira parte deve conter uma série de recomendações de conteúdos extras, que o usuário pode consumir para ajudá-lo em seu processo, e que estará próximo ao rodapé da página. A terceira etapa é mostrada na Figura 23, logo abaixo.

Figura 23. Componente de recomendações da página *Home*.

### Recomendações



Silent Sprint feito por Eduarda Wiltiner com Tailwind, Next.js e ESLint

Fonte: autoria própria

### Página 3: Detalhes de cada etapa

A transparência e a disponibilização de conteúdos aos poucos são aspectos importantes quando se trata da experiência do usuário, então é importante antes de levá-lo para o conteúdo e atividades de cada etapa, mostrar alguns detalhes sobre o que está por vir, como mostrado na Figura 24.

Figura 24. Página de detalhes

O desafio

O que é?

Essa primeira etapa serve para entrar numa linha de pesquisa que possibilite ter mais segurança sobre os passos seguintes do projeto. Alguns projetos podem terminar apenas nesse primeiro passo, fazendo com que os usuários economizem tempo e energia para validar outras ideias

Passo 1: A criação e validação de uma ideia

1.1 Introdução

1.2 Funil de Ideias "A partir da sua vivência"

1.3 Funil de Ideias "A partir da sua vivência"

Fonte: autoria própria

### Página 4: Conteúdo e tarefas

O conteúdo é parte mais importante da plataforma e é o que vai guiar o usuário durante todo o processo, por isso ele terá uma página reservada em que será possível entregar os textos e as atividades para ele gradualmente, para poder consolidar conceitos e realizar práticas ligadas ao conteúdo estudado em cada passo. A Figura 25, logo abaixo, ilustra a página de conteúdo e tarefas do projeto.

Figura 25. Conteúdo na plataforma

The screenshot shows a user interface for a challenge step. At the top, there are icons for a user profile, a name, and a clock. The main title is "O desafio -> Passo 1: A criação e validação de uma ideia". Below this, there are "Anterior" and "Próximo" buttons. The main content area is titled "O você vai fazer?". It contains a text block about the first step being a research phase, followed by a list of four bullet points. To the right of this is a video player with a play button, volume controls, and a progress bar. The progress bar is mostly grey with a small yellow segment on the left. At the bottom right is a "Próximo" button. The left side of the interface features a sidebar with sections like "Introdução", "Funil de Ideias - Sua Vivência", and "Hora da prática!", each with "O que é?" and "Como fazer?" sub-sections.

Fonte: autoria própria

## Página 5: Mecanismo de *feedback*

Para uma plataforma sempre crescer e avançar com melhorias é preciso de *feedbacks*, apenas assim pode-se mensurar e fazer uma análise de dados para tomadas de decisão que impliquem na melhoria da usabilidade e interface, bem como no conteúdo. Um botão vermelho e redondo que possa disponibilizar perguntas para a avaliação da experiência na plataforma, é uma ótima opção e pode ser visto na Figura 26, logo abaixo.

Figura 26. Sistema de coleta de *feedback* na plataforma



Fonte: autoria própria

### Página 6: *Loading*

Todo site que possui uma preocupação com a demora do carregamento de cada componente das páginas acessadas deve ter uma página de *loading*, como mostrado na Figura 27, para manter a interação do usuário enquanto os componentes são carregados.

Figura 27. Página de *Loading* na plataforma



Fonte: autoria própria

### Página 7: Erro 404

O status 404 representa um erro na plataforma e significa “Página não encontrada” e normalmente aparece quando o usuário digita a URL do site errado, esse erro é um dos mais

comuns na experiência do usuário em qualquer site. Então, para que o usuário entenda o que precisa ser feito e o que aconteceu, esse erro deve ser tratado e mostrado de uma forma mais compreensível para quem está acessando a plataforma. Na Figura 28, logo abaixo, é mostrada a página de Erro 404.

Figura 28. Erro 404 na plataforma



Fonte: autoria própria

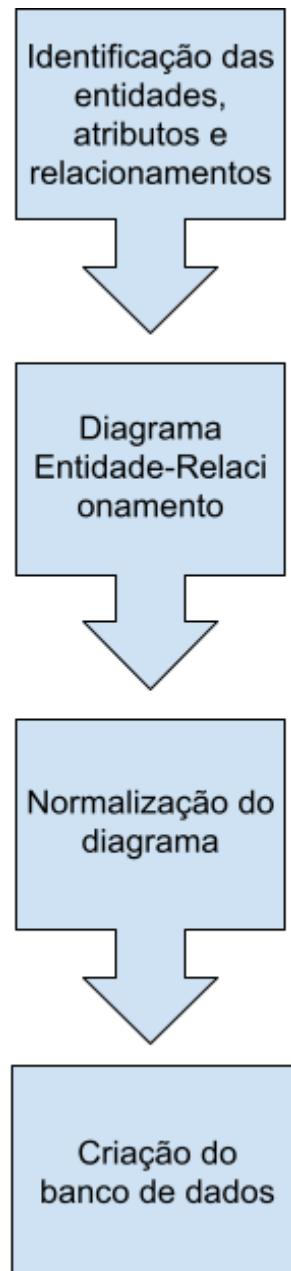
#### 4.4 Modelagem do banco de dados

O *back-end* de uma aplicação é toda parte programada e desenvolvida que não é visível ao usuário e que possui uma conexão direta com o banco de dados, seja para armazenar, buscar, atualizar e deletar dados.

Então, para desenvolver o *back-end* é preciso utilizar alguns conceitos, arquiteturas e bibliotecas importantes para assim conseguir oferecer uma experiência mais personalizada para cada usuário. Para isso, a parte inicial e mais importante é realizar a modelagem de um banco de dados.

Já que se trata de uma plataforma bem guiada para realização de projetos de IoT, o usuário deve ter a visibilidade e a compreensão do seu próprio avanço no projeto e para isso é necessário criar um banco de dados para guardar as informações de cada usuário. A modelagem do banco de dados foi realizada em quatro etapas, como mostrado na Figura 29, logo abaixo:

Figura 29. Etapas para a modelagem do banco de dados



Fonte: autoria própria

### **Etapa 1: Identificação das entidades, atributos e relacionamentos**

Durante a modelagem de um banco de dados, em que se passa por algumas etapas para criar e estruturar o banco de dados, é necessário saber alguns conceitos para a obtenção de algumas informações que vão ajudar no processo, como saber o que são entidades, atributos e relacionamentos.

As entidades são comumente conhecidas por tabelas no banco de dados e representam definições mais próximas da realidade como, por exemplo, uma tabela que armazena dados dos usuários chamada de “Pessoas” é uma entidade. Nessa entidade teremos dados como CPF, idade, altura e peso para identificar cada pessoa, esses são conhecidos como colunas e chamam-se atributos. Se se tem outra entidade chamada “Livros Comprados”, tem-se uma ligação com a entidade “Pessoas”, pois é preciso saber e identificar quem comprou cada livro, logo tem-se o chamado relacionamento entre elas.

Para o caso da plataforma *Silent Sprint* as entidades, atributos e relacionamentos para que possa realizar a autenticação, acompanhar o progresso do usuário e mostrá-lo a ele como está indo e armazenar as informações de *feedback*, são:

- **Entidade “Users”:** representa os dados de cada usuário importantes para a autenticação na plataforma. **Atributos:** o identificador único do usuário (*user\_id*), o nome do usuário (*user\_name*), a senha do usuário (*user\_password*), o email (*user\_email*) e a data de criação (*created\_data*). **Relacionamento:** relacionamento com as tabelas *feedback* e *progress*.
- **Entidade “Content”:** representa cada conteúdo que o usuário vai ler na forma de códigos numéricos e inteiros criados e que possuem 4 números que se dividem em 4 colunas respectivas. **Atributos:** o identificador único do conteúdo (*content\_id*), o código numérico referente ao conteúdo (*content\_code*), o código numérico referente a etapa (*stage\_code*), o código numérico referente ao passo (*step\_code*) e o código numérico referente ao tópico de estudo (*topic\_code*). **Relacionamento:** relacionamento com tabela *progress*.
- **Entidade “Feedback”:** a tabela que deve armazenar todos os *feedbacks*, com indicadores de Nível de Satisfação do Cliente (CSAT ou *Customer Satisfaction Score*) na escala de 1 a 5, no formato de estrelas, e comentários digitados e compartilhados pelos usuários junto com a identificação do mesmo. **Atributos:** o identificador único do *feedback* (*feedback\_id*), o número do CSAT (*csat*), o comentário enviado (*comment*), a data do feedback (*feedback\_date*) e o identificador do usuário (*user\_id*). **Relacionamento:** relacionamento com a tabela *Users*.
- **Entidade “Progress”:** a tabela que vai relacionar a identificação do usuário junto a identificação do conteúdo que ele já leu para medir o seu progresso. **Atributos:** o identificador único do progresso do usuário (*progress\_id*), o identificador do usuário (*user\_id*), o identificador do conteúdo estudado pelo usuário (*content\_id*) e a data de progresso (*progress\_date*). **Relacionamento:** relacionamento com a tabela *Users* e *Content*.

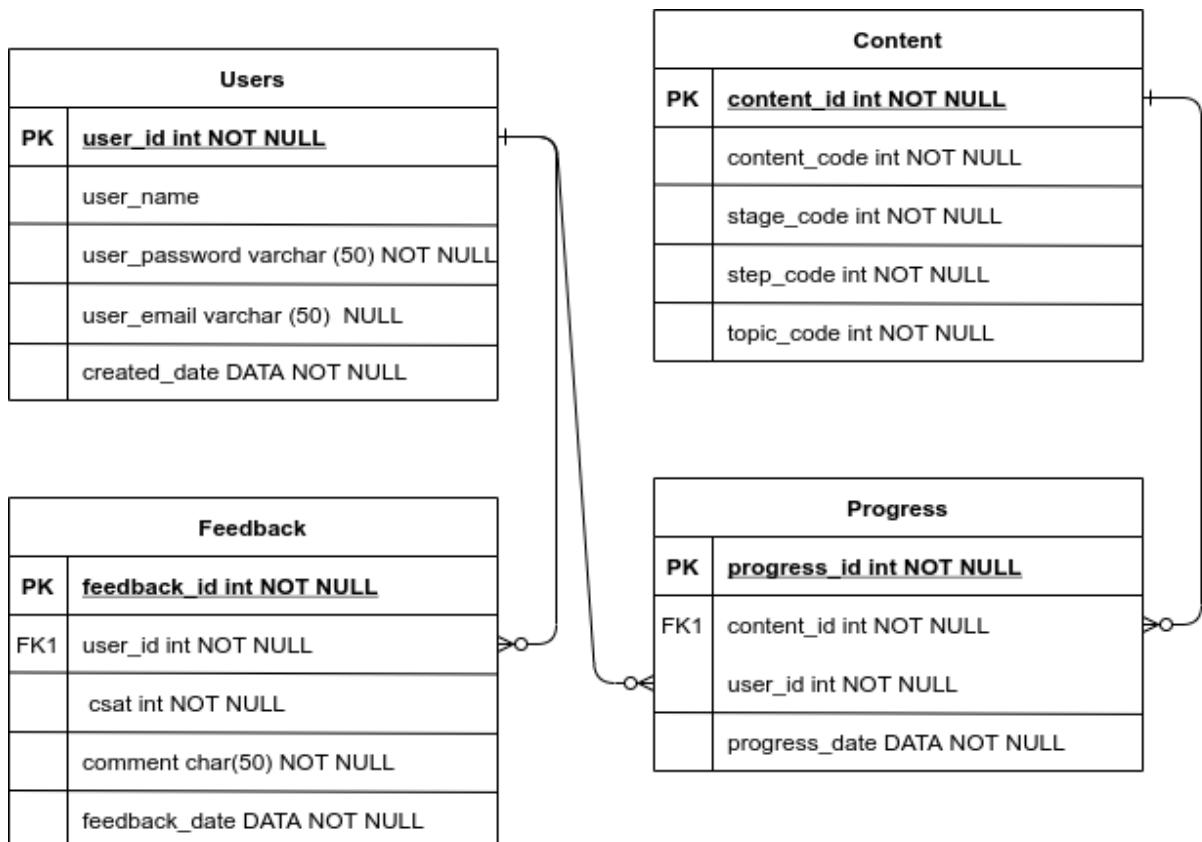
## **Etapa 2: Construção de um diagrama entidade-relacionamento**

Para organizar e mostrar o relacionamento entre as entidades é necessário fazer um diagrama de entidade-relacionamento, que nada mais é do que uma representação gráfica das entidades com os seus respectivos atributos, conectadas através de linhas que representam os

relacionamentos. O diagrama pode ser criado através do site [draw.io](http://draw.io) que possui um modelo a ser preenchido para essa categoria e é uma boa ferramenta de construção de esquemas gráficos em projetos (SEIBERT MEDIA).

Então para representar de maneira gráfica as informações conseguidas anteriormente, temos o seguinte diagrama com as entidades e seus atributos junto aos seus respectivos tipos de variáveis, apresentado na Figura 30.

Figura 30. Diagrama entidade-relacionamento



Fonte: autoria própria

#### Etapa 4: Normalização do diagrama

Uma parte importante de se fazer durante a modelagem de um banco de dados é normalizar toda a estrutura criada no diagrama entidade-relacionamento, isso significa que é necessário atender algumas exigências na estrutura para conseguir garantir uma baixa chance de gerar erros ao armazenar os dados, a falta de redundâncias, uma facilidade no entendimento das colunas e tabelas, uma boa organização e melhor manutibilidade. Para isso, tem que fazer com que a estrutura atenda a 1º, 2º e 3º Forma Normal (ALEXANDRUK, 2007). Explicando cada uma das formas normais têm-se:

- Na 1º Forma Normal: a tabela deve ter atributos que não contenham grupos de repetição.

- Na 2º Forma Normal: a tabela não deve conter colunas que não estejam ligadas diretamente ao identificador único (id), pois não deve haver uma dependência funcional parcial, por exemplo, na tabela “*User*” com o identificador único sendo “*user\_id*”, não deve conter os comentários de *feedback*, pois não estão diretamente ligados ao usuário.
- Na 3º Forma Normal: é parecida com a segunda forma normal, mas agora não é permitido a dependência funcional, ou seja, não seria permitido uma coluna na tabela “*Content*” que pudesse concatenar os códigos de cada coluna.

Feita a Normalização do diagrama, o próximo passo deve ser a criação do banco de dados.

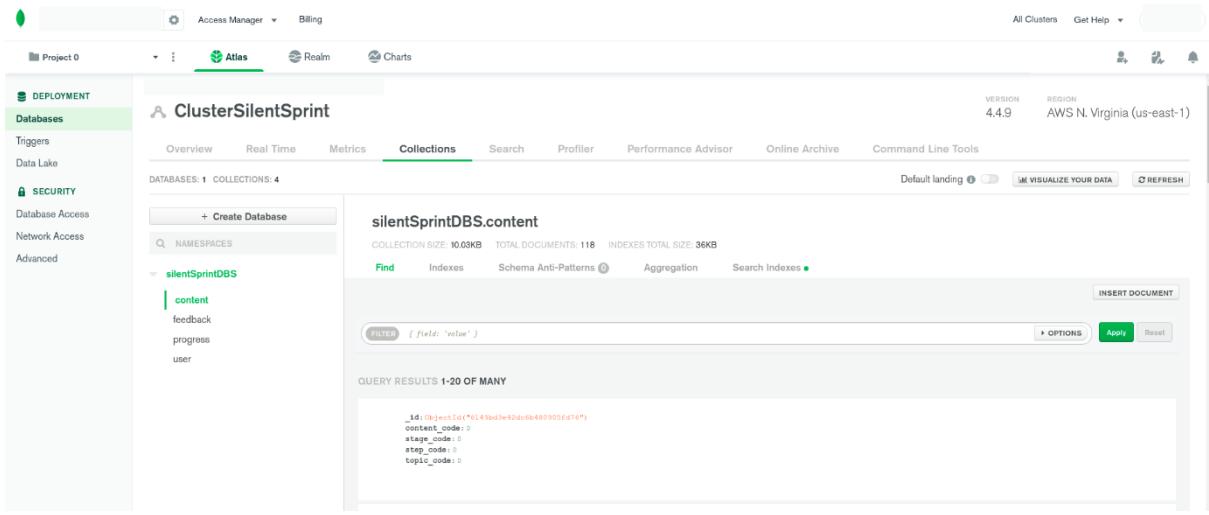
### **Etapa 5: Criação do banco de dados**

Para se criar um banco de dados você precisa definir qual o Sistema de Gerenciamento de Banco de Dados (SGBD) você vai usar para a manipulação dos seus dados, o que vai definir também que tipo de linguagem você vai utilizar para realizar consultas ao seu banco de dados e onde você poderá hospedar o seu banco de dados. Para este presente trabalho foi escolhido o SGBD MongoDB.

MongoDB é um sistema de gerenciamento de banco de dados NoSQL, em que é possível lidar com a criação de consultas, que são linhas de códigos que através de uma linguagem de programação conseguem realizar chamadas ao banco de dados, seja para criar, ler, atualizar ou deletar os dados. No caso do MongoDB, por ser NoSQL, ele não fica restrito ao uso da linguagem específica SQL, por isso é um SGBD que vai facilitar na criação de Interfaces de Programação de Aplicativo (API) para este trabalho usando a linguagem JavaScript (JS) para se conectar e consultar o banco de dados criado em MongoDB (MONGODB). Uma outra vantagem que o uso de SGBD NoSQL traz é a ausência da obrigação de se ter as regras de negócio implementadas previamente no seu banco de dados, o que por um lado pode se comprometer a integridades dos dados, por outro lado pode ser reparado com atenção e segurança na implementação após alguns testes no ambiente de desenvolvimento.

Escolhido o SGBD, resta agora hospedar o banco de dados num serviço ou plataforma que tenha suporte para o MongoDB. Logo, foi escolhido o MongoDB Atlas que é um DBaaS (*Database-as-a-Service*), um serviço hospedado em nuvem para bancos de dados de documentos do MongoDB, em que os usuários não precisam se preocupar com a configuração de *hardware* ou instalação de *software*. Ele oferece suporte a todos os três principais provedores de nuvem (AWS, Azure e GCP) e para este trabalho foi escolhida a provedora AWS, por ser uma provedora muito utilizada por empresas de tecnologia (MONGODB ATLAS). Abaixo se encontra o serviço contratado na plataforma Atlas, na Figura 31.

Figura 31. Serviço no Atlas de gerenciamento do banco de dados

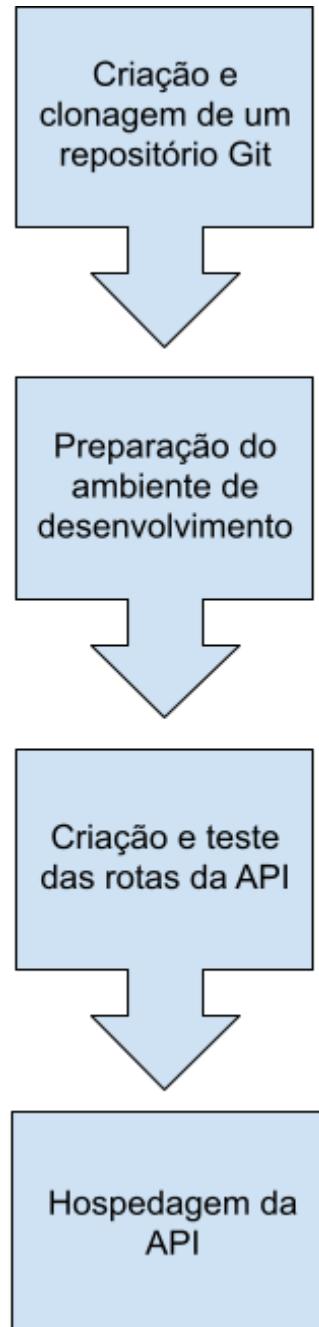


Fonte: autoria própria

## 4.5 Criação e hospedagem da API

O papel de uma API é conectar duas aplicações diferentes através de padrões e protocolos e ela é normalmente usada na integração do *back-end* com o *front-end*. Para que haja essa conexão do *back-end*, que vai interagir diretamente com o banco de dados, com o *front-end*, a interface do usuário, e estabelecer uma estrutura que manipule os dados conforme as regras de negócio e, em simultâneo, mantenha nas condições aceitas pelo SGBD MongoDB, é necessário a criação de uma API. Foram em torno de sete etapas para criação e hospedagem da API como pode-se perceber no esquema da Figura 32.

Figura 32. Esquema de criação da API



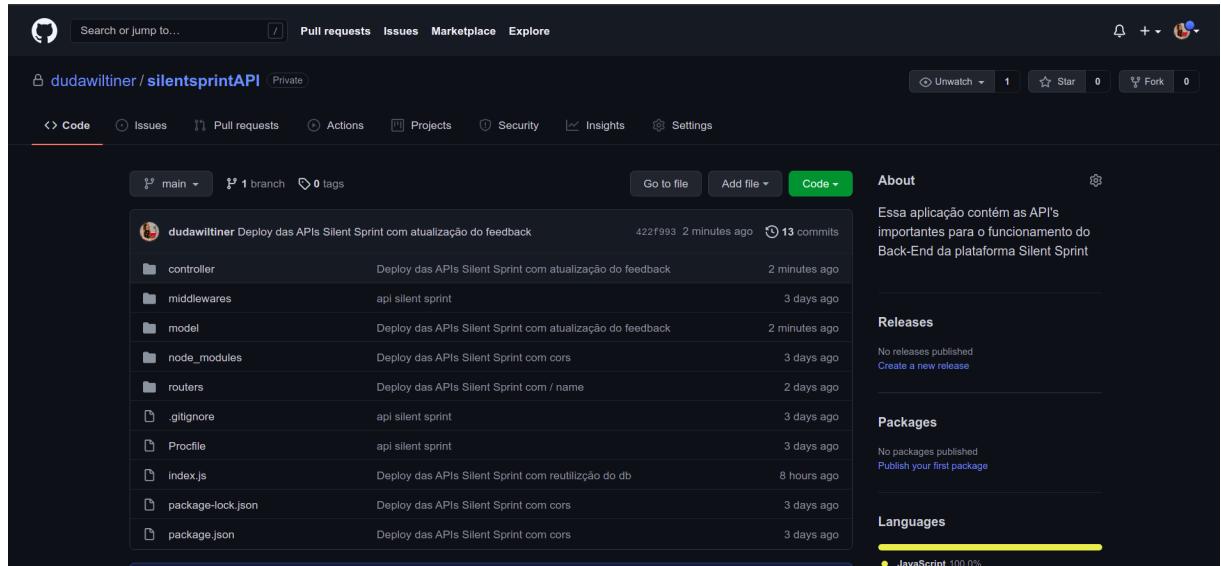
Fonte: autoria própria

### **Etapa 1: Criação e clonagem de um repositório Git**

Primeiramente foi criado um repositório particular no GitHub, como mostrado na Figura 33, para controle de versionamento da aplicação back-end através do Git. O GitHub é a forma líder de hospedagem de código com literalmente milhões de projetos com seus códigos hospedados nele. Em conjunto com o Git, um sistema open-source para controle de

versionamento, fornece os meios para um fluxo de trabalho de desenvolvimento produtivo e é, atualmente, a ferramenta referida entre os desenvolvedores (PIPINELLIS, 2018).

Figura 33. Repositório da API *Silent Sprint* no GitHub



Fonte: autoria própria

Com o repositório remoto criado, para utilizá-lo localmente para o desenvolvimento da aplicação tem-se que clona-lo no computador para ajudar na preparação do ambiente de desenvolvimento, em que será basicamente formado pelo repositório clonado, a IDE de desenvolvimento da aplicação e bibliotecas importantes para criação da API.

## Etapa 2: Preparação do ambiente de desenvolvimento

Após ter clonado o repositório localmente, foram criados os arquivos separados no *Visual Studio Code* (VSC), a IDE de programação escolhida para o desenvolvimento da aplicação por ser um *software* intuitivo e muito utilizado no desenvolvimento de aplicações, atualmente, e também foram instaladas as bibliotecas e os frameworks necessários e mais importantes para poder se criar a API *Silent Sprint*.

As bibliotecas são um conjunto de linhas de código salvas num arquivo ou módulo que reúnem classes, funções e objetos que vão auxiliar no desenvolvimento web, elas possuem um único objetivo de existência e possuem compatibilidade limitada a cada linguagem usada para acessá-la. Já um *framework* a sua única diferença para uma biblioteca é que ele não possui apenas um objetivo de uso, pois suas funcionalidades são compatíveis com mais de uma linguagem, o que ajuda em um projeto que por exemplo, deverá usar linguagens como JavaScript, HTML e CSS em simultâneo.

As bibliotecas instaladas são:

- **Dontev:** É um módulo de dependência zero que serve para carregar e armazenar variáveis ambiente em um arquivo *.env*. Fazer isso faz com que dados sensíveis como o link de conexão com o banco de dados e chaves secretas não apareçam para usuários sem permissão quando realizar o *deploy* da aplicação e ela estiver em produção (MOTTE, 2013).
- **Biblioteca de Token Web em Json:** Uma biblioteca que tem a função de possibilitar a implementação do *Json Web Token* (JWT) uma maneira compacta para transmitir informações e requisições de forma segura. Essas informações podem ser verificadas e são confiáveis, pois recebem uma assinatura criptografada (AUTH0).
- **Cors:** É um pacote node.js que pode ser instalado na máquina sendo utilizado como uma biblioteca que fornece um *middleware* Connect/Express que pode ser usado para dar permissão ao CORS, que é quem protege as requisições do lado do cliente para o servidor. Ele é importante, para permitir que a API possa ser usada no front-end de uma aplicação sem ser barrada por padrões de segurança (GOODE, 2018).
- **Body-parser:** Essa é uma biblioteca muito importante pois, nas requisições HTTP o *body* da chamada, que normalmente são no formato *json*, deve ser uma string, por isso chega num formato diferente nas solicitações recebidas. É por isso que para tratar essa informação enviada no corpo da solicitação e aplicar as regras de negócio nos dados enviados para então fazer a consulta ao banco de dados, é necessário aplicar o *body-parser* (WILSON, 2018).

Os frameworks instalados são:

- **Express:** É um framework minimalista e de fácil uso para Node.js para facilitar a criação de APIs em JavaScript (STRONGLOOP).
- **MongoDB Driver:** Permite que aplicativos Node.js se conectem ao MongoDB e trabalhem com dados através de um *driver* que pode ser instalado na máquina e usado como uma biblioteca (MONGODB, INC).

Como a aplicação back-end será feita com JavaScript, é preciso ter um interpretador de código que possa ser usado para “correr” os arquivos de desenvolvimento e assim testá-los localmente antes de subir o projeto para produção. Para o uso do JavaScript o Node.js é uma ótima opção. O Node é um *runtime* ou interpretador de JavaScript (JS) que pode ser instalado no computador e é usado para rodar arquivos de JS desenvolvidos num ambiente de programação como, por exemplo, o VSC. É muito utilizado na criação de APIs no *back-end* (LINUX FOUNDATION, 2021).

Então, preparado o ambiente de desenvolvimento pode-se iniciar a programação das rotas da API.

### Etapa 3: Criação e teste das rotas da API

Na criação de uma API é fundamental seguir uma arquitetura de software de referência, pois isso vai ajudar na organização do seu código, nas divisões de funções e responsabilidades específicas e na manutenção do código. Para a criação de APIs, a arquitetura mais utilizada no mercado é a MSC (*Model, Service and Controller*).

- **Modelo (M):** a camada de modelo reúne arquivos JavaScript que vão realizar operações com o banco de dados importantes para gerar uma conexão contínua e executar consultas.
- **Serviço (S):** a camada de serviço reúne arquivos JavaScript em que terão o papel de usar métodos desenvolvidos na camada de modelo baseado nas regras de negócio criadas pensando nas requisições a serem feitas e nas respostas correspondentes da API.
- **Controladores (C):** a camada de controladores é a parte mais próxima da pessoa usuária ou de uma requisição e com base nelas vai chamar as devidas funções da camada de serviço.

Além do uso da arquitetura MSC é recomendado que se utilize o padrão de boas práticas REST que é um conjunto de boas práticas que faz com a API esteja clara e facilite a sua implementação no *front-end*. Então para se obter uma API RESTful é necessário seguir algumas restrições como (RESTFULAPI):

- A interface de comunicação entre o servidor e o cliente deve ser uniforme e deve ser definida e seguida à risca para que ela se mantenha consistente.
- O recurso a ser acessado pela API deve ser identificado pelo *endpoint*, ou seja, a url que quando usada vai realizar a requisição.
- A ação que a API deve realizar deve estar bem clara e identificada pelo verbo HTTP da requisição. Sendo os principais: POST, GET, PUT e DELETE.

Bom, com o uso da arquitetura MSC e das boas práticas REST foi possível desenvolver a API *Silent Sprint* com suas respectivas rotas que vão fazer consultas ao banco de dados para manipular as tabelas ou entidades criadas. Mas antes de hospedá-las e subi-las para produção é necessário testá-las e para isso foi utilizado o Postman, que é uma plataforma de API que serve para criar e testar APIs de forma fácil e intuitiva. No software consegue-se realizar as requisições pela API, verificar se a autenticação do usuário está funcionando e se as respostas estão chegando no formato certo, por exemplo (POSTMAN).

Depois de ter feito testes manuais simples para conferir se as rotas estão funcionando, pode-se hospedá-las, então.

## Etapa 4: Hospedagem da API

Para que quando a plataforma esteja no ar e qualquer pessoa cadastrada possa acessá-la, a API também deve estar no ar para poder, por exemplo, fazer a autenticação do usuário remotamente, a aplicação *back-end* precisa ser hospedada e para isso foi escolhido o Heroku. O Heroku é uma PaaS (*Platform-as-a-Service*), um serviço que consegue abstrair as complexidades de se administrar um servidor e que permite aos desenvolvedores construir, executar e operar aplicativos inteiramente na nuvem. Aplicativos que podem ser escritos em Node.js, Ruby, Java e Python, isso vai ajudar pois a API foi desenvolvida com o Node.js (HEROKU). Logo abaixo, pode-se visualizar a Figura 34.

Figura 34. Repositório da API *Silent Sprint* no Heroku

The screenshot shows the Heroku dashboard for the 'silentprint' application. At the top, there are tabs for Overview, Resources, Deploy, Metrics, Activity, Access, and Settings. The Overview tab is selected. The dashboard displays the following information:

- Installed add-ons:** \$0.00/month (Configure Add-ons)
- Dyno formation:** \$0.00/month (Configure Dynos)
- Collaborator activity:** eduardawiltiner@gmail.com (Manage Access)
- Latest activity:**
  - Today at 4:45 PM - v15: Deployed (422e993e) - Build succeeded
  - Today at 4:44 PM: View build log
  - Today at 9:08 AM - v14: Deployed (e6e6b6d8) - Build succeeded
  - Today at 9:07 AM: View build log
  - Yesterday at 12:08 AM - v13: Deployed (4341e34e) - Build succeeded
  - Yesterday at 12:08 AM: View build log
  - Sep 28 at 9:47 PM - v12: Deployed (70aa360e) - Build succeeded
  - Sep 28 at 9:47 PM: View build log

Fonte: autoria própria

Agora a API pode ser usada pelo *front-end* através das requisições feitas pelas seguintes rotas com a autorização sempre verificada de quem está realizando a requisição:

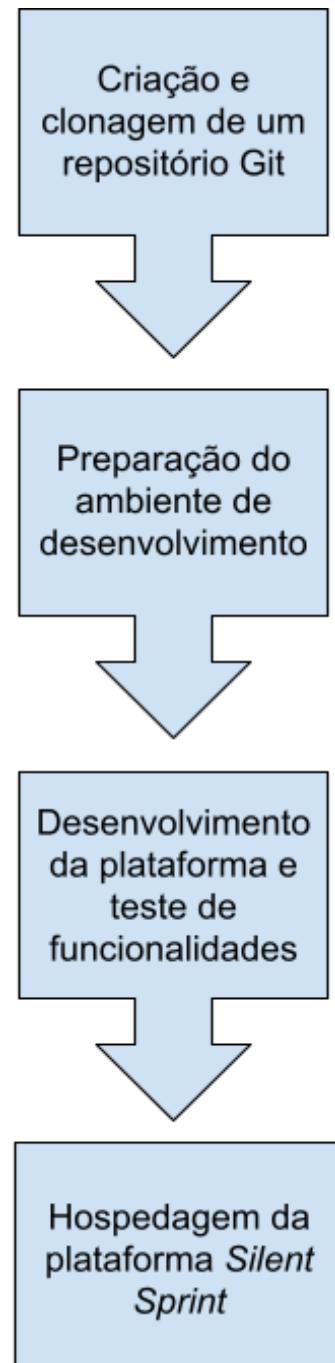
- <https://silentprint.herokuapp.com/progress>
- <https://silentprint.herokuapp.com/user>
- <https://silentprint.herokuapp.com/feedback>

## 4.6 A programação e hospedagem da plataforma

O *front-end* de uma aplicação é toda parte programada e desenvolvida que é visível ao usuário que está acessando o serviço na internet, pode se dizer que o *front-end* compõe a interface do usuário. Quando se interage com o LinkedIn, por exemplo, se interage com o *front-end* deste site.

Para que o usuário tenha uma boa experiência com a plataforma o front-end deve ser muito bem elaborado e conseguir disparar eventos e carregar páginas de forma dinâmica e rápida para manter o engajamento do mesmo. As etapas, com passos similares aos do *back-end*, seguidas para produzir o código da aplicação e sua hospedagem foram as seguintes:

Figura 35. Esquema de criação da plataforma *Silent Sprint*

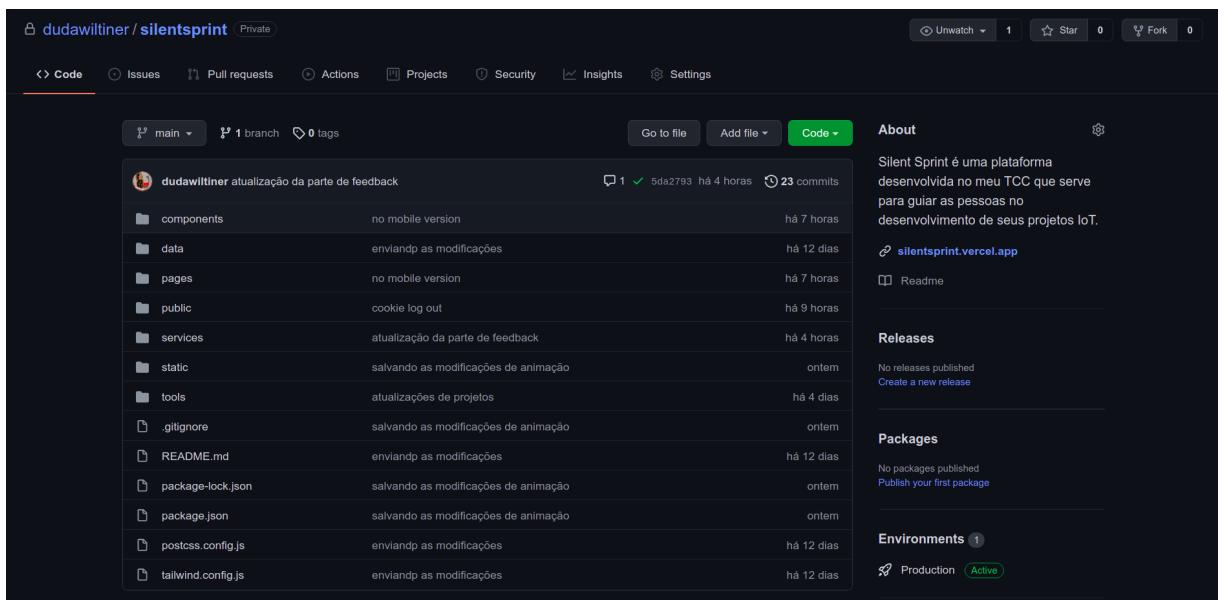


Fonte: autoria própria

## Etapa 1: Criação e clonagem de um repositório Git

Primeiramente foi criado e clonado localmente um repositório particular no GitHub, como mostrado na Figura 36, para controle de versionamento da aplicação *front-end* assim como foi feito para o *back-end*.

Figura 36. Repositório da aplicação *Silent Sprint* no GitHub



Fonte: autoria própria

Com o repositório clonado e assim, com a certeza de que agora o código desenvolvimento pode ser mantido e supervisionado com mais segurança, pode-se iniciar a preparação do ambiente para o desenvolvimento da plataforma.

## Etapa 2: Preparação do ambiente de desenvolvimento

Assim como no *back-end* na preparação de um ambiente de desenvolvimento, deve-se instalar as principais bibliotecas para que você consiga “rodar” e *debugar* o seu código e conseguir desenvolver a aplicação de acordo com a linguagem escolhida.

As bibliotecas instaladas são:

- **Tailwind.css:** uma biblioteca CSS com classes utilitárias que podem ser usadas para realizar a parte de estilização de qualquer projeto com muita facilidade. Pode possibilitar ao desenvolvedor chegar a não desenvolver nenhum arquivo CSS manualmente (TAILWINDCSS).
- **Cookie Cutter:** uma biblioteca importante para guardar dados do lado do cliente na forma de cookies e possibilita que através do token armazenado o

usuário não tenha que se autenticar novamente para acessar o conteúdo da plataforma (WUYTS, 2016).

Os frameworks instalados são:

- **React.js:** um framework JavaScript para construir uma UI (*user interface*). Ele ajuda na componentização do código, fazendo com que arquivos de códigos separados desempenhem funções específicas e independentes. O programa feito em React se torna mais previsível e de fácil manutenção (FACEBOOK OPEN SOURCE).
- **Next.js:** um framework baseado em React que consegue gerar algumas facilidades de criação de rotas para acesso a páginas e é quem possibilita a hospedagem de aplicações na plataforma Vercel (VERCEL).

### **Etapa 3: Desenvolvimento da plataforma e teste de funcionalidades**

No desenvolvimento da plataforma *Silent Sprint* foi utilizado muitos conhecimentos de *React* para programar toda estrutura aplicada, podendo através desta linguagem separar o código em pequenas partes para ajudar na manutenção, reaproveitamento do código e organização da aplicação, chama-se isso de “componentização do código”.

Primeiramente, foram criadas páginas estáticas sem interação, seguindo o *desgin Mockup* desenvolvido anteriormente, como por exemplo, botões sem funcionalidades, para apenas definir cada corpo de cada página e seus componentes. Logo depois foram criadas rotas para cada página ser acessada com uma url diferente, por exemplo, para a página home a url vai ter uma terminação de “/home” e para o login a url do site deverá ter uma terminação de “/login”, isso permite que o usuário possa dar um *reload* na página em que está sem sair da mesma e possa acessar um determinado recurso pela url. Caso uma rota não tenha sido definida no código, quando o usuário digitar a url na barra de busca ela vai gerar a página de erro 404.

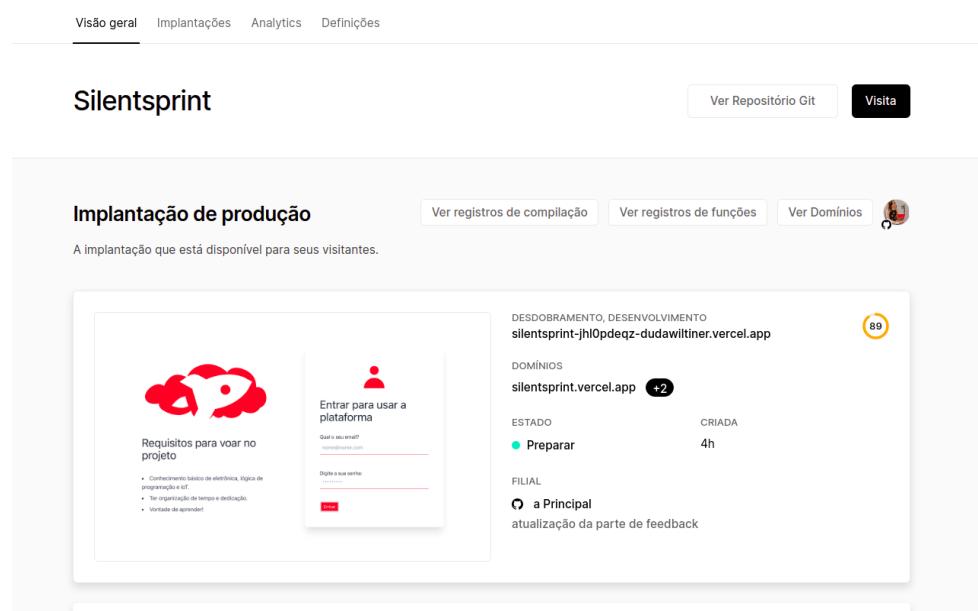
Para implementar as funcionalidades, foram usados alguns recursos específicos, como classes e funções do Next.js e do próprio React para criar ações dinâmicas, como um botão de enviar um *feedback* para ser registrado no banco de dados. Essas funcionalidades estão ligadas aos links ou rotas criadas para API *Silent Sprint* para realizarem requisições e por meio delas consultas ao banco de dados MongoDB hospedado no Atlas.

Por fim, para testar essa aplicação localmente, utilizou o comando do Gerenciador de Pacote Node (NPM), que ajuda na integração web com os arquivos desenvolvidos em React. Então, foi só “rodar” um “*npm run*” no terminal para conseguir enviar os arquivos para o navegador pelo servidor local (<http://localhost:3000>) para conseguirem ser interpretados e mostrados com o *design* planejado e estruturado pelo código.

## Etapa 4: Hospedagem da API

Para que quando a plataforma esteja no ar e consiga ser acessada por qualquer pessoa usando a internet ela precisa ser hospedada em uma plataforma que suporte o Next.js, ferramenta principal usada para o desenvolvimento da plataforma. Logo, a aplicação foi hospedada no Vercel, como mostrado na Figura 37, uma plataforma de hospedagem de aplicações feitas com o Next.js e é vista como o melhor lugar para implantar qualquer aplicativo de *front-end* para a construção de um site moderno. Ela oferece um serviço de conexão com o repositório do GitHub o que facilita na manutenção e no *deploy* do código desenvolvido (VERCEL).

Figura 37. Plataforma web *Silent Sprint* hospedado no Vercel



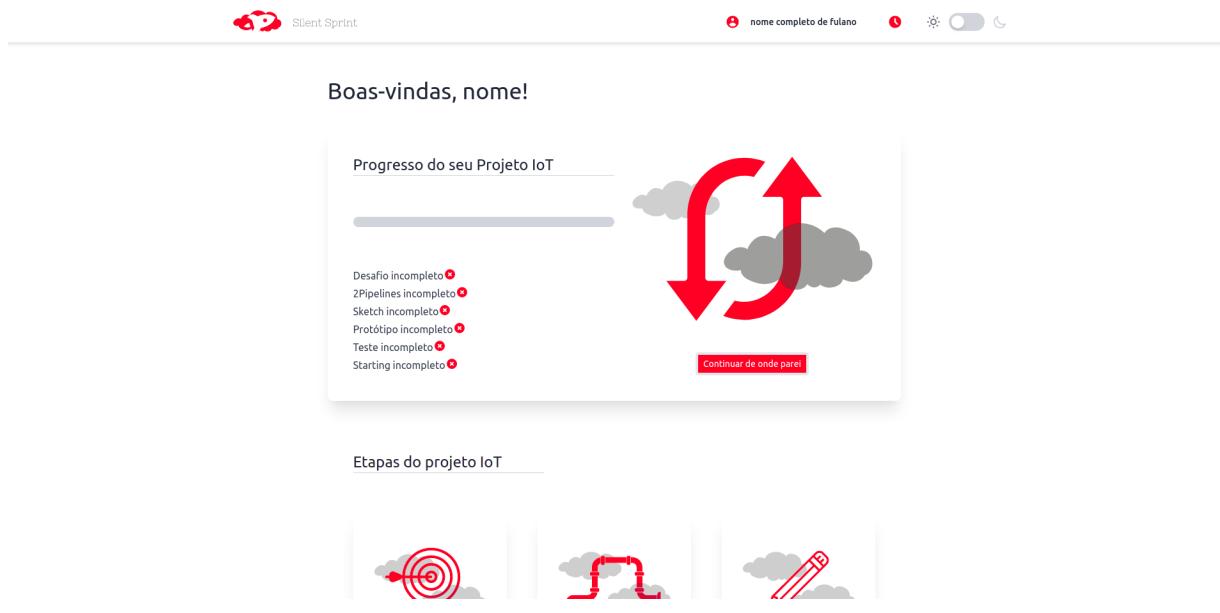
Fonte: autoria própria

## 5 RESULTADOS E DISCUSSÃO

O resultado do trabalho é a plataforma web hospedada que possui a principal função de compartilhar o conteúdo do processo de realização de um projeto empreendedor de IoT de forma ágil e bem guiada para permitir que as pessoas que a usam, criem produtos e serviços incríveis para a sociedade. A ferramenta, então, já está disponível na Internet e pode ser acessada pelo seguinte link: <https://silentsprint.vercel.app/>.

Apenas os usuários cadastrados, ou seja, com informações salvas no banco de dados podem ser autenticados na página de *login* e assim poderão ter acesso ao conteúdo do site *Silent Sprint*. A primeira página que os usuários autenticados vão poder interagir é a página *Home*, mostrada abaixo na Figura 38.

Figura 38. A página que aparece após o *login* do usuário



Fonte: autoria própria

## 6 CONCLUSÃO

A partir das minhas tentativas falhas de empreender com a Internet das Coisas, ainda na faculdade, pude perceber que tentar empreender com usa da tecnologia sempre vai ser um processo difícil e cansativo, porém, isso não significa que as metodologias não devam gerar uma experiência agradável e transmitir a segurança de que será gerado um produto incrível utilizando o processo focado no uso de uma tecnologia específica e emergente como a IoT para alavancar um negócio. Em plena Revolução 4.0 e com a Internet das Coisas se tornando um assunto cada vez mais discutido e aplicado em empresas tradicionais e modernas, o processo *Silent Sprint* pode contribuir com uma Metodologia Ágil aplicada e passos com questões importantes a serem discutidas até se chegar a um produto IoT de sucesso.

Com o ensino à distância se tornando cada vez mais popular, consequentemente aumentando o uso de plataformas educativas, disponibilizar o processo numa plataforma, se torna uma boa estratégia para se chegar a mais pessoas que possam criar coisas incríveis através do uso dessa ferramenta. Na plataforma *Silent Sprint* é combinado os assuntos empreendedorismo, tecnologia e gestão de projetos para descrever os passos imprescindíveis para se alcançar o sucesso do produto de IoT, ou seja, o que vai fazer com que o produto de IoT seja muito competitivo, que basicamente é atender o resultado esperado pelo cliente.

A visão de negócio e de tecnologia que a plataforma disponibiliza sobre projetos de IoT pode trazer vantagens para quem a usa e segue o processo ágil completamente, tais como, a redução de investimento de tempo na realização de projetos de Internet das Coisas que não vão sobreviver ao mercado, o conhecimento sobre a padronização e as vantagens geradas pelo uso de uma tecnologia emergente e o acompanhamento transparente do próprio avanço do projeto.

A próxima etapa do projeto será a etapa de “Validação da Ideia”, em que serão realizados testes da plataforma em ambiente universitário para garantir que haverá pessoas capacitadas e com os requisitos necessários para conseguir realizar o projeto de IoT guiado pela metodologia. A plataforma *Silent Sprint* será oferecida para ser utilizada por grupos de pessoas envolvidas com entidades e projetos que possuem o tema de empreendedorismo tecnológico de modo a validar a experiência do usuário e a eficiência do processo quando aplicado no desenvolvimento de produtos inteligentes.

## REFERÊNCIAS

- ABINC. **Associação Brasileira de Internet das Coisas**. Disponível em: <https://abinc.org.br/>. Acesso em: 4 set. 2021.
- ALEXANDRUK, M. **Modelagem em banco de dados**. Unilivros, São Paulo, SP, 2007 Disponível em: <https://unilivros.com.br/pdf/dbmod.pdf>. Acesso em: 20 nov. 2021.
- ASHTON, K.: *That “Internet of Things” Thing*. RFID Journal, 2009. Disponível em: <http://www.rfidjournal.com/articles/view?4986>. Acesso em: 20 out. 2021.
- AUTH0. Json Web Token Library. Disponível em: <https://jwt.io/libraries>. Acesso em: 14 set. 2021.
- BRASIL. **Decreto nº 9.854, de 25 de junho de 2019. Plano Nacional de Internet das Coisas e Dispõe Sobre A Câmara de Gestão e Acompanhamento do Desenvolvimento de Sistemas de Comunicação Máquina A Máquina e Internet das Coisas**. Brasília, 26 jun. 2019. Seção 1, p. 10. Disponível em: <https://www2.camara.leg.br/legin/fed/decret/2019/decreto-9854-25-junho-2019-788557-publicacaooriginal-158443-pe.html>. Acesso em: 20 out. 2020.
- CANVA. **Canva: Uma plataforma para criação de layouts**. Disponível em: <https://www.canva.com/>. Acesso em: 4 set. 2021.
- CHIAVEGATTO, F.; A. D. P. **Uso de big data em saúde no Brasil: perspectivas para um futuro próximo**. Epidemiologia e Serviços de Saúde, 24n 2, 2015.
- CODEIOT. **Code IOT: Uma plataforma para criação de objetos inteligentes**. Disponível em: <https://codeiot.org.br/>. Acesso em: 4 out. 2021.
- COUGHTER, P. *The Art of the Pitch: persuasion and presentation skills that win business*. 1nd ed. An Advertising Age Publication, 2012. 237 p.
- FACEBOOK OPEN SOURCE. *React.js: A JavaScript library for building user interfaces*. Disponível em: <https://reactjs.org/>. Acesso em: 22 ago. 2021.
- FREITAS, A. **A internet das coisas e seus efeitos na indústria 4.0**. Trabalho de conclusão de curso (Bacharelado em Sistemas de Computação) — Universidade Federal Fluminense, Rio de Janeiro, 2017.

- GABARDO, P. et al. **Estudo comparativo das plataformas de ensino-aprendizagem.** Encontros Bibli: Revista Eletrônica de Biblioteconomia e Ciência da Informação, n. (Especial), 2010.
- GAZIS, V. et al. *Short paper: IoT: challenges, projects, architectures.* In: INTERNATIONAL CONFERENCE ON INTELLIGENCE IN NEXT GENERATION NETWORKS, 18., 2015.
- GOODE, T. *A biblioteca cors.* GitHub, 2018. Disponível em: <https://github.com/expressjs/cors#readme>. Acesso em: 12 set. 2021.
- HEROKU. Heroku: Cloud application platform. Disponível em: <https://www.heroku.com/>. Acesso em: 18 set. 2021.
- HODA, R.; NOBLE, J.; MARSHALL S. *Agile Project Management.* ACADEMIA, Accelerating the world's research, Nova Zelândia, v. 2, p. 219-221, 2008.
- KAWAMOTO, Y.; HIROKI, N.; NEI, K.; NAOKO, Y.; SHINICHI, Y. *Internet of things (IoT): Present state and future prospects.* IEICE Transactions on Information and Systems, E97.D(10):2568–2575, 2014. xi, 59, 60
- KEERTIKUMAR, M.; SHUBHAM, M.; BANAKAR, R. *Evolution of IoT in smart vehicles: An overview.* International Conference on Green Computing and Internet of Things (ICGCloT), Delhi, India, 2015, pp 804-809.
- KELLMEREIT, D.; OBODOVSKI D.; *The Silent Intelligence: The Internet of Things.* 1nd ed. Dnd Ventures LLC, 2013. 156 p.
- KNAPP, J.; ZARETSKY, J.; KOWITZ B. *Sprint: O método usado no Google para testar e aplicar novas ideias em apenas cinco dias.* 1nd ed. Intrínseca, 2017. 318 p.
- LASI, H.; FETTKE, P.; KEMPER, H.; FELD, T.; HOFFMANN, M. *Industry 4.0. Business & Information Systems Engineering*, 6, 239-242, 2014.
- LIN, M.; CHEN, H.; LIU, K. *A Study of the Effects of Digital Learning on Learning Motivation and Learning Outcome.* Jurnal de Matemática, Ciência e Educação Tecnológica da Eurásia, 2017.
- LINUX FOUNDATION. *Node.js: A JavaScript runtime.* Disponível em: <https://nodejs.org/en/download/>. Acesso em: 21 ago. 2021.<https://nodejs.org/en/download/>
- LOPES, J. COR e LUZ. Instituto Superior Técnico, Universidade Técnica de Lisboa, Portugal, v. 4, p.1-47, 2013.

- MAGRANI, E. **A Internet das Coisas**. Ed. FGV, 2018, São Paulo, SP.
- MICROSOFT. **Visual Studio Code: Code Editing Software**. Versão 1.60, 25 mai. 2018. Disponível em: <https://code.visualstudio.com/download>. Acesso em: 01 set. 2021.
- MONGODB ATLAS. MongoDB Atlas: **The most advanced cloud database service on the market**. Disponível em: <https://www.mongodb.com/cloud/atlas>. Acesso em: 17 set. 2021.
- MONGODB, INC. **MongoDB NodeJS Driver**. Disponível em: <https://mongodb.github.io/node-mongodb-native/>. Acesso em: 14 set. 2021.
- MONGODB. MongoDB: Sistema de Gerenciamento de Banco de Dados NoSQL. Disponível em: <https://www.mongodb.com/pt-br>. Acesso em: 19 set. 2021.
- MOTTE, S. **A biblioteca dontev**. GitHub, 2013. Disponível em: <https://github.com/motdotla/dotenv#readme>. Acesso em: 12 set. 2021.
- NETSCAPE. **JavaScript: Uma linguagem de programação compilada**. Disponível em: <https://developer.mozilla.org/en-US/docs/Web/JavaScript>. Acesso em: 21 set. 2021.
- NUCCIO, D. D.; WOLLHEIM, B. **Nasce um empreendedor**. 1nd ed. Portfolio Penguin, 2016. 177 p.
- PALLAVI, S.; SMRUTI, S. **Internet of Things: Architectures, Protocols, and Applications, Journal of Electrical and Computer Engineering**, vol. 2017, Article ID 9324035, 25 pages, 2017
- PEREIRA, A.; SIMONETTO, E. de O. **Indústria 4.0: conceitos e perspectivas para o Brasil**. Revista da Universidade Vale do Rio Verde, v. 16, n. 1, jan./jul 2018. Disponível em: <http://periodicos.unincor.br/index.php/revistaunincor/article/view/4938>. Acesso em: 27 out. 2021.
- PIPINELLIS, A. **GitHub Essebtials: Unleash the power of collaborative development workflows using GitHub**. 2nd ed. Packt, 2018. 158 p.
- POSTMAN. **Plataforma: The API platform**. Disponível em: <https://www.postman.com/downloads/>. Acesso em: 10 set. 2021.
- RESTFULAPI. **O que é REST?** Disponível em: <https://restfulapi.net/>. Acesso em: 7 set. 2021.
- ROCKSET. **Git & Github: O que é? Por que? Como iniciar?** Disponível em: <https://blog.rocketseat.com.br/iniciando-com-git-github/>. Acesso em: 10 set. 2021.

SEIBERT MEDIA. Draw.io: Diagrams for Confluence and Jiara. Disponível em: <https://drawio-app.com/>. Acesso em: 10 set. 2021.

SINCLAIR, B. **IoT: Como usar a “Internet das Coisas” para alavancar seus negócios.** 1nd ed. Autêntica Business, 2018. 262 p.

STEVAN, S. **Internet das Coisas: Fundamentos e aplicações em Arduino e NodeMCU.** 1nd ed. Casa do Código, 2018. 223 p.

STRONGLOOP. *Express: Fast, unopinionated, minimalist web framework for Node.js.* Disponível em: <https://expressjs.com/> Acesso em: 16 set. 2021.

SURESH, P. e colab. *A state-of-the-art review on the Internet of Things (IoT) history, technology and fields of deployment.* 2014 International Conference on Science Engineering and Management Research (ICSEMR), p. 1–8, 2014.

Santos, B. P.; Silva, L.; Celes, C.; Borges, J. B.; Neto, B. S. P.; Vieira, M. A. M.; Vieira, L. F. M.; Goussevskaia, O. N.; Loureiro, A. **Internet das coisas: da teoria à prática.** Minicursos SBRC-Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos, 2016.

TAILWINDCSS. *Tailwind.css: A CSS Library.* Disponível em: <https://tailwindcss.com/>. Acesso em: 14 set. 2021.

TAURION, C. **Cloud Computing: Computação em Nuvem, transformando o mundo da Tecnologia da Informação.** 1nd ed. Brasport Livros e Multimídia Ltda, 2009. 195 p.

TEIXEIRA, F. Introdução e boas práticas em UX Design. 1nd ed. Casa do Código, 2017. 263 p.

TIC DOMICILIO. **Tecnologia da Informação e Comunicação de Domicílio.** Disponível em: <https://cetic.br/pesquisa/domicilios/>. Acesso em: 4 set. 2021.

TOYOFUKU, A. **Uma visão sistêmica de um Projeto IoT.** Disponível em <https://www.embarcados.com.br/projeto-iot/>. Acesso em 22 set. 2021.

VERCEL. **Next.js: The React Framework.** Disponível em: <https://nextjs.org/learn/basics/create-nextjs-app>. Acesso em: 21 ago. 2021.

VERCEL. Vercel: An application hosting platform. Disponível em: <https://vercel.com/>. Acesso em: 22 set. 2021.

WILCOX, J. ***Customer Discovery: What Do You Ask, with Justin Wilcox***. 2017. (6m59s). Disponível em: <<https://www.youtube.com/watch?v=OTkP2JDeGWM>>. Acesso em: 11 set. 2021.

WILSON, D. A **biblioteca body-parser**. GitHub, 20114. Disponível em: <https://github.com/expressjs/body-parser#readmee>. Acesso em: 12 set. 2021.

WUYTS, Y. A **biblioteca cookie-cutter**. GitHub, 2016. Disponível em: <https://github.com/substack/cookie-cutter>. Acesso em: 14 set. 2021.