

UNIVERSIDADE FEDERAL DE JUIZ DE FORA
CURSO DE GRADUAÇÃO EM ENGENHARIA DE PRODUÇÃO

GABRIEL D'AGOSTO PACHE DE FARIA

**DESENVOLVIMENTO DE UM PRODUTO MÍNIMO VIÁVEL DE UM
SOFTWARE DE GESTÃO DE EVENTOS ATRAVÉS DA METODOLOGIA
*SCRUM***

JUIZ DE FORA

2021

GABRIEL D'AGOSTO PACHE DE FARIA

**DESENVOLVIMENTO DE UM PRODUTO MÍNIMO VIÁVEL DE UM
SOFTWARE DE GESTÃO DE EVENTOS ATRAVÉS DA METODOLOGIA
*SCRUM***

Trabalho de Conclusão de Curso apresentado
a Faculdade de Engenharia da Universidade
Federal de Juiz de Fora, como requisito
parcial para a obtenção do título de
Engenheiro de Produção.

Orientador: DSc Roberta Cavalcanti Pereira Nunes

JUIZ DE FORA

2021

Ficha catalográfica elaborada através do programa de geração automática da Biblioteca Universitária da UFJF, com os dados fornecidos pelo(a) autor(a)

d'Agosto Pache de Faria, Gabriel.

Desenvolvimento de um Produto Mínimo de um software de gestão de eventos através da metodologia *Scrum* / Gabriel d'Agosto Pache de Faria. -- 2021.

47 p.

Orientadora: Robera Cavalcanti Pereira Nunes

Trabalho de Conclusão de Curso (graduação) - Universidade Federal de Juiz de Fora, Faculdade de Engenharia, 2021.

1. Eventos. 2. MVP. 3. Protótipo. 4. Aplicativo. I. Cavalcanti Pereira Nunes, Robera, orient. II. Título.

GABRIEL D'AGOSTO PACHE DE FARIA

**DESENVOLVIMENTO DE UM PRODUTO MÍNIMO VIÁVEL DE UM
SOFTWARE DE GESTÃO DE EVENTOS ATRAVÉS DA METODOLOGIA
*SCRUM***

Trabalho de Conclusão de Curso apresentado
a Faculdade de Engenharia da Universidade
Federal de Juiz de Fora, como requisito
parcial para a obtenção do título de
Engenheiro de Produção.

Aprovada em 15 de março de 2021.

BANCA EXAMINADORA



Prof. DSc. Roberta Cavalcanti Pereira Nunes
Universidade Federal de Juiz de Fora



pp

Prof. DSc. Antônio Ângelo Missaggia Picorone
Universidade Federal de Juiz de Fora



pp

Prof. DSc. Romir Soares de Souza Filho
Universidade Federal de Juiz de Fora

AGRADECIMENTOS

Dedico este trabalho à minha família. A meus pais, que me educaram e me trataram desde sempre com a maturidade e respeito que qualquer jovem sonha em conquistar. Que sempre batalharam muito para que eu tivesse as melhores oportunidades e trilhasse o melhor caminho possível. À minha mãe Daniela e ao meu pai, Wladimir, obrigado.

Agradeço também meus avós, Vera e José, que me mostraram um mundo sob uma ótica diferente em todos os finais de semana em que passamos juntos.

Agradeço também a meus amigos. Pois foram os irmãos que, não são irmãos por mera casualidade de nascer na mesma família, mas irmãos que eu escolhi para a vida.

E a minha namorada, Flavia, que sempre foi paciente, compreensiva, e que me mantém são e calmo nos momentos mais difíceis. Obrigado por estar sempre ao meu lado.

Por fim, gostaria de agradecer a todos aqueles que mudaram minha forma de ver um mundo, pelo menor que fosse. O convívio, respeitoso, com pessoas de diferentes crenças, com diferentes convicções e diferentes maneiras de enxergar o mundo me tornou uma pessoa melhor.

RESUMO

O presente trabalho trata do desenvolvimento de um Produto Mínimo Viável (MVP) de um software de gestão de eventos no Brasil, a partir de um escopo de projeto previamente desenvolvido com base na metodologia Scrum. O trabalho versa sobre as tecnologias utilizadas no desenvolvimento do MVP, escolhidas a partir de dados extraídos de literatura pertinente ao tema, e apresenta os passos que foram tomados durante o desenvolvimento, expondo as dificuldades e facilidades encontradas em cada um dos passos da execução do MVP. Ao final do trabalho, pretende-se não só terminar o desenvolvimento do MVP, mas também entender as dificuldades de um desenvolvimento de uma primeira versão de um produto real, dentro de uma organização, e as implicações que as decisões tomadas ao longo da elaboração do escopo e da seleção das tecnologias podem causar no resultado do Produto Mínimo Viável.

Palavras-chave: Eventos, MVP, Protótipo, Aplicativo.

ABSTRACT

The present work aims to showcase the development of a Minimum Viable Product (MVP) of an events management software. The software specifications are taken from a previously constructed Scrum based project scope. This work talks about the technologies utilized during the development of the MVP, chosen based on data extracted from available literature of the subject. It also presents the steps taken during development, illustrating both the challenges and the easy parts found in each step of the MVP development. By the end of this work, the objective is not only to finish developing the MVP, but to also understand the challenges that are present during development of an early version of a real product, inside an organization, and the implications that decisions made during formulation of the project's scope and technology selection can make on the MVP result.

Keywords: Events, MVP, Prototype, Application

LISTA DE FIGURAS

Figura 1 – <i>Outras Bibliotecas, Ferramentas e Arcabouços mais populares no mercado ...</i>	21
Figura 2 – <i>Bancos de dados mais populares no mercado</i>	22
Figura 3 – <i>Arcabouços mais desejados por desenvolvedores</i>	25
Figura 6– <i>Landing Page</i>	33
Figura 7 – <i>Telas de cadastro de usuário e fornecedor.....</i>	34
Figura 8 – <i>Protótipo de Mapa interativo.....</i>	36
Figura 9 – <i>Cadastro de Eventos</i>	38
Figura 10 – <i>Agendamento Proposto</i>	39
Figura 11 – <i>Mapa Interativo Com Fornecedores.....</i>	40
Figura 12 – <i>Burndown Chart do MVP</i>	45

LISTA DE TABELAS

Tabela 1 – <i>Requisitos de um Arcabouço</i>	24
Tabela 2 – <i>Sprint Backlog 1</i>	29
Tabela 3 – <i>Sprint Backlog 1 Revisado</i>	30
Tabela 4 – <i>Acompanhamento do Sprint 1</i>	31
Tabela 5 – <i>Sprint Backlog 2</i>	35
Tabela 6 – <i>Sprint Backlog 2 Revisado</i>	36
Tabela 7 – <i>Acompanhamento do Sprint 2</i>	37
Tabela 8 – <i>Sprint Backlog 3</i>	41
Tabela 9 – <i>Sprint Backlog 3 Revisado</i>	41
Tabela 10 – <i>Acompanhamento do Sprint 3</i>	42

LISTA DE ABREVIATURAS, SIGLAS E SÍMBOLOS

ABEOC – Associação Brasileira de Eventos

MVP – *Minimum Viable Product* (Produto Mínimo Viável)

SUMÁRIO

1. INTRODUÇÃO	12
1.1 CONSIDERAÇÕES INICIAIS	12
1.2 JUSTIFICATIVA	13
1.3 ESCOPO DO TRABALHO.....	13
1.4 ELABORAÇÃO DOS OBJETIVOS	14
1.5 DEFINIÇÃO DA METODOLOGIA.....	14
1.6 ESTRUTURA DO TRABALHO.....	15
2. REVISÃO BIBLIOGRÁFICA.....	16
2.1 PRODUTO MÍNIMO VIÁVEL (MVP).....	16
2.2 SCRUM.....	18
2.3 ARQUITETURA DE SOFTWARE	19
2.3.1 CAMADA DE APRESENTAÇÃO	20
2.3.2 CAMADA DE NEGÓCIO E CAMADA DE ACESSO AOS DADOS	20
2.3.3 BANCO DE DADOS	22
2.4 ARCABOUÇOS	23
3. DESENVOLVIMENTO	27
3.1 A EMPRESA E O PRODUTO	27
3.2 O ESCOPO	28
3.2.1 REVISÃO DO ESCOPO.....	28
3.3 DESENVOLVIMENTO DO MVP.....	28
3.3.1 SPRINT 1	29
3.3.1.1 RESULTADOS DO SPRINT 1.....	30

3.3.1.2	ITEM 1 – SELEÇÃO DAS TECNOLOGIAS	31
3.3.1.3	ITEM 2 – CONFIGURAR O AMBIENTE DE DESENVOLVIMENTO	32
3.3.1.4	ITEM 3 – LANDING PAGE.....	32
3.3.1.5	ITENS 4 E 5 – CADASTRO DE USUÁRIO E DE FORNECEDOR	33
3.3.2	SPRINT 2	35
3.3.2.1	RESULTADOS DO SPRINT 2.....	37
3.3.2.2	ITEM 6 – FUNCIONALIDADES DE LOGIN	37
3.3.2.3	ITEM 7 – CADASTRO DE EVENTOS	38
3.3.2.4	ITEM 8 – AGENDAMENTO DE FORNECEDORES.....	39
3.3.2.5	ITEM 9 – MAPA INTERATIVO COM FORNECEDORES	40
3.3.3	SPRINT 3	41
3.3.3.1	RESULTADOS DO SPRINT 3.....	42
3.3.3.2	ITEM 10 – FUNCIONALIDADES DE AVALIAÇÃO	42
3.3.3.3	ITEM 11 – ESTILIZAÇÃO DO WEBSITE	43
4.	<u>RESULTADOS E DISCUSSÕES</u>	<u>44</u>
4.1	RESULTADOS ALCANÇADOS	44
4.2	PRÓXIMOS PASSOS	45
5.	<u>CONCLUSÃO</u>	<u>47</u>
	<u>REFERÊNCIAS.....</u>	<u>48</u>

1. INTRODUÇÃO

1.1 CONSIDERAÇÕES INICIAIS

O último dimensionamento de mercado feito pela Associação Brasileira de Empresas de Eventos (ABEOC), em 2013, mostra o panorama do mercado de eventos no Brasil. Segundo a pesquisa, que analisou 2784 empresas organizadoras de eventos, foram realizados 590.913 eventos no Brasil em 2013. Em média, cada evento contou com 342,1 participantes, totalizando 202.171.787 participantes de eventos, em apenas 1 ano. (ABEOC, 2013).

Em sua pesquisa própria de mercado própria com 53 organizadores de eventos de diversos países, Oliveira (2019) mostra que, dentre os entrevistados, que, em sua maioria, realizam de 2 a 3 eventos por ano e não trabalham em uma empresa organizadora de eventos, existem dificuldades relacionadas a infraestrutura e atrações. A pesquisa também mostra que 81,8% dos entrevistados pagariam por um serviço que os auxiliassem na realização de seus eventos.

Uma segunda pesquisa de mercado, realizada com 149 participantes de 12 países diferentes, apresentada por Oliveira (2019), mostra que, dentro os entrevistados, 62,8% acredita que um aumento no número de eventos em sua cidade contribuiria para um aumento no número de turistas.

Combinando os resultados das pesquisas de mercado própria com a discrepância entre o pequeno número de empresas organizadoras de eventos em relação ao número de eventos realizados no ano, Oliveira (2019) e o autor do presente trabalho identificaram uma oportunidade de desenvolver um software na área de organização e gestão de eventos, que auxiliaria organizadores de eventos a encontrarem fornecedores de serviços para seus eventos. Oliveira (2019), em seu trabalho "GERENCIAMENTO DE PROJETOS PARA O DESENVOLVIMENTO DE UM SOFTWARE NA ÁREA DE EVENTOS", desenvolveu então um escopo do projeto de software. O presente trabalho dá continuidade ao projeto de Oliveira (2019), desenvolvendo um Produto Mínimo Viável (MVP – *Minimum Viable Product*) a partir do escopo definido, que será reapresentado no presente trabalho.

1.2 JUSTIFICATIVA

Ao participar de uma competição de empreendedorismo no segundo semestre de 2018, o grupo de trabalho de desenvolvimento, entre eles Oliveira (2019) e o autor do presente trabalho observaram uma oportunidade de negócio no segmento de eventos. Porém, a falta de know-how na aplicação de metodologias de gerenciamento foi um fator crítico para diversos problemas que surgiram ao longo do percurso. Não havia um plano de negócio, uma direção a ser seguida. Não havia um cronograma de entregas, e as atividades eram desenvolvidas conforme a disponibilidade de cada membro.

Souza (2014) explica que a falta de organização pode gerar problemas de comunicação entre os membros de uma equipe. A falta de organização, por sua vez, pode gerar tanto desmotivação entre os membros de uma equipe, uma vez que não possuem canal fácil para apresentação das conquistas e dificuldades, quanto retrabalho, uma vez que os requisitos do projeto podem mudar constantemente durante a fase inicial. Além disso, a dificuldade em encontrar essas informações pode resultar em tempo gasto desenvolvendo componentes que já não fazem mais parte do escopo.

Uma vez que a equipe esteve de acordo com a implementação da metodologia Scrum de desenvolvimento ágil, Oliveira (2019) desenvolveu o escopo de um projeto para o desenvolvimento de um software para a área de eventos.

Após a definição do escopo do software, o próximo passo definido pela equipe composta por Oliveira (2019) e o autor do presente trabalho é o desenvolvimento de um MVP. Abrahamsson (2016) define um MVP como um produto que possui apenas funcionalidades suficientes para garantir aprendizado validado sobre os produtos. De acordo com RIES (2011), toda empresa iniciante deveria começar construindo um MVP, e usando-o para validar suas hipóteses sobre as necessidades do consumidor.

Para a disseminação do produto, Enge (2019), em um estudo com 1 trilhão de acessos a sites em 2018, mostra que 58% dos acessos e 42% do tempo gasto em sites vieram de dispositivos móveis. Faz-se necessário levar em consideração aparelhos móveis na tomada de decisão da construção do MVP.

1.3 ESCOPO DO TRABALHO

Todos os anos, o site StackOverflow (2019) realiza uma pesquisa anual com desenvolvedores de software. Segundo a pesquisa, o "StackOverflow Developer Survey

Results" é um dos maiores questionários de pessoas que programam ao redor do mundo, e cerca de 90.000 desenvolvedores responderam o questionário. O presente trabalho utiliza essa pesquisa como ponto de partida para a avaliação de tecnologias e Arcabouços populares no mercado.

De acordo com o site institucional da ABEPRO (2019), o presente trabalho se enquadra na área de Engenharia do Produto, uma vez que trata do processo de desenvolvimento do produto, listado como a subárea 5.2 da engenharia de produção. Para a ABEPRO, Engenharia do Produto é também as "ferramentas e processos [...] de decisão e execução envolvidas nas atividades estratégicas e operacionais de desenvolvimento de novos produtos."

Ao final deste trabalho, pretende-se apresentar o resultado do desenvolvimento de um produto mínimo viável de um software para web, que conectará fornecedores de serviços a organizadores de eventos, a partir das definições de escopo feitas por Oliveira (2019).

1.4 ELABORAÇÃO DOS OBJETIVOS

O trabalho tem como objetivos ilustrar o estado da arte das tecnologias disponíveis para desenvolvimento de aplicativos e soluções web, revisar o escopo proposto por Oliveira (2019), e desenvolver um produto mínimo viável seguindo o escopo revisado, avaliando as dificuldades e adaptações necessária, e discutir lições que podemos aprender com a tentativa de implementar a metodologia *Scrum* no desenvolvimento do MVP.

O objetivo acadêmico do trabalho também é acompanhar as diferenças entre as métricas do escopo desenvolvido por Oliveira (2019) e sua aplicação prática, observando as definições de cada etapa e os tempos necessários para suas realizações.

1.5 DEFINIÇÃO DA METODOLOGIA

Alves (2013) define que a metodologia pode ser entendida como uma maneira de conduzir uma pesquisa. A metodologia deve ser observada como as habilidades e conhecimentos gerais que o pesquisador precisa ter para a utilização de um método de pesquisa. Já o método é o caminho a ser trilhado, a partir de uma lista de atividades sistemáticas, para que se possa atingir o objetivo do trabalho proposto.

Para o desenvolvimento do presente trabalho, os passos que foram seguidos são descritos abaixo:

1. Revisão bibliográfica sobre as principais tecnologias disponíveis para o desenvolvimento de aplicativos e soluções web. A revisão teve como fontes a literatura disponível, como livros, artigos e publicações, tanto brasileiras quanto estrangeiras;
 - a. Quanto ao tema de Produto Mínimo Viável, os principais artigos utilizados como referência são Abrahamsson & Duc (2016) e Ries (2011);
 - b. Acerca de arquitetura de software, utiliza-se o artigo de Smith (2017) como a base, e os artigos de Keith (2006), Mao (2018) e Xiaojie (2011) para itens específicos definidos a partir do artigo base.
2. Revisão dos conjuntos de ferramentas disponíveis no mercado para a aplicação das tecnologias relevantes para o escopo. Os principais autores são Abrahamsson & Graziotin (2016), que mostram que o número de arcabouços disponíveis atualmente já passa dos milhares. Portanto, utilizou-se a pesquisa do site Stackoverflow (2019) para selecionar os arcabouços mais populares e fazer uma revisão bibliográfica dos mesmos. Além disso, Wohlgethan (2018) faz um comparativo entre os arcabouços mais populares da pesquisa do Stackoverflow (2019), mostrando os pontos positivos e negativos de cada arcabouço.
3. Revisão do escopo de software definido por Oliveira (2019), com enfoque em observar os tipos de tecnologias que serão necessários para o desenvolvimento do trabalho;
4. Desenvolvimento de um produto mínimo viável, pautado no escopo definido por Oliveira (2019), e nas tecnologias e arcabouços selecionados durante a revisão bibliográfica;

1.6 ESTRUTURA DO TRABALHO

O primeiro capítulo do presente trabalho apresenta as considerações iniciais a respeito do tema a ser estudado. Também traz as motivações e objetivos do autor na realização do trabalho, além da metodologia a ser empregada para alcançar tais objetivos.

O segundo capítulo faz uma revisão da literatura disponível sobre os temas mais relevantes do trabalho. Neste capítulo, aborda-se os conceitos de MVP, seus tipos, as diversas tecnologias e arcabouços disponíveis para o desenvolvimento de software, além de gerenciamento de projetos através da metodologia *Scrum*. Esse capítulo visa dar embasamento teórico aos assuntos a serem discutidos nos capítulos subsequentes.

O terceiro capítulo trata do desenvolvimento do projeto. São apresentados o objetivo, o escopo proposto por Oliveira (2019) e sua revisão e o processo de desenvolvimento do MVP.

O quarto capítulo apresenta o MVP concluído. Neste capítulo, discute-se as diferenças entre o que foi definido no escopo do software revisado e os atrasos e adiantamentos que aconteceram durante o desenvolvimento. E o quinto capítulo apresenta as considerações finais do autor acerca do trabalho realizado, discutindo as diferenças entre as recomendações teóricas e as alterações necessárias no escopo durante a prática.

2. REVISÃO BIBLIOGRÁFICA

O presente capítulo, a partir de revisão da literatura, discorre sobre o conceito de Produto Mínimo Viável (MVP), e os diversos tipos de MVP, discutindo as motivações para a escolha de cada possibilidade, além das vantagens e desvantagens de cada escolha.

Posteriormente, é exposto qual o estado da arte em termos de desenvolvimento de aplicações para internet. Abordam-se os temas de engenharia de software para a internet, tecnologias a serem utilizadas e os arcabouços disponíveis para facilitar o desenvolvimento de um software desenvolvido nas referidas tecnologias.

Este capítulo também tem o objetivo de justificar as decisões do autor a respeito das tecnologias e arcabouços que foram escolhidos para o prosseguimento do presente trabalho.

Por fim, o capítulo também apresenta uma introdução à Metodologia Scrum, tratando da definição da metodologia, da sua aplicação e das etapas que compõem seu ciclo de desenvolvimento.

2.1 PRODUTO MÍNIMO VIÁVEL (MVP)

Ao longo da literatura, existem diversos tipos de definição a respeito do que é um MVP. Lenarduzzi & Taibi (2016), em seu trabalho, elaboram uma revisão completa a respeito da definição de MVP, fazendo uma revisão dos termos mais utilizados para definir um MVP. De acordo com eles, entre as definições clássicas de o que é um Produto Mínimo Viável, a mais popular ao longo de publicações acadêmicas é a de Ries (2011).

Ries (2011), define um MVP como uma versão de um novo produto, que permite que times coletem a maior quantidade de aprendizado validado sobre seus consumidores, empregando o mínimo esforço. Ou seja, um MVP é um produto simples, visando adquirir mais conhecimento do consumidor acerca de quais necessidades precisam ser atendidas. Um bom MVP é aquele que dará possibilidade aos desenvolvedores de refinarem sua ideia de produto inicial, permitindo que os mesmos desenvolvam um produto final melhor.

Abrahamsson & Duc (2016) defendem que, no contexto de startups, os termos MVP e protótipo são utilizados de maneira intercambiável, e apresentam benefícios que empresas encontraram ao utilizar um MVP. Dentre os objetivos, encontram-se a visualização da ideia inicial de design, a capacidade de criar pontes de comunicação entre diversos *stakeholders* e a capacidade de entender divergências entre as expectativas do usuário e o produto.

Ries (2011), define diversos tipos de MVP. São eles:

1. Vídeo explicativo: Uma animação curta que explica o que seu produto faz e porque usuários deveriam comprá-lo. O vídeo é muitas vezes simples e dura entre 30 segundos e alguns minutos;
2. *Landing Page*: Uma página em que visitantes chegam após clicar um link em um e-mail ou outro tipo de campanha. Uma *landing page* é utilizada para comunicar rapidamente as propostas da startup, difundir objeções e para realizar chamadas de ação;
3. Mágico de Oz: Uma interface de usuário que parece um produto real funcional, mas o processo real é feito manualmente. O propósito desse MVP é demonstrar o trabalho completo feito pelo produto;
4. *Concierge MVP*: Um serviço manual que consiste nos mesmos passos que um usuário final iria realizar no produto final;
5. MVP fragmentado: Similar ao MVP do Mágico de Oz, mas a execução das tarefas é feita a partir de ferramentas já existentes;
6. MVP maquete: Representação da interface de usuário sem qualquer funcionalidade.
7. Proposta pública de projeto: Plataformas de *crowdsourcing* que permitem que usuários comprem o produto antecipadamente, promovendo uma ótima maneira de levantar dinheiro para as necessidades iniciais;
8. MVP de função única: Um protótipo que implementa a função mais importante do produto;
9. MVP *rip-off*: Um produto simples feito com o intuito de conseguir feedback, para depois pivotar-se para uma direção diferente.

Abrahamsson & Duc (2016) mostram que não é preciso ater-se a apenas um tipo de MVP. Nos estudos de caso do trabalho de Abrahamsson & Duc, companhias utilizam mais de um tipo de MVP. Todas as companhias avaliadas utilizaram os MVPs de tipo *landing page* e maquete, enquanto que 4 a cada 5 companhias avaliadas utilizaram o MVP de função única durante seu processo de desenvolvimento.

Para o presente trabalho, julgou-se que seriam utilizados os MVPs *Landing Page* e MVP de função única. Para o software de gestão de eventos, a função mais importante é

permitir organizadores de eventos consigam se conectar com fornecedores e agendar o fornecimento para seus eventos.

2.2 SCRUM

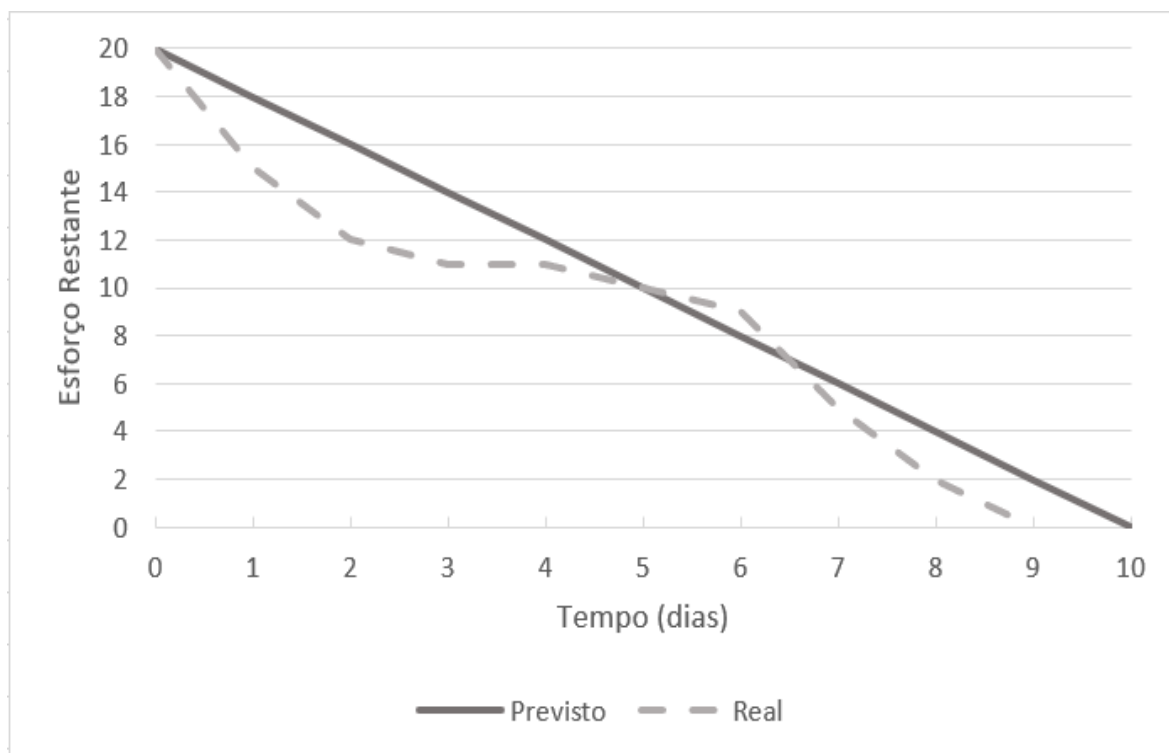
Embora não seja o foco do presente trabalho detalhar como funciona a metodologia *Scrum*, alguns termos mostram-se importantes para o entendimento do projeto.

Sutherland (2012) define que o *Scrum* é uma metodologia iterativa e incremental de desenvolvimento de projetos, produtos ou aplicações. Um *Sprint* é um ciclo de desenvolvimento da metodologia *Scrum*. Cada *Sprint* tem duração de até um mês, e devem ser determinados por um período de tempo imutável. Um *Sprint* não pode ser estendido. Caso as tarefas propostas para um *Sprint* não sejam concluídas ao final do tempo proposto, elas devem ser colocadas no próximo *Sprint*. Um *Sprint* deve ser iniciado após o término do *Sprint* que o antecede, sem pausas.

Sutherland (2012) também define que um *Sprint Backlog* é um documento que define todas as atividades a serem realizadas durante o *Sprint*. O *Sprint Backlog* descreve a tarefa a ser realizada, o membro da equipe responsável pela realização da tarefa e a estimativa de tempo para que a atividade seja concluída.

Por fim, Sutherland (2012) apresenta o *Burndown Chart*, gráfico importante que possibilita uma visualização do ritmo de desenvolvimento do projeto. Segundo SBROCCO (2012), o *Burndown Chart* é um gráfico cujo eixo vertical representa o esforço despendido, e o eixo horizontal representa o tempo. São traçadas uma linha contínua, que representa a quantidade planejada de esforço despendido por tempo de projeto, e uma linha tracejada, que representa o fluxo real de esforço no tempo. Se a linha tracejada estiver abaixo da linha contínua, a equipe estará então superando as expectativas do projeto. Caso contrário, a equipe está atrasada. Um exemplo de *Burndown Chart* pode ser observado na figura 4.

Figura 4 - Exemplo de Burndown Chart



Fonte: o Autor

2.3 ARQUITETURA DE SOFTWARE

Aplicativos Web modernos devem ser capazes de atender uma demanda cada vez maior, com usuários cada vez mais exigentes. Dado as necessidades atuais, tais aplicativos tornam-se cada vez mais complexos, com a necessidade de serem tanto flexíveis, quanto escalonáveis, para serem capazes de atender aos altos picos de demanda (SMITH, 2017).

Com aplicativos cada vez mais complexos, uma maneira de lidar com toda essa complexidade de maneira mais fácil é dividir o aplicativo em camadas. Uma arquitetura de software é tipicamente composta por três camadas: A camada de apresentação, a camada de negócio e a camada de acesso aos dados (SMITH, 2017).

A camada de apresentação provém a interface com o usuário final (UI – *User Interface*). Na camada de negócio ficam contidas as regras de negócio da aplicação. E por fim, a camada de acesso aos dados permite o acesso a sistemas externos como banco de dados (SMITH, 2017).

2.3.1 CAMADA DE APRESENTAÇÃO

Na camada de apresentação, JavaScript (JS) é a linguagem de programação a ser utilizada. JavaScript é necessário pois é a linguagem que os navegadores modernos, de computadores, *tablets* e aparelhos móveis, são capazes de interpretar. JavaScript é necessário para especificar o comportamento das páginas na internet, sendo assim utilizado na grande maioria dos sites da atualidade (FLANAGAN, 2004).

JavaScript é uma linguagem que trouxe a possibilidade de tornar páginas da internet mais interativas. Anteriormente, navegadores de internet eram softwares relativamente simples, com capacidade de mostrar documentos de texto com links. Já com o JavaScript, tornou-se possível manipular as páginas, tornando-as dinâmicas. A linguagem foi desenvolvida originalmente pela Netscape, e sua primeira versão estava disponível para o navegador de internet Netscape Navigator 2, em setembro de 1995 (KEITH, 2006).

Antes do surgimento do JavaScript, o Netscape já era o navegador mais utilizado no mundo. Em uma pesquisa realizada por Berghel (1995), 54% dos respondentes utilizavam o navegador Netscape. Já em 1996, uma nova pesquisa, realizada pelo Centro de Gráficos, Visualização e Usabilidade da Escola de Computação do Instituto Tecnológico da Geórgia, mostra o impacto do JavaScript. O estudo continha a pergunta “que navegador de internet você se veria utilizando daqui a 1 ano?” 89,4% dos respondentes disseram que estariam utilizando o Netscape. (GEORGIA TECH COLLEGE OF COMPUTING, 1996)

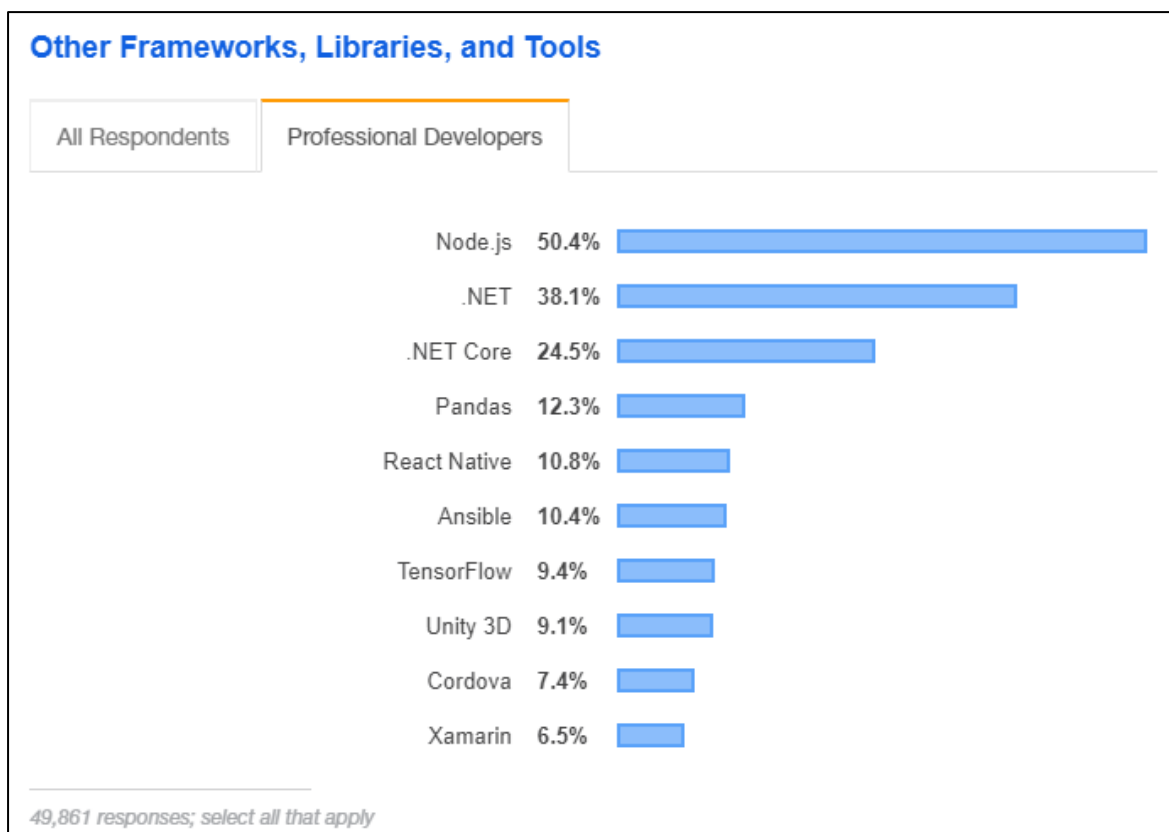
Nos anos que se seguiram, o JavaScript foi padronizado, junto à Associação Europeia dos Fabricantes de Computador (ECMA). Surgiu assim o ECMAScript, versão da linguagem JavaScript que seria utilizada por todos os navegadores de internet. Contudo, o nome não vingou, e continuamos nos referindo ao ECMAScript como JavaScript (KEITH, 2006).

2.3.2 CAMADA DE NEGÓCIO E CAMADA DE ACESSO AOS DADOS

Para a camada de negócio e para a camada de acesso aos dados, a figura 1 ilustra a pesquisa do Stackoverflow (2019) mostra as opções mais populares no mercado atualmente.

Embora seja uma pesquisa mais abrangente, envolvendo arcabouços que não são específicos de desenvolvimento web, além de bibliotecas e outras ferramentas, ambas as opções mais populares apresentadas na pesquisa, Node.js e .NET, são utilizadas para desenvolver tanto a camada de negócio quanto a camada de acesso aos dados.

Figura 1 – *Outras Bibliotecas, Ferramentas e Arcabouços mais populares no mercado*



Fonte: STACKOVERFLOW (2019)

Segundo o site institucional da Microsoft (2019), .NET é uma plataforma gratuita, de código aberto e multiplataforma para desenvolver diversos tipos de aplicação. Para desenvolver em .NET o desenvolvedor pode escolher as linguagens C#, F# ou Visual Basic.

No site da Microsoft (2019) é expresso que, para o desenvolvimento de aplicações Web, o ideal seria a utilização do ASP.NET, arcabouço do .NET que adiciona ferramentas e bibliotecas específicas para o desenvolvimento de aplicações web.

Já o Node.js, linguagem de camada de negócio lançada próxima de 2009, utiliza o JavaScript como linguagem de programação. O Node.js utiliza do fato de que o JavaScript já é utilizado no desenvolvimento da camada de apresentação para se popularizar, já que desenvolvedores só precisariam de uma linguagem para desenvolver tanto a camada de apresentação quanto a camada de negócio (MAO, 2018).

Ao comparar as tecnologias citadas acima, Mao (2018) mostra que o ASP.NET se sobressaiu nos testes de performance ao rodar num ambiente Windows, mas que o Node.js foi superior na performance em sistemas Linux. Por fim, Mao (2018) conclui que, por ser desenvolvido pela Microsoft, o ASP.NET é mais adequado para websites empresariais

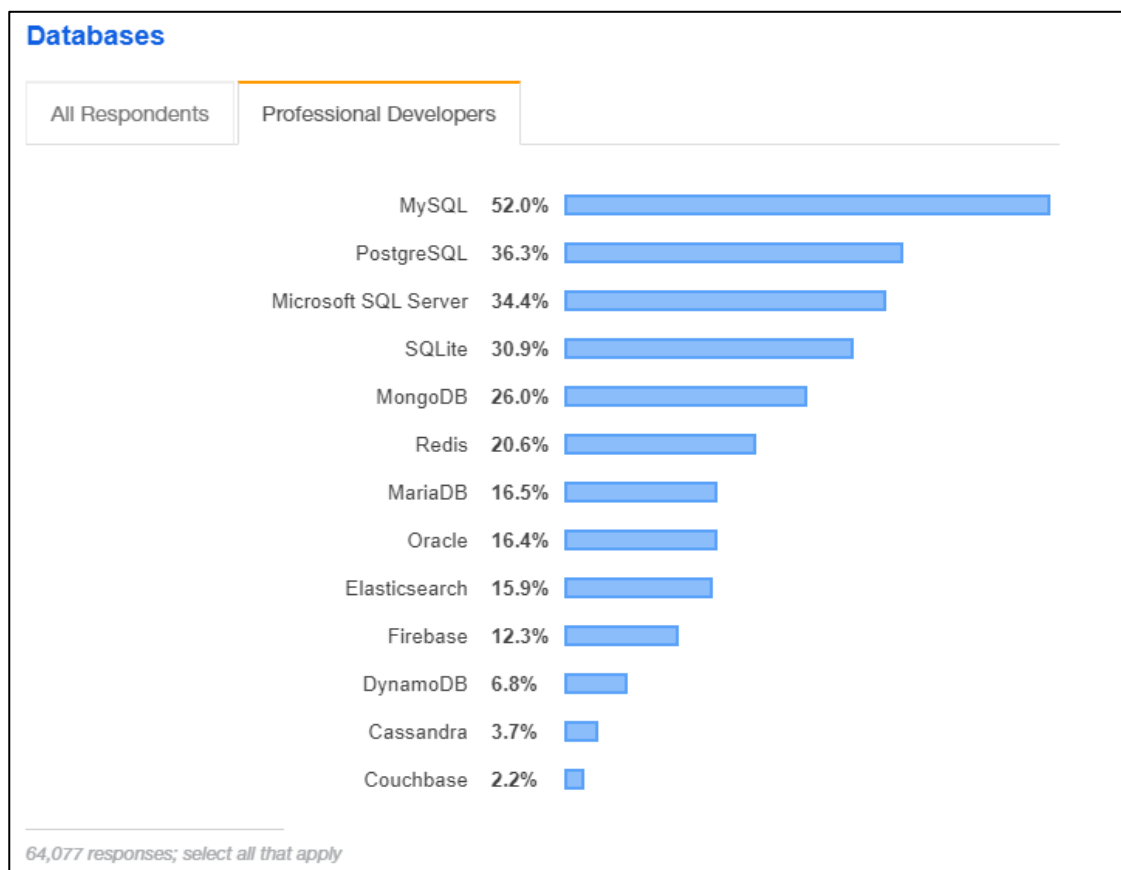
grandes, enquanto que o Node.js se sobressai em aplicações que requerem alta performance e velocidade, ou que sejam de computação intensiva.

Para o desenvolvimento de um MVP, foi escolhida a tecnologia Node.js. A decisão da tecnologia a ser utilizado foi tomada com base em três aspectos: a equipe não está lidando com um site empresarial grande, a equipe gostaria de que sua aplicação pudesse ser servida com alta performance e velocidade, e a equipe estará utilizando JavaScript na programação da Camada de Apresentação. Como Node.js é escrito em JavaScript, é uma linguagem de programação a menos para ser aplicada no projeto, facilitando a inclusão de novos desenvolvedores caso o projeto continue crescendo.

2.3.3 BANCO DE DADOS

Por fim, temos o banco de dados propriamente dito. A figura 2 mostra que na pesquisa do Stackoverflow (2019), os 2 bancos de dados estruturados mais populares no mercado são o MySQL e o PostgreSQL.

Figura 2 – Bancos de dados mais populares no mercado



Fonte: STACKOVERFLOW (2019)

Em seu trabalho, Xiaojie (2011) mostra que, embora o MySQL tenha uma reputação superior e uma participação maior de mercado, o PostgreSQL tem as melhores funcionalidades dentre os dois bancos de dados. Em outras vantagens, o PostgreSQL foi pioneiro em muitos conceitos que só se tornaram disponíveis para outros bancos de dados comerciais muito depois. Em contrapartida, o MySQL é mais simples de ser utilizado, e que sua reputação faz com que desenvolvedores acreditem no potencial do MySQL em superar o PostgreSQL ao longo dos anos.

Além do comparativo, ao acessar os respectivos sites dos 3 bancos de dados citados acima, o autor do presente trabalho observou que apenas o PostgreSQL é distribuído de forma gratuita para uso comercial. Tanto o MSSQL quanto o MySQL incorrem custo de licenciamento para utilização de seus bancos de dados.

Por conta do maior número de funcionalidades, e da licença comercial gratuita, o PostgreSQL foi o banco de dados escolhido para ser utilizado no desenvolvimento do MVP.

2.4 ARCABOUÇOS

Um arcabouço “é um conjunto de classes cooperantes que constroem um projeto reutilizável para uma determinada categoria de software” (GAMMA et al. 2006, p. 41). Uma categoria de software, por exemplo, pode ser o de editores gráficos, abrangendo tanto desenho artístico quanto desenho de máquinas através de um CAD. Outra categoria pode ser aplicações de modelagem financeira (GAMMA et. al. 2006).

Abramhansson & Graviotin (2016) mostram que a popularidade da linguagem JavaScript levou a uma variedade de Arcabouços (Frameworks) sendo desenvolvidos, que visam auxiliar os desenvolvedores em suas tarefas de programação. Pelas facilidades oferecidas por um Arcabouço, é interessante avaliar as opções para a utilização de um Arcabouço durante o MVP. A tabela 1 ilustra os fatores determinantes para a escolha de um arcabouço, de acordo com Abramhansson & Graviotin (2016).

Com o aumento da popularidade do JavaScript, inúmeros arcabouços foram desenvolvidos para auxiliar o desenvolvimento, chegando na casa dos milhares. A figura 3 ilustra os arcabouços que, segundo a pesquisa do site Stackoverflow (2019), são os arcabouços de desenvolvimento de sistemas para internet que tem a maior porcentagem de desenvolvedores que ainda não os utilizam, mas querem começar a utilizar. A escolha de utilizar os arcabouços que mais desenvolvedores querem aprender, e não os mais utilizados,

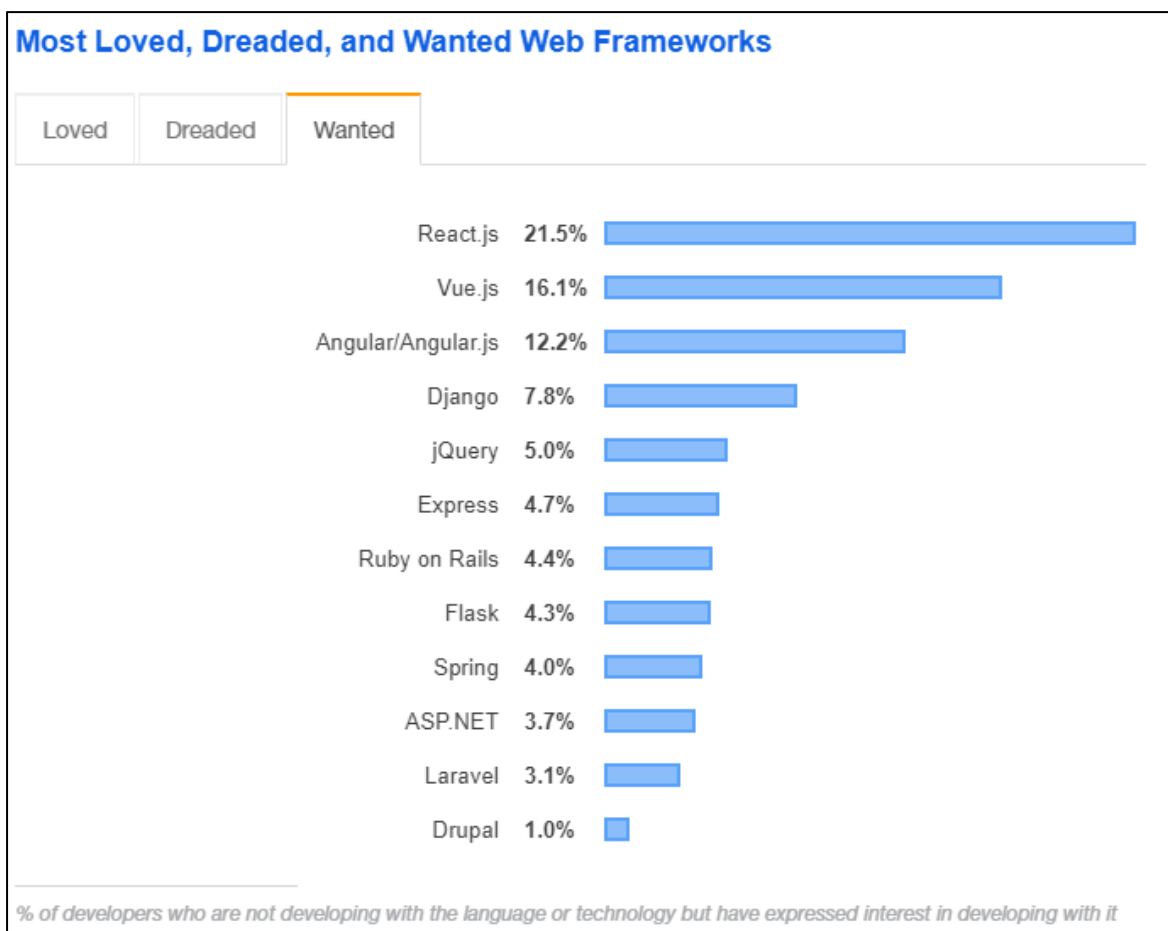
se dá pelo fato de que o estado da arte não se encontra nas tecnologias mais difundidas, mas sim nas mais novas e, conseqüentemente, mais desejadas por aqueles que utilizam arcabouços para desenvolver softwares.

Tabela 1 – *Requisitos de um Arcabouço*

Fator	Descrição
Desempenho	Os ganhos de desempenho que a utilização de um arcabouço trará para o software, que pode ser medido através da carga de processamento, tanto na máquina do usuário, quanto no servidor da aplicação.
Tamanho da aplicação	O número de linhas de código que um arcabouço deve gerar para executar deve ser o menor o possível.
Automatização de tarefas repetitivas	Um arcabouço deve ser capaz de automatizar tarefas que não são relevantes para o desenvolvimento do produto, mas que consomem tempo do desenvolvedor. Tarefas específicas de comunicação entre o software e o navegador, que não agregam para o produto final, devem ser preferencialmente automatizadas pelo arcabouço.
Esforço de aprendizado	Um arcabouço deve ser o mais simples o possível de aprender, pois o tempo de aprendizado não é tempo em que há produção de software.
Flexibilidade	Um arcabouço deve ser capaz de ser flexível para aceitar outros módulos JavaScript, que auxiliam a realizar tarefas mais específicas.
Complexidade	Um arcabouço deve ser capaz de reduzir a complexidade de realizar tarefas de JavaScript. Funcionalidades que antes eram complexas de serem desenvolvidas tornam-se mais simples de serem adicionadas com um arcabouço capaz
Documentação	As funcionalidades de um arcabouço devem ser bem documentadas, com exemplos claros de como realizar tarefas mais comuns. Um desenvolvedor tem de ser capaz de entender rapidamente como implementar as funcionalidades do arcabouço.
Tamanho da Comunidade	O número de contribuintes ativos em um arcabouço é percebido por desenvolvedores como uma garantia de que aquela tecnologia será utilizada no longo prazo, e que o arcabouço receberá melhorias constantes.
Responsividade da Comunidade	Além do tamanho da comunidade, a Responsividade da mesma também influencia na decisão. Uma comunidade responsiva ajudará o desenvolvedor a sanar suas dúvidas com mais agilidade, resolvendo seus problemas em menos tempo.
Aptidão	Um arcabouço deve ser capaz de cumprir com os requisitos do produto.
Atualizações	Um arcabouço deve receber atualizações constantes, estendendo suas funcionalidades para se manter competitivo no mercado. Atualizações garantem compatibilidade com navegadores mais modernos também.
Modularidade	Um arcabouço deve ser capaz de ser modular. Remover ou alterar uma funcionalidade do arcabouço não deve afetar as outras partes do mesmo.
Isolamento	Um arcabouço deve ser capaz de importar bibliotecas externas sem ter de adaptar as mesmas.
Custo	Arcabouços gratuitos são preferidos aos pagos.

Fonte: o Autor

Figura 3 – Arcabouços mais desejados por desenvolvedores



Fonte: STACKOVERFLOW (2019)

Ao comparar os aspectos técnicos e funcionalidades do React.js e Vue.js, os dois arcabouços mais bem ranqueados de acordo com a pesquisa do Stackoverflow (2019), ilustrada na figura 3, vemos que ambos os arcabouços tem conceitos muito parecidos dentro de sua filosofia. Tanto o React.js quanto o Vue.js são arcabouços baseado no conceito de componentização da aplicação. Em termos de linguagem de programação, enquanto o Vue.js dá liberdade para o programador escolher sua linguagem, podendo escolher entre JavaScript ou TypeScript, o React.js exige que o desenvolvedor escreva as aplicações com a notação JSX, que mistura o JavaScript com as linguagens de estruturação de páginas web (HTML – *Hyper Text Markup Language*) e de estilização de páginas web (CSS – *Cascading Style Sheets*). No Vue.js, as linguagens HTML, CSS e JavaScript ficam separadas, embora contidas no mesmo arquivo, uma filosofia que força o desenvolvedor a escrever um código mais conciso e simples, garantindo a capacidade de se reutilizar componentes (WOHLGETHAN, 2018).

Entre suas diferenças, podemos citar que o Vue.js tem uma curva de aprendizagem mais simples, por utilizar JavaScript para programar. O fato de o desenvolvedor ter de aprender a programar em JSX é uma barreira de entrada no React.js durante este comparativo. Já em termos de engajamento da comunidade, podemos notar que, até dezembro de 2018, o React.js tinha uma comunidade maior do que o Vue.js (WOHLGETHAN, 2018).

Dois indicadores utilizados para medir o tamanho de uma comunidade são o número de estrelas que um projeto recebeu no site GitHub, e o número de *downloads* semanais do arcabouço. Em dezembro de 2018, o arcabouço React.js tinha cerca de 3,6% mais estrelas no GitHub (93.299 contra cerca de 90.000 estrelas do Vue.js), mas tinha 4 vezes mais *downloads* (mais de 200.000 *downloads* semanais, contra menos de 50.000 do Vue.js) (WOHLGETHAN, 2018).

Por fim, Wohlgethan (2018) conclui que tanto o React.js quanto o Vue.js não diferem tanto, e que nenhum dos arcabouços é superior ao outro, e que, portanto, aspectos como conhecimento dos desenvolvedores e experiências prévias contam muito para a escolha de qual arcabouço adotar. A equipe de desenvolvimento então decidiu por utilizar o React.js por conta de experiências prévias da equipe com a tecnologia.

3. DESENVOLVIMENTO

O presente capítulo apresenta o histórico da empresa alvo deste trabalho, e do surgimento da ideia que se tornou o MVP desenvolvido pelo presente trabalho. Após, o capítulo versa sobre as escolhas da equipe de projeto a respeito do tipo de MVP desenvolvido, e das escolhas de tecnologia para desenvolver o referido MVP, a partir das tecnologias apresentadas na revisão bibliográfica, e os motivos que levaram a equipe a tomar tais decisões.

Além disso, o capítulo apresenta o desenvolvimento do MVP, seguindo o planejamento e escopo desenvolvidos por Oliveira (2019). Serão demonstradas as decisões de alterações no planejamento de Oliveira (2019), quando necessárias, para se adaptar à realidade do desenvolvimento de um software. E por fim, será apresentado o MVP desenvolvido ao longo do presente trabalho.

3.1 A EMPRESA E O PRODUTO

A empresa alvo do presente trabalho, denominada “Empresa A”, foi criada a fim de desenvolver o software tema deste trabalho. Composta de 2 integrantes, a Empresa A está desenvolvendo um produto que será a ponte entre uma pessoa, física ou jurídica, que deseja organizar um evento, mas não tem know-how para tal, e uma pessoa, física ou jurídica, que deseja fornecer serviços a eventos da região. Em um único *software* o cliente que deseja organizar seu evento deve ser capaz de realizar a maior parte das tarefas que envolver organizar seu evento por completo, enquanto que o cliente que deseja fornecer seus serviços deve ser capaz de organizar seus agendamentos e contatos com possíveis clientes através da plataforma.

Oliveira (2019) mostra que a ideia do produto veio de uma observação da concentração de muitos eventos na mão de poucas empresas dentro do mercado de eventos. Segundo levantamento da ABEOC (2013), mais de quinhentos e noventa mil eventos foram realizados em solo brasileiro no ano de 2013, ao passo que menos de oitocentas empresas são cadastradas na associação.

3.2 O ESCOPO

Neste capítulo, avalia-se o escopo proposto por Oliveira (2019) para o desenvolvimento de um MVP de um software de eventos, com enfoque nas tecnologias a serem utilizadas para o desenvolvimento do trabalho.

3.2.1 REVISÃO DO ESCOPO

Em seu trabalho, Oliveira (2019) define uma lista inicial de atividades, chamada de *Product Backlog*, que são subdividas em 5 *Sprints*, representados através de *Sprint Backlogs*, para o desenvolvimento e manutenção do MVP. Para o presente trabalho, foram desenvolvidos os 3 primeiros *Sprints*, que se referem ao desenvolvimento do MVP. Os *Sprints* 4 e 5 referem-se à manutenção do produto e autorreflexão sobre o que foi desenvolvido nos *Sprints* anteriores, e portanto, não fazem parte da parte principal do desenvolvimento do MVP, objetivo do presente trabalho.

3.3 DESENVOLVIMENTO DO MVP

O processo de desenvolvimento do MVP se dividiu em 3 *Sprints*. Cada *Sprint* teria duração de 4 semanas, totalizando 12 semanas de desenvolvimento.

Para o desenvolvimento dos *Sprints*, seguiu-se a seguinte metodologia. Primeiro, a equipe de desenvolvimento do projeto se reuniu para discutir as atividades definidas em cada *Sprint Backlog* e propor as alterações que julgou necessárias. Como Oliveira (2019) afirma a respeito do *Product Backlog*, “Essa lista pode e deverá ser alterada durante o projeto, principalmente a cada *Sprint*.” Após a revisão do *Sprint*, a equipe de desenvolvimento então seguiu as atividades propostas e acompanhou o tempo de desenvolvimento, completando o *Burndown Chart* ao longo do desenvolvimento.

É importante frisar que, durante o desenvolvimento de uma funcionalidade, nem sempre a estilização da mesma foi feita no mesmo Item ou *Sprint* do desenvolvimento da funcionalidade. Para facilitar o entendimento do resultado do desenvolvimento dos itens, todas as figuras abaixo que se referem ao resultado do desenvolvimento de um item representam o produto final, após a conclusão da “Estilização do website”, do *Sprint* 3 (item 11).

3.3.1 SPRINT 1

Oliveira (2019) define que o primeiro *Sprint* é um dos mais importantes, pois aqui tomam-se decisões fundamentais para o sucesso do projeto, como as tecnologias que serão utilizadas e a configuração do ambiente de trabalho. É no primeiro *Sprint* também que são criadas as funcionalidades de cadastro de usuários e de fornecedores, para que possa ser feito o pré-registro antes da plataforma estar disponível para utilização.

Tabela 2 – *Sprint Backlog 1*

SPRINT 1			
Item	Descrição	Estimativa de tempo (semanas)	Sprint #
1	Landing page	1	1
2	Seleção das Tecnologias		
3	Configurar o Ambiente de desenvolvimento	0.5	1
4	Cadastro de usuário	1	1
5	Cadastro de fornecedor	1	1

Fonte: OLIVEIRA (2019)

Revisando o *Sprint Backlog* contido na tabela 2, a equipe de desenvolvimento do presente projeto recomenda a alteração da estimativa de tempo do item ‘Configurar o Ambiente de desenvolvimento’. Embora à primeira vista pareça uma tarefa simples, como a equipe de desenvolvimento do projeto não é experiente em todas as tecnologias abordadas anteriormente, certas especificidades podem tornar essa tarefa mais complexa do que prevista inicialmente.

Portanto, o a equipe recomenda que uma estimativa mais cautelosa, de 1 semana para os itens 2 e 3, seja escolhida. Além disso, o desenvolvimento da *Landing Page* depende também das tecnologias escolhidas, e pode ser beneficiado pela configuração do Ambiente de Desenvolvimento. Logo, a equipe de desenvolvimento do projeto sugere a mudança na ordem dos itens, passando a *Landing Page* para o item 3. Com isso, a tabela 3 representa o *Sprint Backlog* 1 revisado.

Tabela 3 – *Sprint Backlog 1 Revisado*

SPRINT 1			
Item	Descrição	Estimativa de tempo (semanas)	Sprint #
1	Seleção das tecnologias		
2	Configurar o Ambiente de desenvolvimento	1	1
3	Landing page	1	1
4	Cadastro de usuário	1	1
5	Cadastro de fornecedor	1	1

Fonte: o Autor

3.3.1.1 RESULTADOS DO SPRINT 1

No sprint 1, conforme mostra a tabela 3, desenvolveram-se os itens numerados de 1 a 5. Embora certos itens tenham consumido mais tempo do que o previsto, outros provaram ser mais rápidos do que as estimativas. A tabela 4 mostra o tempo real gasto com o desenvolvimento de cada uma das funcionalidades.

Tabela 4 – Acompanhamento do Sprint 1

SPRINT 1				
Item	Descrição	Estimativa de tempo (semanas)	Tempo real gasto (semanas)	Atrasado?
1	Seleção das tecnologias			
2	Configurar o Ambiente de desenvolvimento	1	2	Não
3	Landing page	1	1	Não
4	Cadastro de usuário	1	0.5	Não
5	Cadastro de fornecedor	1	0.5	Não

3.3.1.2 ITEM 1 – SELEÇÃO DAS TECNOLOGIAS

Para a tomada de decisão na seleção das tecnologias, utilizou-se da Revisão Bibliográfica desenvolvida no capítulo 2.2 do presente trabalho. Conforme apresentado no capítulo referido, faz-se necessário escolher uma tecnologia para a camada de apresentação, uma tecnologia para a camada de negócio, e um banco de dados.

Para a camada de apresentação, embora o projeto tenha de ser desenvolvido em JavaScript, pois é a linguagem de programação interpretada pelos navegadores (Flanagan, 2004), foram apresentados dois arcabouços que poderiam auxiliar no desenvolvimento. Wohlgethan (2018) mostra que tanto o React.js quanto o Vue.js não diferem tanto em suas especificidades, e que aspectos como o conhecimento dos desenvolvedores e experiências prévias eram os fatores que levavam a adoção de um arcabouço específico. A equipe de desenvolvimento do MVP tem experiências prévias com React.js, e não possui nenhuma experiência com Vue.js. Por este quesito, foi decidido adotar o arcabouço React.js para o projeto.

Na camada de negócios, Mao (2018) avalia as tecnologias ASP.NET e Node.js, e explica que o Node.js se sobressai em aplicações que requerem alta performance e velocidade. Smith (2017) mostra que aplicativos modernos devem ser capazes de atender altos picos de demanda de usuários cada vez mais exigentes. Por este motivo, foi escolhido

o Node.js para o desenvolvimento da camada de negócios, visto que velocidade nos aplicativos modernos é um fator crucial.

Por fim, na escolha do banco de dados, Xiaojie (2011) compara os bancos de dados mais populares, o MySQL e o PostgreSQL, e mostra que o PostgreSQL, além de não possuir custo de licenciamento para uso comercial, é também o banco que possui as melhores funcionalidades dentre os dois bancos. Portanto, decidiu-se utilizar o banco de dados PostgreSQL para este projeto.

3.3.1.3 ITEM 2 – CONFIGURAR O AMBIENTE DE DESENVOLVIMENTO

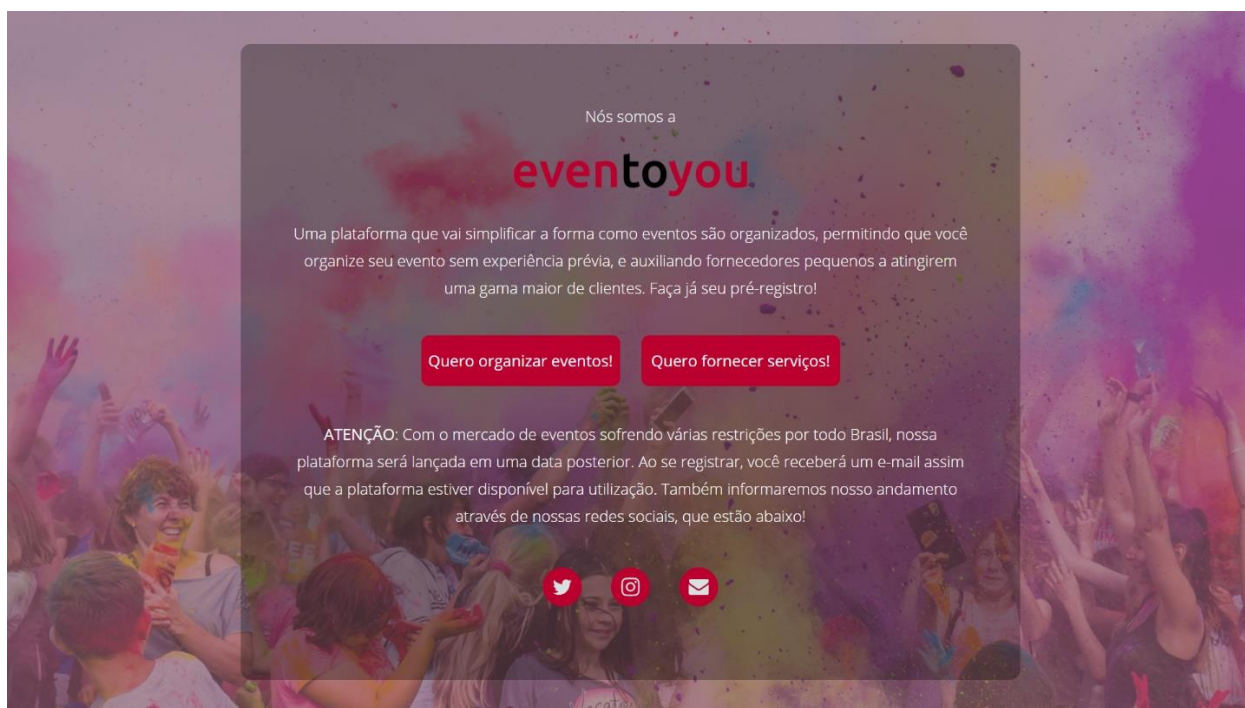
Conforme avaliado no capítulo 3.2.2.1, certas especificidades na configuração das tecnologias selecionadas acima poderiam tornar esta atividade mais trabalhosa do que o previsto. Embora a equipe de desenvolvimento tenha experiências prévias com o React.js, tecnologia escolhida para a camada de apresentação, foi a primeira experiência da equipe com as tecnologias Node.js e PostgreSQL. Tanto a configuração do banco de dados PostgreSQL e a configuração da camada de acesso aos dados via Node.js provaram-se mais complicadas do que o previsto, o que ocasionou um atraso significativo na configuração do ambiente de desenvolvimento. Embora tenha sido executado com sucesso, o item 2 atrasou em uma semana quanto à estimativa de tempo inicial, levando duas semanas para ser concluído.

3.3.1.4 ITEM 3 – LANDING PAGE

Foi desenvolvida uma *Landing Page*, que, conforme definido por Ries (2011), é uma página utilizada para comunicar rapidamente as propostas do projeto, difundir objeções e realizar chamadas de ação. Para o projeto atual, as chamadas de ação na *Landing Page* são o convite a realizar o pré-cadastro de Clientes e Fornecedores, para que a startup tenha uma base de fornecedores a ser disponibilizada a partir do primeiro dia para clientes que buscam organizar seus eventos. Clientes e Fornecedores que se cadastrarem na *Landing Page* também serão avisados quando o MVP estiver disponível para utilização.

O desenvolvimento da *Landing Page* correu conforme o estimado, não resultando em atrasos.

Figura 4– Landing Page



Fonte: o Autor

3.3.1.5 ITENS 4 E 5 – CADASTRO DE USUÁRIO E DE FORNECEDOR

O desenvolvimento dos itens 4 e 5 é bastante similar. Para o desenvolvimento do cadastro, tanto de usuário quanto o de fornecedor, é necessário criar tabelas no banco de dados para armazenar os dados, desenvolver uma lógica para validar e inserir os dados nas tabelas, a partir do Node.js, e desenvolver as telas no React.js, para que o usuário tenha os campos para digitar e enviar ao servidor. A figura 6 representa o resultado final da camada de apresentação, tanto do cadastro de usuários quanto do cadastro de fornecedores.

O desenvolvimento dos itens 4 e 5 superou as estimativas, com um tempo real gasto de 0,5 semanas para cada um dos itens.

Figura 5 – Telas de cadastro de usuário e fornecedor

eventoyou

SOU UM...

CLIENTE

Nome completo

Seu e-mail

Sua senha

Criar conta

[Já possuo uma conta](#)

eventoyou

SOU UM...

FORNECEDOR

Nome completo

Seu e-mail

Sua senha

INFORMAÇÕES DE FORNECEDOR

Seu nome fantasia

Telefone de contato

E-mail de fornecedor

Seu endereço

Sua descrição

Criar conta

[Já possuo uma conta](#)

Fonte: o Autor

3.3.2 SPRINT 2

Oliveira (2019) descreve o segundo sprint como o de duas funcionalidades importantes para o projeto: a funcionalidade de login, e o mapa interativo com fornecedores.

Tabela 5 – *Sprint Backlog 2*

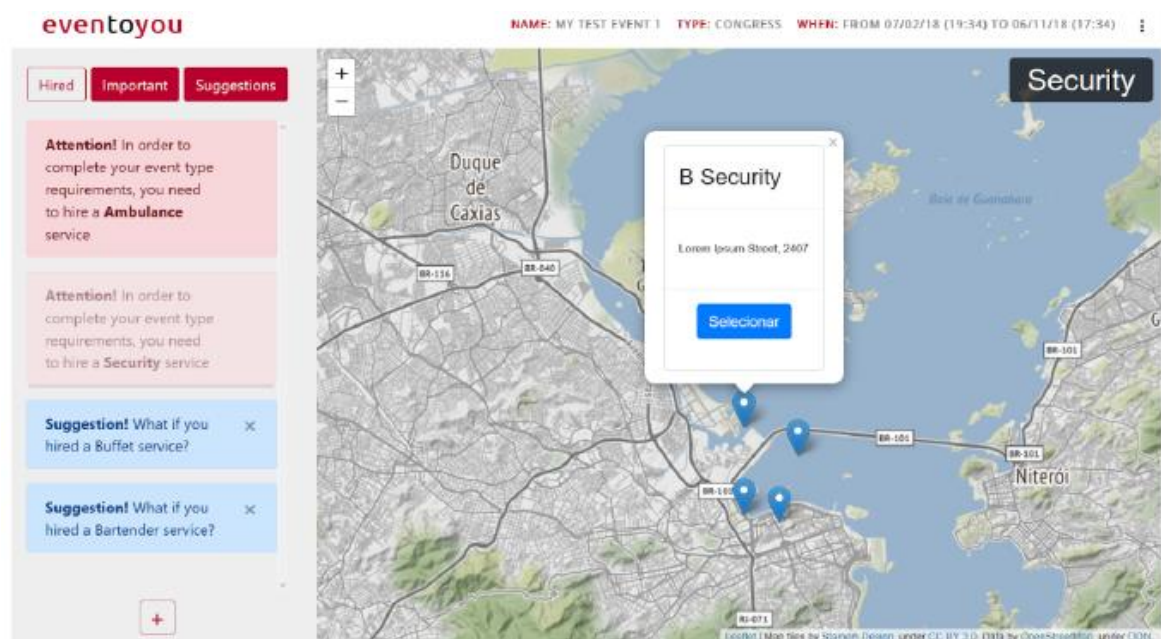
SPRINT 2			
Item	Descrição	Estimativa de tempo (semanas)	Sprint #
6	Funcionalidades de Login	1	2
7	Mapa interativo com fornecedores	3	2

Fonte: OLIVEIRA (2019)

Aqui, há a necessidade de se fazer algumas alterações. Embora Oliveira (2019) cite apenas o mapa interativo, sua descrição do que é o mapa interativo faz com que a atividade tenha de ser subdividida em mais tarefas. A intenção do mapa não é conter apenas o mapa, mas sim ter a capacidade também de selecionar os fornecedores e entrar em contato com os mesmos. E para acessar o mapa interativo, como visto na figura 7, de um protótipo anterior ao início do trabalho de Oliveira (2019), verifica-se que o usuário já criou também um evento, e especificou o tipo do evento e o horário em que irá acontecer.

Portanto, a equipe de desenvolvimento do projeto julga necessária a divisão da atividade ‘Mapa interativo com fornecedores’ em ‘Cadastro de eventos’, ‘Agendamento de fornecedores’ e ‘Mapa interativo com fornecedores’. A tabela 6 representa o *Sprint Backlog 2* revisado.

Figura 6 – Protótipo de Mapa interativo



Fonte: OLIVEIRA (2019)

Tabela 6 – Sprint Backlog 2 Revisado

SPRINT 2			
Item	Descrição	Estimativa de tempo (semanas)	Sprint #
6	Funcionalidades de Login	1	2
7	Cadastro de eventos	0.5	2
8	Agendamento de fornecedores	1	2
9	Mapa interativo com fornecedores	1.5	2

Fonte: o Autor

3.3.2.1 RESULTADOS DO SPRINT 2

No sprint 2, conforme mostra a tabela 7, desenvolveram-se os itens numerados de 6 a 9. Embora certos itens tenham sido realizados antes do previsto, o item 9 ficou atrasado e utilizou tempo de desenvolvimento alocado ao Sprint 3. A tabela 7 mostra o tempo real gasto com o desenvolvimento de cada uma das funcionalidades.

Tabela 7 – Acompanhamento do Sprint 2

SPRINT 2				
Item	Descrição	Estimativa de tempo (semanas)	Tempo real gasto (semanas)	Atrasado?
6	Funcionalidades de Login	1	0.5	Não
7	Cadastro de eventos	0.5	0.5	Não
8	Agendamento de fornecedores	1	1	Não
9	Mapa interativo com fornecedores	1.5	2.5	Sim

3.3.2.2 ITEM 6 – FUNCIONALIDADES DE LOGIN

No item 6, além da autenticação propriamente dita, utilizando o nome de usuário e senha cadastrados no item 4, “Cadastro de Usuário”, também há a necessidade de se desenvolver uma maneira de persistência do login do usuário, para que o mesmo não tenha de se autenticar múltiplas vezes com seu usuário e senha em um curto espaço de tempo, e que seja feita de maneira segura, de modo a não comprometer os dados do usuário.

Para tal, a equipe de desenvolvimento escolheu utilizar do *JSON Web Token* (JWT) para criar o conceito de persistência. Borba (2017) define o JWT como uma chave eletrônica que é retornada do servidor, presente na camada de negócios, para a máquina do usuário, que então a armazena. Ao fazer novas requisições ao servidor, é enviado automaticamente o JWT junto com a requisição.

O servidor então é capaz de utilizar as informações contidas no JWT para, de maneira segura, autenticar o usuário sem que o mesmo precise inserir novamente seus dados de usuário e senha. Tanto a implementação da funcionalidade de login quanto a adição do JWT provaram ser tarefas simples, com um tempo real gasto de 0,5 semanas para o desenvolvimento das atividades listadas no item.

3.3.2.3 ITEM 7 – CADASTRO DE EVENTOS

O desenvolvimento do item 7 se assemelha bastante aos itens 4 e 5, desenvolvidos previamente, por se tratar também de uma funcionalidade de cadastro. Para o item 7, a equipe de desenvolvimento poderia implementar o cadastro de duas maneiras: o evento estará atrelado a apenas um usuário, ou múltiplos usuários podem coordenar um único evento. A equipe de desenvolvimento julgou que a opção mais simples seria a primeira, em que um evento estaria atrelado a apenas um usuário, e que, caso o valor percebido para a funcionalidade fosse suficientemente grande durante a fase de testes do MVP, a equipe incluiria o item para ser desenvolvido a partir do *Sprint* 6. O desenvolvimento do item 7 correu conforme o planejado, com o tempo gasto de 0,5 semanas sendo igual ao tempo estimado.

Figura 7 – Cadastro de Eventos



O formulário, intitulado "Novo evento", contém os seguintes campos:

- Nome do evento
- De: dd/mm/aaaa (com ícone de calendário) e --:-- (com ícone de relógio)
- Até: dd/mm/aaaa (com ícone de calendário) e --:-- (com ícone de relógio)
- Cidade
- Endereço
- Botão "Criar evento" em fundo vermelho.

Fonte: o Autor

3.3.2.4 ITEM 8 – AGENDAMENTO DE FORNECEDORES

O desenvolvimento do item 8 também se assemelha ao desenvolvimento dos demais itens de cadastro. O agendamento de fornecedores pode ser entendido como um cadastro de um agendamento, o que torna o item similar aos itens 4, 5 e 7. No item 8, além do cadastro de agendamento, são necessárias mais duas funcionalidades. A primeira é a de que o fornecedor deve ser capaz de aceitar, modificar ou recusar um agendamento proposto por um usuário, através de um painel. E a segunda é a de que, quando houverem propostas ou alterações de agendamentos, por ambas as partes, a outra parte deve ser notificada. A equipe do projeto reconhece que o ato do fornecedor aceitar ou recusar um agendamento são alterações no sentido de que alteram o status de um agendamento.

A equipe de desenvolvimento também reconheceu que uma funcionalidade que tornaria o agendamento ainda melhor, futuramente, é a de que o usuário pudesse ser informado, antes de propor um agendamento com um fornecedor, se o mesmo tem disponibilidade para atendê-lo. Como fornecedores podem ser capazes de atender múltiplos usuários, e suas capacidades de atendimento podem ser variáveis, as múltiplas possibilidades tornam de difícil avaliação inicial sobre qual método de avaliar a indisponibilidade do fornecedor seria o mais adequado. Com a utilização do MVP por usuários e fornecedores, e com a agregação dos dados de utilização e do *feedback* recebido durante o MVP a equipe então desenvolverá uma solução, a partir do Sprint 6.

Figura 8 – *Agendamento Proposto*



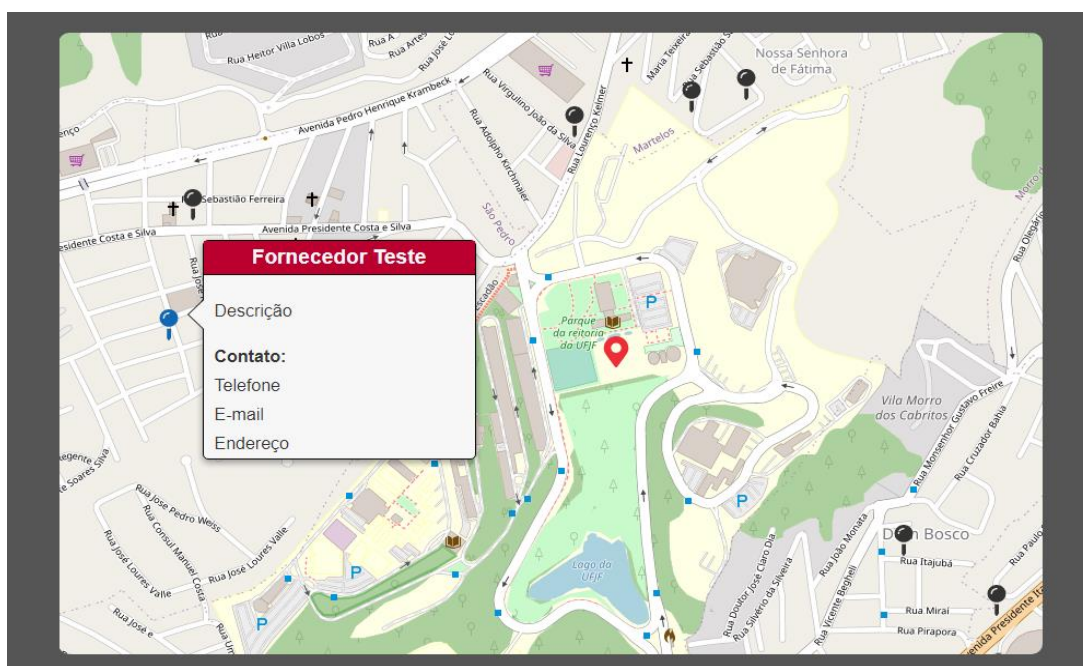
Fonte: o Autor

3.3.2.5 ITEM 9 – MAPA INTERATIVO COM FORNECEDORES

A equipe de desenvolvimento do projeto julgou como sendo uma maneira eficiente de selecionar fornecedores para o seu evento um mapa interativo. O objetivo do mapa interativo é oferecer uma alternativa na hora de selecionar os fornecedores, podendo avaliar os mesmos também pela proximidade com o evento. Especialmente em cidades grandes, a equipe de desenvolvimento julgou que logística pode se tornar um impeditivo na hora de selecionar um fornecedor, e portanto, um mapa que indicasse a distância entre o fornecedor e o local do evento poderia simplificar o processo de escolha.

O mapa interativo já havia sido desenvolvido previamente em um primeiro protótipo do produto, anterior ao trabalho de Oliveira (2019). A figura 7 mostra o protótipo do mapa interativo desenvolvido anteriormente. Para a implementação do item 9 no MVP, a equipe do projeto julgou que poderia aproveitar muitas partes do protótipo desenvolvido anteriormente. Porém, tal julgamento provou-se incorreto, e a equipe acabou levando mais tempo do que o previsto para implementar a funcionalidade. Como funcionalidades possíveis de serem desenvolvidas posteriormente ao *Sprint 5*, a equipe avaliou que poderia também implementar filtros no mapa, para que fosse possível localizar fornecedores, por exemplo, com uma avaliação mínima, ou apenas exibir fornecedores que estivessem a uma certa distância do local do evento.

Figura 9 – Mapa Interativo Com Fornecedores



Fonte: o Autor

3.3.3 SPRINT 3

Neste sprint, Oliveira (2019) define as atividades como sendo o Desenvolvimento do Website, e a funcionalidade de avaliação, como consta na tabela 8.

Tabela 8 – *Sprint Backlog 3*

SPRINT 3			
Item	Descrição	Estimativa de tempo (semanas)	Sprint #
8	Funcionalidades de avaliação	2	3
9	Desenvolvimento de website	2	3

Fonte: OLIVEIRA (2019)

Ao chegar no terceiro *Sprint*, a equipe de desenvolvimento do MVP constatou que já havia desenvolvido a parte do website (camada de apresentação) de todas as funcionalidades propostas anteriormente. Com isso, a equipe de desenvolvimento julgou melhor substituir o item 9 para “Estilização do website”, que agora passa a ser o item 11, visto que foram adicionados mais dois itens no segundo *Sprint*.

Com isso, ajustes de estilização, meramente visuais e que não adicionam novas funcionalidades ao website foram feitas no final dos 3 *Sprints* desenvolvidos. A equipe de desenvolvimento concentrou-se em desenvolver as funcionalidades primeiramente, e ao final, corrigiu detalhes apenas visuais que ainda existiam no MVP. A tabela 9 representa o *Sprint Backlog 3* revisado.

Tabela 9 – *Sprint Backlog 3 Revisado*

SPRINT 3			
Item	Descrição	Estimativa de tempo (semanas)	Sprint #
10	Funcionalidades de avaliação	2	3
11	Estilização do website	2	3

Fonte: o Autor

3.3.3.1 RESULTADOS DO SPRINT 3

No sprint 3, conforme mostra a tabela 9, desenvolveram-se os itens 10 e 11. Diferentemente dos *Sprints* anteriores, todas as tarefas foram cumpridas dentro do tempo planejado ou se adiantaram. A meia semana de atraso do *Sprint* 2 foi compensada pelo item 11, que levou apenas 1,5 semanas, perante a 2 semanas planejadas para o mesmo. A tabela 10 mostra o tempo real gasto com o desenvolvimento de cada uma das funcionalidades.

Tabela 10 – Acompanhamento do Sprint 3

SPRINT 3				
Item	Descrição	Estimativa de tempo (semanas)	Tempo real gasto (semanas)	Atrasado?
10	Funcionalidades de avaliação	2	2	Não
11	Estilização do website	2	1.5	Não

3.3.3.2 ITEM 10 – FUNCIONALIDADES DE AVALIAÇÃO

Para desenvolver a funcionalidade de avaliação, a equipe de desenvolvimento do projeto tomou inspiração em aplicativos que possuem funcionalidades similares executadas com sucesso, como o aplicativo Uber, por exemplo. A equipe decidiu que, para o MVP, o objetivo da funcionalidade era que a avaliação fosse feita de maneira rápida, para que mais clientes avaliassem o desempenho dos fornecedores. Portanto, foi adotado um modelo de avaliação entre 1 a 5 estrelas, que aparece como uma notificação na plataforma após o prazo de término de um evento criado na mesma, e também é enviado um pedido por e-mail para que o cliente avalie os fornecedores que prestaram serviços a seu evento.

Também foi disponibilizado um espaço para que os clientes deixassem comentários a respeito dos fornecedores, caso julgassem que apenas as estrelas não fossem um indicativo suficiente da real avaliação do fornecedor. Como objetivos futuros, a serem desenvolvidos após o *Sprint* 5, a equipe constatou que poderá oferecer também uma área para que fornecedores façam comentários a respeito de seus clientes, ou que possam responder as avaliações comentadas dos mesmos.

Como durante a fase de MVP a utilização prevista do aplicativo é baixa, ainda não há uma área para observar a avaliação atual de um fornecedor, já que a plataforma precisa de algumas avaliações antes de formar uma nota utilizando a média das avaliações. Caso as avaliações fossem exibidas desde a primeira nota, um fornecedor que teve um problema pontual com um primeiro cliente poderia perder futuras oportunidades de ter uma nota média maior, pois desde seu primeiro evento, já teria nota 1 na plataforma, o que poderia distanciar potenciais novos clientes.

3.3.3.3 ITEM 11 – ESTILIZAÇÃO DO WEBSITE

No último item do *Sprint 3*, a equipe de desenvolvimento voltou a todos os itens desenvolvidos anteriormente e ajustou problemas pontuais que eram meramente estéticos. Foram reposicionados objetos na tela, os menus sofreram alterações, a paleta de cores do *website* foi ajustada, utilizando como inspiração outros *websites* populares, cujo design já passou pela aprovação do público ao longo de anos. Sites como os produtos da *Google* e do *Facebook* foram utilizados como referências nos ajustes na estilização do produto desenvolvido no presente trabalho.

4. RESULTADOS E DISCUSSÕES

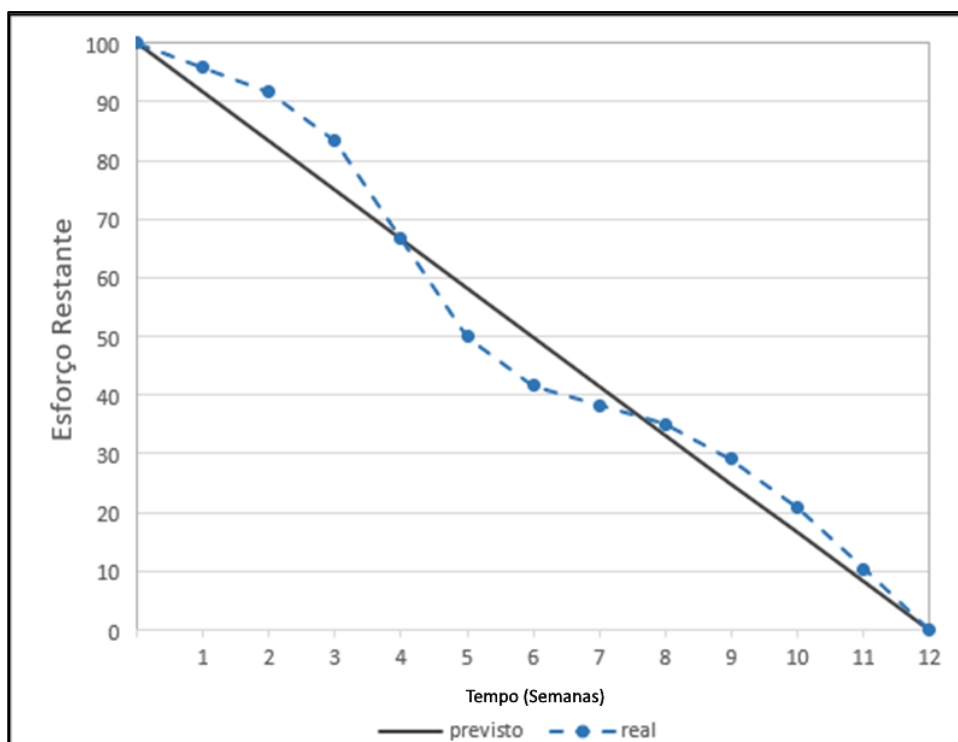
Este capítulo busca apresentar os resultados alcançados durante o capítulo 3, mostrando uma visão geral de como foi o desenvolvimento, assim como discussões do que deve ser feito para continuar melhorando o projeto, nos sprints subsequentes. O capítulo também apresentará o *Burndown Chart* final do projeto, descrito por Oliveira (2019) como ferramenta importante da metodologia *Scrum* para acompanhar o desenvolvimento do projeto.

4.1 RESULTADOS ALCANÇADOS

Ao final das 12 semanas de desenvolvimento, nas quais foram realizados 3 *Sprints* de 4 semanas, todos os itens listados a serem desenvolvidos para o MVP foram concluídos. A figura 11 representa o *Burndown Chart* do projeto, mostrando a evolução do desenvolvimento ao longo do tempo de projeto. O eixo horizontal representa o tempo decorrido do projeto, medido em semanas, e o eixo vertical representa o esforço a ser realizado para a conclusão do projeto.

Pode-se observar que durante o início do projeto e no fim, a equipe esteve atrasada em relação ao tempo previsto. Os itens 1, 2 e 9 atrasaram quanto à sua previsão, fazendo com que o *Burndown Chart* se mantivesse acima da linha planejada após esses itens. Também pode-se observar que, durante o meio do projeto, a equipe conseguiu adiantada em alguns momentos, após os itens 4, 5 e 6, que foram concluídos antes do previsto, o que permitiu a conclusão do projeto no tempo estimado mesmo com alguns itens atrasados.

A equipe de desenvolvimento do projeto também observou que a metodologia *Scrum* auxiliou no desenvolvimento do MVP, mesmo sendo uma equipe de tamanho inferior ao que é definido como mínimo para a utilização do *Scrum*. A organização em *Sprints* e o *Burndown Chart* provaram-se ferramentas de extrema importância para manter o foco da equipe e garantir que o projeto estaria concluído dentro do prazo determinado, e a equipe acredita que seguirá utilizando o *Scrum* para seus desenvolvimentos futuros.

Figura 10 – *Burndown Chart* do MVP

Fonte: o Autor

4.2 PRÓXIMOS PASSOS

Embora a equipe tivesse estabelecido, como descrito por Oliveira (2019), que o objetivo inicial para os *Sprint* 4 e 5 eram a Manutenção do MVP e Autorreflexão sobre o que foi desenvolvido, com usuários já testando o *software*, a situação mundial nos anos de 2020 e 2021 levou a uma alteração no planejamento.

Com grandes restrições sendo aplicadas a Eventos por todo o Brasil durante os anos de 2020 e 2021, a equipe responsável pelo MVP julgou que o lançamento do projeto deve ser adiado para uma data posterior, e que mais funcionalidades, previstas para serem adicionadas após o período de testes, podem ser desenvolvidas enquanto a equipe aguarda um momento mais oportuno para o lançamento do produto.

Ao longo do capítulo 3, foram levantadas diversas ideias pela equipe de projeto acerca de funcionalidades que poderiam ser desenvolvidas. São elas:

1. Possibilitar múltiplos usuários coordenando um único evento, permitindo que várias contas possam realizar alterações e contratações de fornecedores.

2. Verificação automática de disponibilidade do fornecedor para um evento numa data específica, sem que o usuário tenha de fazer contato direto com o mesmo.
3. Filtros disponíveis no mapa interativo, como localização de fornecedores que estivessem a uma distância mínima do local do evento.
4. Permitir que fornecedores respondam comentários deixados por usuários
5. Permitir que fornecedores comentem sobre sua experiência com usuários, para que outros fornecedores possam ler seus comentários antes de aceitar propostas.

5. CONCLUSÃO

Todos os objetivos apresentados durante a metodologia foram concluídos com sucesso. O presente trabalho foi capaz de ilustrar as tecnologias disponíveis para o desenvolvimento de um aplicativo disponível em um navegador de *internet*. A equipe de projeto foi capaz de desenvolver um Produto Mínimo Viável para um *Software* na área de eventos, seguindo um escopo previamente estabelecido.

Também foi possível observar, ao longo das 12 semanas de desenvolvimento do produto, quais alterações seriam necessárias no escopo inicial, visto que este é um produto que ainda não atingiu uma maturidade em seu ciclo de desenvolvimento. A equipe continua tendo ideias para melhorar o produto a cada dia que passa, e tais ideias podem agregar valor ao que foi estabelecido no início do projeto.

Por fim, estudos futuros podem ser realizados no acompanhamento do projeto, observando a adaptação da equipe com a metodologia *Scrum* conforme o projeto e a equipe vão crescendo.

REFERÊNCIAS

ABEOC. **II Dimensionamento Econômico da indústria de Eventos no Brasil** - 2013.

ABRAHAMSSON, Pekka; DUC, Anh Nguyen. **Minimum viable product or multiple facet product? The Role of MVP in software startups**. In: International Conference on Agile Software Development. Springer, Cham, 2016. p. 118-130.

BORBA, Joceano Alves de. **Sistema web para emissão de cupons de desconto**. Monografia (Especialização em Tecnologia Java) - Departamento Acadêmico de Informática, Universidade Tecnológica Federal do Paraná. 2017.

BERGHEL, Hal. **A Web monopoly?**. PC AI, v. 10, n. 1, p. 22-23, 1996. Disponível em: <<http://www.berghel.net/col-edit/cybernautica/jan-feb96/pcai961.php>>. Acesso em: 25 de out. de 2019

ENGE, Eric. **Where is the Mobile vs. Desktop Story Going?** Perficient Digital, 2019. Disponível em: <<https://www.perficientdigital.com/insights/our-research/mobile-vs-desktop-usage-study>> . Acesso em: 12 de set. de 2019

FLANAGAN, David. **JavaScript: o guia definitivo**. Bookman Editora, 2004.

GAMMA, Erich. **Padrões de projetos: soluções reutilizáveis**. Bookman editora, 2009.

GEORGIA TECH COLLEGE OF COMPUTING. **GVU's 5th Web Survey**. Disponível em: < https://www.cc.gatech.edu/gvu/user_surveys/survey-04-1996/ >. Acesso em: 22 de out. de 2019.

GRAZIOTIN, Daniel; PANO, Amantia; ABRAHAMSSON, Pekka. **What leads developers towards the choice of a JavaScript framework?**. arXiv preprint arXiv:1605.04303 2016.

LENARDUZZI, Valentina; TAIBI, Davide. **MVP explained: A systematic mapping study on the definitions of minimal viable product**. In: 2016 42th Euromicro Conference on Software Engineering and Advanced Applications (SEAA). IEEE, 2016. p. 112-119.

KEITH, Jeremy. **DOM scripting: web design with JavaScript and the Document Object Model**. Apress, 2006.

MAO, Xiaoli. **Comparison between Symfony, ASP. NET MVC, And Node. js Express for Web Development**. 2018.

MICROSOFT. **What is .NET?** 2019. Disponível em: <<https://dotnet.microsoft.com/learn/dotnet/what-is-dotnet>>. Acesso em: 20 de set. de 2019

OLIVEIRA, Guilherme Freitas de. **Aplicação de metodologia Scrum: Estudo de caso no desenvolvimento de software na área de eventos**. TCC (Graduação em Engenharia de Produção) – Universidade Federal de Juiz de Fora. 2019.

SMITH, Steve. **Architecting Modern Web Applications with ASP .NET Core and Microsoft Azure**. Microsoft Corporation, 2017.

RIES, Eric. **The lean startup: How today's entrepreneurs use continuous innovation to create radically successful businesses.** Crown Books, 2011.

SBROCCO, José Henrique Teixeira de Carvalho; MACEDO, Paulo Cesar. **Metodologias ágeis: engenharia de software sob medida.** 1ª ed. São Paulo: érica, 2012

SOUZA, D. Rodrigues. **Implantação da metodologia ágil Scrum em um ambiente de desenvolvimento.** Aranguá, 2014.

SUTHERLAND, Jeff. **The Scrum Papers: Nut, Bolts and Origins of an agile Framework.** version 1.1, Scrum Inc., Cambridge 2012.

STACKOVERFLOW. **Developer Survey Results 2019.** Disponível em: <<https://insights.stackoverflow.com/survey/2019>>. Acesso em: 12 de set. de 2019

WOHLGETHAN, Eric. **Supporting Web Development Decisions by Comparing Three Major JavaScript Frameworks: Angular, React and Vue.js. 2018.** Tese de Doutorado. Hochschule für Angewandte Wissenschaften Hamburg.

XIAOJIE, Yang. **Analysis of DBMS: MySQL Vs PostgreSQL.** Bachelor Thesis. The Bachelor's Thesis Information Technology program, Kemi-Torinio University, Kemi, Finnland, 2011.

ANEXO E – TERMO DE AUTENTICIDADE



UNIVERSIDADE FEDERAL DE JUIZ DE FORA
FACULDADE DE ENGENHARIA

Termo de Declaração de Autenticidade de Autoria

Declaro, sob as penas da lei e para os devidos fins, junto à Universidade Federal de Juiz de Fora, que meu Trabalho de Conclusão de Curso do Curso de Graduação em Engenharia de Produção é original, de minha única e exclusiva autoria. E não se trata de cópia integral ou parcial de textos e trabalhos de autoria de outrem, seja em formato de papel, eletrônico, digital, áudio-visual ou qualquer outro meio.

Declaro ainda ter total conhecimento e compreensão do que é considerado plágio, não apenas a cópia integral do trabalho, mas também de parte dele, inclusive de artigos e/ou parágrafos, sem citação do autor ou de sua fonte.

Declaro, por fim, ter total conhecimento e compreensão das punições decorrentes da prática de plágio, através das sanções civis previstas na lei do direito autoral¹ e criminais previstas no Código Penal², além das cominações administrativas e acadêmicas que poderão resultar em reprovação no Trabalho de Conclusão de Curso.

Juiz de Fora, 17 de março de 2021.

GABRIEL D'AGOSTO PACHE DE FARIA

NOME LEGÍVEL DO ALUNO (A)

2014 49 067

Matrícula

Gabriel Augusto Pachê de Faria

ASSINATURA

019 475 706 41

CPF

¹ LEI N° 9.610, DE 19 DE FEVEREIRO DE 1998. Altera, atualiza e consolida a legislação sobre direitos autorais e dá outras providências.

² Art. 184. Violar direitos de autor e os que lhe são conexos: Pena - detenção, de 3 (três) meses a 1 (um) ano, ou multa.