

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE CIÊNCIA DA COMPUTAÇÃO

RAFAEL PETERSEN

**Templater: extensão do editor *VS Code*
para armazenamento de *snippets* e busca na
base de conhecimentos do *Stack Overflow***

Monografia apresentada como requisito parcial
para a obtenção do grau de Bacharel em Ciência
da Computação

Orientador: Profa. Dra. Renata Galante

Porto Alegre
2019

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. Rui Vicente Oppermann

Vice-Reitora: Prof^a. Jane Fraga Tutikian

Pró-Reitor de Graduação: Prof. Vladimir Pinheiro do Nascimento

Diretora do Instituto de Informática: Prof^a. Carla Maria Dal Sasso Freitas

Coordenador do Curso de Ciência de Computação: Prof. Sérgio Luis Cechin

Bibliotecária-chefe do Instituto de Informática: Beatriz Regina Bastos Haro

AGRADECIMENTOS

Professora Renata Galante por ter disponibilizado seu tempo e atenção no desenvolvimento desta extensão de caráter diferenciado baseado no conceito de busca em base de dados web em larga escala. Aos colegas que se dispuseram a testar e adotar a extensão no seu cotidiano de trabalho. Agradeço imensamente o esforço da minha namorada que conseguiu achar as forças para não me deixar parar o trabalho enquanto estava fazendo seu próprio artigo. Sem esse incentivo nada teria acontecido.

RESUMO

Templates são fundamentalmente trechos de código reutilizáveis que atendem uma funcionalidade simples e específica. Conhecidos como *Snippets* em algumas plataformas de edição de código, o objetivo destes fragmentos é que o programador possa rapidamente incluir um bloco de código e adequar este para o seu caso de uso atual, evitando a necessidade de escrever todo o seu programa do zero. No caso do editor *VS Code* este conceito é chamado *Snippet* e suportado pela plataforma naturalmente, oferecendo a possibilidade de criar e utilizar facilmente estes trechos. Em muitos momentos durante o desenvolvimento de código os programadores encaram dificuldades ao escrever seus programas e precisam buscar soluções para seus problemas, tipicamente em fóruns de sua área de trabalho. Um exemplo deste tipo de fórum é o *Stack Overflow*, que é um fórum aberto onde os usuários cadastrados podem fazer e responder perguntas da área de desenvolvimento de software e computação em geral. O *Stack Overflow* é uma boa fonte de informações, pois mesmo usuários não cadastrados podem pesquisar a base de conhecimentos em busca de questionamentos similares que possam sanar suas necessidades. Nesse contexto, seria muito conveniente para um usuário da ferramenta *VS Code*, por exemplo, que a pesquisa na base de conhecimento pudesse ser realizada através da própria plataforma de edição e que o código apresentado como solução para a pergunta pudesse ser salvo como um *Snippet* para ser reutilizado quando novamente surgir o cenário que causou dificuldades ao programador. O ambiente de edição *VS Code* é extensível através da adição de funcionalidades programadas como pequenos pacotes de ferramentas para que possam ser adicionadas a ele melhorias específicas para facilitar seu uso. Para proporcionar uma experiência melhorada para o usuário, é proposta a extensão *Templater*. *Templater* é uma ferramenta adicional que expande as funcionalidades do editor de texto incluindo pesquisa no fórum e armazenamento facilitado dos trechos selecionados.

Palavras-chave:

VS Code, template, snippet, stackoverflow, stackexchange, extension.

Templater: VS Code extension to search and save code snippets from stackexchange answer base

ABSTRACT

Templates are basically code fragments that can be reused and fulfill a specific function. Known as Snippets in some code editing platforms, the main objective is to provide a kick way to add blocks of code and adapt them to the current needs, avoiding the need to write the whole code from scratch. In the VS Code environment this concept is known as Snippet has native support, offering the possibility to create an use such easily. On the process of developing code, programmers often face challenges and must search for solutions typically on forums in their field of work. A good example of such forums is Stack Overflow, which is an open forum where registered users can ask and answer questions about software development and computing general aspects. Stack Overflow is a good source of information for even unregistered users can search the knowledge base for similar questions to address their needs. In this context it would be convenient for a VS Code user, for instance, to be able to search this knowledge base through the editor and also save the code that was presented as an answer in the form of a Snippet for re-utilization when the difficult scenario arises again. The VS Code environment is fully extensible through the addition of functionalities programmed as small packages of tools to add specific improvements to make it more usable. To provide a better experience for the users, the Templater extension is proposed. Templater is an additional tool that expands the functionalities of the text editor including searching in the forum and easy storage of selected code blocks.

Keywords: *VS Code*, template, snippet, stack overflow, stack exchange, extension.

LISTA DE FIGURAS

Figura 2.1	Interface padrão da ferramenta <i>VS Code</i>	12
Figura 2.2	Extensão desenvolvida como é apresentada na loja.	13
Figura 2.3	Exemplo da formatação de um <i>Snippet</i> para <i>VS Code</i>	14
Figura 2.4	Exemplo de pergunta e resposta disponível na base do Stackoverflow	17
Figura 3.1	Código da extensão <i>Templater</i> contribuindo comandos	21
Figura 3.2	NPM - Módulos da comunidade de desenvolvimento node	22
Figura 4.1	Comandos contribuídos na extensão.....	25
Figura 4.2	Exemplo de utilização do comando de busca	26
Figura 4.3	Exemplo de resposta da busca após seleção	27
Figura 4.4	Exemplo de utilização do comando de armazenamento.....	28
Figura 4.5	Resultado do comando de armazenamento	29
Figura 4.6	Exemplo de utilização do comando de uso.....	30
Figura 4.7	Resultado da aplicação do <i>snippet</i>	31
Figura 5.1	Gráficos de caracterização dos participantes do experimento	34
Figura 5.2	Gráficos de resultados votados nas perguntas negativas.....	36
Figura 5.3	Gráficos de resultados votados nas perguntas positivas.....	37
Figura 5.4	Respostas sobre as vantagens do uso do <i>Templater</i>	38
Figura 5.5	Respostas sobre as desvantagens do uso do <i>Templater</i>	39
Figura 5.6	Respostas sobre as mudanças propostas para o <i>Templater</i>	39

SUMÁRIO

1 INTRODUÇÃO	8
1.1 Objetivos do trabalho	9
1.1.1 Metodologia e detalhamento	10
1.2 Organização do trabalho	10
2 CONTEXTUALIZAÇÃO DAS FERRAMENTAS UTILIZADAS	12
2.1 VS Code	12
2.1.1 Extensões e loja	13
2.1.2 Snippets	14
2.2 Stack Exchange	15
2.2.1 Por que foi escolhido	15
2.2.2 Como é utilizada	16
2.2.3 Exemplo de pergunta no site <i>Stack Overflow</i>	16
2.3 Sumarização dos conceitos	17
3 PROPOSTA: TEMPLATER	19
3.1 Visão Geral	19
3.2 Tecnologias utilizadas	20
3.2.1 Framework de Extensões - <i>VS Code</i>	20
3.2.2 NPM - módulos utilizados	21
3.2.3 Typescript	23
4 TEMPLATER - APLICAÇÃO DESENVOLVIDA	24
4.1 Comandos contribuídos	24
4.2 Como utilizar a aplicação	25
4.2.1 Busca de código	25
4.2.2 Como salvar um <i>snippet</i>	27
4.2.3 Como utilizar um <i>snippet</i> salvo	29
5 AVALIAÇÃO COM USUÁRIOS	32
5.1 Metodologia	32
5.2 Aplicação do método de avaliação	33
5.3 Participantes do experimento	33
5.4 Resultados obtidos das questões objetivas	34
6 CONCLUSÃO	41
6.1 Dificuldades encontradas	41
6.2 Melhorias para o <i>Templater</i>	42
6.3 Trabalhos futuros	43
REFERÊNCIAS	44
APÊNDICE A — QUESTIONÁRIO APLICADO NOS TESTES COM USUÁRIOS	46

1 INTRODUÇÃO

Desenvolvedores convivem com a experiência de buscar ajuda em fóruns de programação para encontrar respostas para suas dúvidas e erros que acabam sendo inseridos no código por falta de algum conhecimento específico. Além disso, a área de trabalho é tão variada e complexa que acaba por exigir dos profissionais um leque ao mesmo tempo muito vasto e muito especializado em cada ponto de conhecimento para a criação de produtos de qualidade. Há muitas informações para coletar que se torna ocasionalmente difícil manter os conhecimentos adquiridos durante a solução de um problema até a próxima vez que o mesmo tipo de dificuldade se repete, e mesmo as coisas mais simples são esquecidas quando se tornam raramente praticadas. Em busca de técnicas para armazenar esses conhecimentos, foi encontrado o conceito de criação de *templates* (MICROSOFT INC., 2019b) reutilizáveis, conhecidos como *snippets* no ambiente de desenvolvimento que se selecionou, *VS Code* (MICROSOFT INC., 2019c), que seriam adequados para a função de guardar exemplos de trechos de códigos com espaço até mesmo para anotar uma descrição da motivação na estrutura de registro destes.

Para a fonte de conhecimentos, por preferência pessoal do autor e popularidade na comunidade, foi escolhido o fórum *Stack Overflow* (H, 2009), uma fonte de conhecimentos baseada em perguntas e respostas dos usuários. O fórum recebe perguntas de todos os níveis de usuários, das mais casuais e simples até às mais específicas e pontuais. As respostas são eleitas pelos autores das perguntas e votadas pela comunidade criando uma rede de confiança nos resultados da solução dada com base na experiência dos usuários. Este é o cenário ideal para buscar as respostas para os questionamentos cotidianos e também os casos de exceção mais discrepantes diretamente das experiências de especialistas dispostos a colaborar.

Existem algumas ferramentas que contribuem separadamente com a pesquisa em fóruns, como por exemplo *Stack Overflow Search* (CREVECOEUR, 2019), e outras que contribuem com pacotes de *snippets* genéricos ou específicos de um contexto de desenvolvimento, como *Angular Snippets (Version 8)* (PAPA, 2019), mas que não direcionam o usuário à criação de seus próprios trechos a seu gosto. Dessa forma não atingem o objetivo que o produto que foi desenvolvido busca: oferecer um caminho de obtenção de respostas mas ao mesmo tempo fornecer uma técnica estável e escalável de armazenamento do aprendizado para fácil reutilização prática.

A atividade de desenvolvimento profissional de código é desempenhada se-

guindo padrões e estruturas comuns globalmente e frequentemente repetitivas. De forma geral, trata-se da composição de comportamentos conhecidos para desempenhar uma tarefa que resolve um problema específico para o projeto do desenvolvedor. Ou seja, aplicação das mesmas estruturas em projetos completamente diferentes, a parte original e criativa deste trabalho está na escolha da abordagem do problema, e não na reinvenção dos padrões e convenções já estabelecidos.

Uma consequência benéfica desta forma de trabalho é que as metodologias de resolução das atividades pontuais que um algoritmo pretende resolver são amplamente aplicadas por uma grande quantidade de programadores com conhecimentos e base diferenciados em uma comunidade que compartilha melhorias e casos de uso para seu próprio crescimento. Isso significa que é possível encontrar muitos exemplos de uso das técnicas que o desenvolvedor escolhe usar e principalmente soluções para os erros e dúvidas mais frequentes.

1.1 Objetivos do trabalho

Sabendo que se conta com uma comunidade ativa, e com foco em um grupo de discussão de desenvolvimento muito conhecido, o propósito da aplicação desenvolvida é proporcionar ao usuário acesso rápido às respostas validadas deste fórum sem necessidade de sair do ambiente de desenvolvimento (o editor de texto escolhido para esta monografia) e ferramentas para salvar a solução com liberdade para adaptá-la ao formato mais conveniente para que possa ser sempre replicada quando necessária em qualquer ponto do algoritmo. Portanto, de forma prática, *Templater* adiciona ao ambiente de desenvolvimento escolhido um par de funcionalidades, uma para oferecer pesquisa por respostas em um fórum conceituado da área de programação; a segunda para oferecer uma forma simplificada de criar estruturas reutilizáveis de código a partir de trechos existentes que pode ser usada tanto nas respostas obtidas da primeira ferramenta como do resultado da adaptação destas ao gosto do usuário e até mesmo trechos de código originais de forma independente da primeira ferramenta.

1.1.1 Metodologia e detalhamento

Para a pesquisa dos exemplos de código e respostas foi escolhida uma comunidade de compartilhamento baseada nos mesmos critérios: Stackoverflow da base de sites do Stackexchange. O modelo de desenvolvimento adotado para a solução foi a criação de uma extensão interativa seguindo os padrões de programação, paradigma e meio de disponibilização da ferramenta apresentados pelos desenvolvedores do editor *VS Code* e esta nova extensão foi elaborada de forma a oferecer ao usuário duas novas contribuições (chamadas de linha de comando) para o ambiente de edição de texto.

A extensão concluída foi disponibilizada publicamente na loja de extensões de forma gratuita para integração ao editor dos usuários.

Testes com usuários foram realizados para aferir a usabilidade e coletar dados para melhoria da solução a partir das sugestões de um grupo de usuários que está dentro do público alvo deste produto: desenvolvedores amadores, estudantes e profissionais do desenvolvimento de software. A avaliação completa e análise de dados foi realizada e está na seção de 5 deste trabalho.

1.2 Organização do trabalho

Neste trabalho apresentam-se os conceitos necessários para o entendimento do que foi desenvolvido. Inicialmente no Capítulo 2, descreve-se o ambiente de edição de código escolhido, *VS Code*, e os conceitos deste: extensões e *snippets*. A seguir neste mesmo capítulo descreve-se o fórum escolhido como fonte dos dados obtidos na extensão e a motivação desta escolha.

No Capítulo 3 são apresentadas as tecnologias utilizadas e os métodos de implementação da extensão **Templater**, objeto de aplicação desta monografia. São apresentados aspectos técnicos e a abordagem da solução.

O Capítulo 4 descreve em detalhes o resultado obtido do esforço de implementação do **Templater** e como pode ser utilizada cada funcionalidade contribuída nesta extensão, mostrando as interfaces criadas para tanto.

Os resultados da avaliação com usuários e a análise dos dados obtidos e como são percebidos pelo autor estão apresentados na avaliação de resultados, indexado no Capítulo 5.

O Capítulo final, de número 6, apresenta a avaliação de sucesso da ferramenta,

pontos de melhoria e fraquezas apontadas pelos resultados da avaliação por usuários e discute as melhorias necessárias e projetos futuros para viabilizar a extensão de forma a contribuir relevantemente para a melhoria da qualidade do trabalho dos usuários.

2 CONTEXTUALIZAÇÃO DAS FERRAMENTAS UTILIZADAS

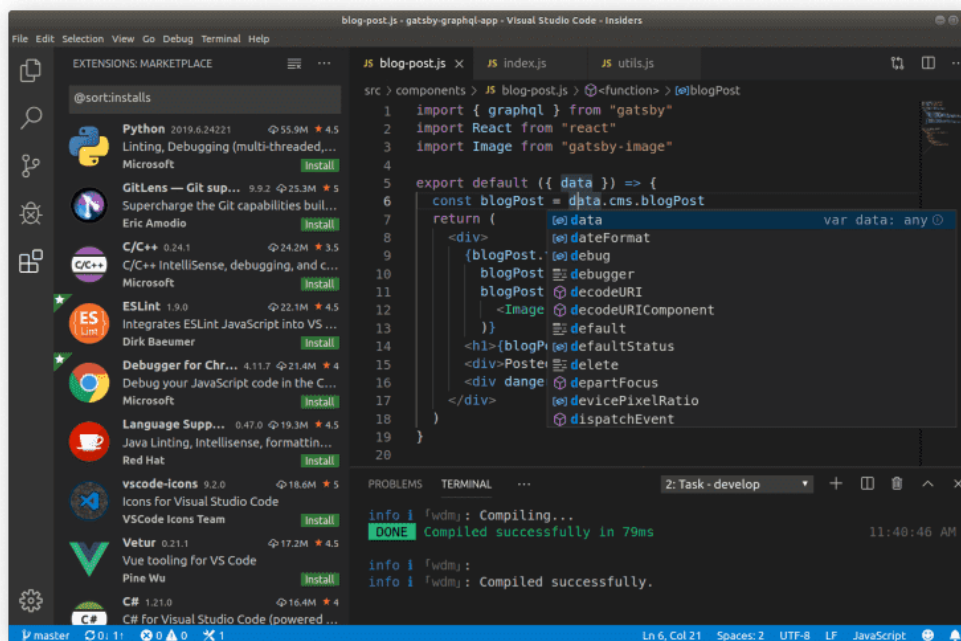
Este capítulo tem como objetivo discutir os principais conceitos usados na proposta deste trabalho e contextualizar o leitor sobre a utilização do editor e sintaxe dos *snippets* de código.

2.1 VS Code

VS Code uma solução multiplataforma da Microsoft (MICROSOFT INC., 2019a) para edição e desenvolvimento de código. A ferramenta é uma aplicação de código aberto (MASSACHUSETTS INSTITUTE OF TECHNOLOGY, 1988) que conta com uma série de recursos de depuração e análise de código fonte que permite que os usuários contribuam com melhorias e extensões para o programa.

O ambiente foi desenvolvido na linguagem TypeScript (MICROSOFT INC., 2012) que é abordada na seção 3.2.3 assim como a maioria das extensões contribuídas para a biblioteca do *VS Code*.

Figura 2.1: Interface padrão da ferramenta *VS Code*



Fonte: Autor

Pode-se observar na Figura 2.1 a interface da ferramenta *VS Code*, com a loja de extensões aberta na esquerda e um exemplo de código na direita. Na parte inferior,

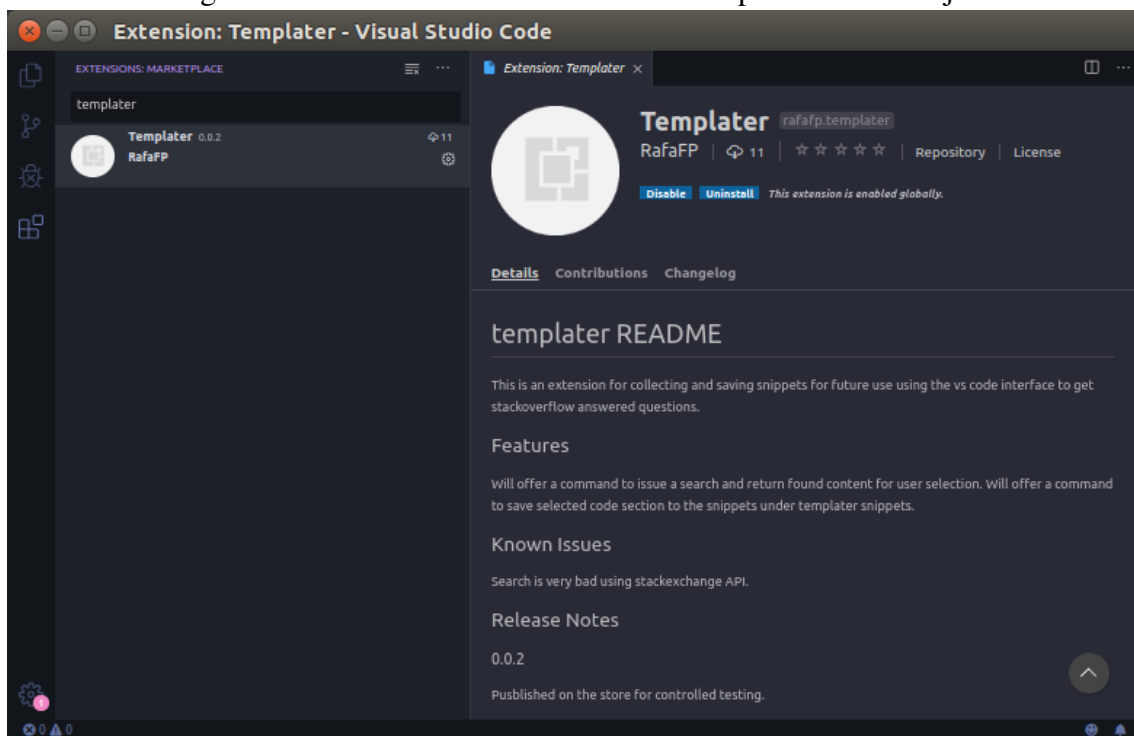
encontra-se o terminal próprio da plataforma.

2.1.1 Extensões e loja

Extensões são conjuntos de comandos contribuídos por desenvolvedores que buscam ampliar as capacidades da interface de desenvolvimento *VS Code*. Uma extensão contribui com múltiplos comandos conforme a necessidade e escopo das novas funcionalidades que o criador decidiu prover em seu programa.

Existem diferentes tipos de extensões que podem ser submetidas para a loja da plataforma, como por exemplo Temas (cores de fundo e fontes), pacotes de atalhos para as ações da IDE, pacotes de *Snippets* criados para um contexto específico e extensões com código próprio para execução de ações especiais que não estão disponíveis na versão original do *VS Code*, sendo este último o tipo de aplicação que foi desenvolvida para esta monografia como pode ser visto na Figura 2.2 que mostra a extensão como é atualmente apresentada na loja.

Figura 2.2: Extensão desenvolvida como é apresentada na loja.



Fonte: Autor

Este tipo de aplicação com código próprio não é nada mais que um programa completo desenvolvido na plataforma **Node.js** e seguindo alguns padrões do *framework*

de desenvolvimento de extensões, como, por exemplo, a criação de um arquivo que defina as chamadas aos comandos contribuídos e como o usuário pode acionar a extensão. Arquivos de configuração à parte, o código é um *Typescript/Javascript* comum e pode se beneficiar de todas as ferramentas e módulos disponíveis na base do **NPM**. O uso de módulos e quais módulos foram utilizados no desenvolvimento deste trabalho serão discutidos em seções no Capítulo 3.

2.1.2 Snippets

Snippets de código são fragmentos sem limitação de tamanho, linguagem e em geral sem limitação de escopo, como é o caso dos *snippets* criados pela aplicação desenvolvida. No ambiente do editor escolhido para desenvolvimento, os *snippets* têm um formato bem definido em uma *metalinguagem*, para a qual o *Templater* é um tradutor/adaptador, minimizando o retrabalho por parte do usuário.

Figura 2.3: Exemplo da formatação de um *Snippet* para *VS Code*

```

home > petersen > .config > Code > User > snippets > {} templater.code-snippets > ...
1  {
2    "Snippet Model": {
3      "prefix": "templater model",
4      "body": [
5        "if((${1:variable}) != 'undefined'){",
6        "  console.log(${1:variable});",
7        "} else {",
8        "  console.log(${2:error})",
9        "  ",
10     ],
11     "description": "a model for you to edit your own snippets"
12   },
13   "Java Loop JSON array": {
14     "prefix": "java loop over json array",
15     "body": [
16       "JSONArray $1:array = jsonObj.getJSONArray(\"dataArray\");",
17       "for (int i = 0 ; i < $1:array.length(); i++) {",
18       "  JSONObject obj = $1:array.getJSONObject(i);",
19       "  String $2:res1 = obj.getString(\"A\");",
20       "  String $3:res2 = obj.getString(\"B\");",
21       "  String $4:res3 = obj.getString(\"C\");",
22       "  System.out.println(A + \" \" + B + \" \" + C);",
23       "}"
24     ],
25     "description": "Simple Java Loop over a JSON array"
26   }
27 }

```

Fonte: Autor

A Figura 2.3 mostra a estrutura básica de um *snippet*. Cada objeto é composto por um nome que deve ser único e indexa o mesmo na lista. No primeiro objeto este seria *"Snippet Model"*. Os objetos têm uma chave *prefix* que representa o atalho que é usado para aplicar o *snippet* durante o desenvolvimento de código. A chave *body* representa o

conteúdo do *snippet* e as ocorrências do padrão $\{\mathbf{1:nome}\}$ são marcações que permitem que no momento da aplicação do *snippet* o usuário substitua ao mesmo tempo todas as ocorrências deste termo por um nome de variável que faça sentido com o contexto aplicado e permita utilizar a tecla *TAB* para alternar entre as diferentes marcações ($\$1$, $\$2$...) e substituir as marcações de mesma chave, todas ao mesmo tempo. Por fim é definida uma chave *description* que é o texto exibido quando o usuário está selecionando o *snippet* a ser usado e portanto deve conter de forma sucinta as informações necessárias para entendimento da utilização deste.

Um tutorial completo sobre o uso deste tipo de padrão pode ser encontrado no site dos desenvolvedores do editor de texto em (MICROSOFT INC., 2019b).

2.2 Stack Exchange

Stack Exchange é uma rede com 174 comunidades que são criadas e mantidas por especialistas e entusiastas de diversas áreas de conhecimento, não limitado à programação (STACK EXCHANGE, 2009). O modo de operação dessas comunidades é um tipo de fórum em que usuários têm liberdade para fazer questionamentos relacionados ao escopo da comunidade e os demais membros da comunidade podem prover respostas para as perguntas que podem, então, ser avaliadas pelo questionador, sendo uma eleita a resposta aceita para resolução do problema que originou a pergunta. Dessa forma, trata-se de uma rede de conhecimentos compartilhada, descentralizada e totalmente mantida pela comunidade.

2.2.1 Por que foi escolhido

A escolha desta plataforma, mais especificamente de um dos fóruns de sua rede de comunidades, deve-se à grande popularidade, comunidade ativa e público alvo da aplicação. De acordo com a própria página de apresentação do *Stack Exchange* e conforme apontado em avaliações do fórum escolhido (MISHRA, 2019), o objetivo é colocar em contato para resolução de problemas pontuais e disseminação de conhecimentos específicos entre os membros da comunidade, sendo estes programadores experientes ou entusiastas de programação que podem se colocar no mercado de trabalho de desenvolvimento de software, ou seja, pessoas com nível de conhecimento específico em sua área que são

capazes de dar respostas avançadas para dúvidas e problemas que já aprenderam como vencer com suas experiências.

2.2.2 Como é utilizada

A plataforma oferece uma *API* de pesquisa da sua base de perguntas e respostas que é utilizada para obter uma lista de perguntas relacionadas com os termos que o usuário da aplicação deve informar para realizar sua pesquisa. Neste trabalho, usa-se fundamentalmente a funcionalidade de busca de perguntas por termos no título. A *API* não oferece muita flexibilidade para a busca e procura pelos termos definidos sequencialmente, portanto, uma pesquisa exata da frase definida pelo usuário, que é bastante limitante do ponto de vista de encontrar respostas similares cuja pergunta foi escrita com termos similares. A questão de falta de flexibilidade é discutida nos fóruns da plataforma e foi uma decisão adotada nas versões mais recentes da ferramenta para tornar a busca mais rápida devido à grande quantidade de perguntas disponíveis na base de dados do *Stack Overflow*, que é o fórum utilizado neste projeto.

A busca realizada tem seus resultados filtrados para apresentar ao usuário apenas perguntas que tiveram uma resposta aceita pelo seu criador e esta resposta é então buscada na base de dados para compor o resultado da pesquisa. A resposta da pesquisa do usuário desta extensão é uma lista dos títulos das perguntas para que possa ser escolhido o mais adequado para responder à sua dúvida. Quando uma pergunta é escolhida, ela é apresentada em uma janela de texto do editor com o título, corpo e resposta aceita em forma de texto simples para que o usuário possa continuar o uso da extensão selecionando a parte que interessar à criação de um *snippet* de código.

2.2.3 Exemplo de pergunta no site *Stack Overflow*

A Figura 2.4 ilustra o formato de interação entre os usuários como é abordado pela interface *web* do fórum *Stack Overflow*. Um usuário faz uma pergunta que inclui um título curto apresentando o escopo da pergunta e um corpo explicando o contexto específico em que precisa aplicar o conhecimento buscado. Usuários que tiverem interesse em compartilhar de sua experiência oferecem respostas na próxima seção e uma destas respostas é aceita pelo criador da pergunta como a solução escolhida, o que fica denotado

na imagem pelo "V" verde ao lado da resposta no topo da lista.

Tudo que o usuário precisa para receber uma resposta do *Templater* no campo de busca pode ser simplificado em:

- Encontrar múltiplos títulos correspondentes com a busca;
- Ter um dos títulos selecionado pelo usuário;
- Apresentar o texto da pergunta de um usuário do *Stack Overflow*;
- Apresentar o texto da resposta considerada solução do problema.

Este comportamento é idêntico ao caso de uso de uma busca feita manualmente no fórum pelo usuário de nossa aplicação.

Figura 2.4: Exemplo de pergunta e resposta disponível na base do Stackoverflow

The screenshot shows a Stack Overflow page for the question "How do I compare strings in Java?". The question text is: "I've been using the == operator in my program to compare all my strings so far. However, I ran into a bug, changed one of them into .equals() instead, and it fixed the bug. Is == bad? When should it and should it not be used? What's the difference?". The question has 726 votes and is tagged with 'java', 'string', and 'equality'. The top answer, by user 'Nathan H', is marked as the correct solution with a green checkmark. The answer text is: "== tests for reference equality (whether they are the same object). equals() tests for value equality (whether they are logically 'equal'). Objects.equals() checks for null before calling .equals() so you don't have to (available as of JDK7, also available in Guava). String contentEquals() compares the content of the String with the content of any CharSequence (available since Java 1.5). Consequently, if you want to test whether two strings have the same value you will probably want to use Objects.equals().". The page also features a sidebar with navigation links and a right-hand sidebar with a blog and remote jobs section.

Fonte: Autor

2.3 Sumarização dos conceitos

Neste capítulo foram apresentados os conceitos básicos e o ferramental utilizado pelo *Templater* para desempenhar suas funcionalidades. Foi apresentado o ambiente de edição de código *VS Code*, bem como uma visão do que são extensões e *snippets*. Foi contextualizada também a base de conhecimentos *Stack Overflow*, pertencente ao grupo *Stack Exchange* e apresentados exemplos de perguntas como são feitas no site.

Dados os conceitos necessários como finalizados e considerando que foi feita a devida apresentação do ferramental utilizado, passa-se a descrição da proposta de desen-

volvimento do *Templater*.

3 PROPOSTA: TEMPLATER

Este capítulo tem como objetivo descrever as tecnologias e módulos utilizados no desenvolvimento desta aplicação. A seção 3.1 apresenta uma contextualização do que foi elaborado para implementação da extensão. A seção 3.2 apresenta as tecnologias utilizadas em maior detalhamento.

3.1 Visão Geral

Considerando o cenário apresentado: um editor de código utilizado amplamente para desenvolvimento de aplicações em múltiplas linguagens por programadores que com certa frequência enfrentam dificuldades no desenvolvimento que levam o usuário a fazer buscas por respostas em uma ferramenta externa. Ao obter respostas sobre a implementação o programador aprende a utilizar trechos de código que se repetirão durante a atividade de programação e podem voltar a ser utilizados no futuro de forma idêntica.

A proposta deste projeto é a implementação de uma ferramenta interna de busca utilizando um padrão conhecido do ambiente de desenvolvimento, que é a chamada de comandos através da paleta de comandos (*control + shift + P*). A segunda ferramenta implementada trata do armazenamento dos exemplos buscados em formato de *snippets* de código que se tornam acessíveis para aplicação em qualquer editor do ambiente.

Este projeto busca adicionar uma extensão ao ambiente de desenvolvimento a ser disponibilizada gratuitamente sob a licença *MIT* (MASSACHUSETTS INSTITUTE OF TECHNOLOGY, 1988) na loja de extensões da plataforma de forma compatível com os sistemas operacionais mais relevantes (Linux, Windows, MacOS) que cubra as funcionalidades de busca e armazenamento. O limite planejado do escopo de busca é o uso de pesquisa sequencial de palavras na base de títulos de perguntas do *Stack Overflow* que já tiveram uma resposta aprovada. Pretende-se oferecer uma lista de bom tamanho (cinquenta itens se existirem) e filtrável de respostas encontradas e exibir todo o conteúdo da pergunta e resposta selecionada em uma nova aba do editor *VS Code*. A ferramenta de armazenamento planejada registra os trechos em um arquivo local de *snippets*, de acordo com o formato de criação apresentado no manual do ambiente de desenvolvimento com *placeholders* para os campos que o usuário deve modificar e um exemplo no início do arquivo mostrando a estrutura básica do *snippet*.

A aplicação de um *snippet* já é nativamente garantida pelo editor de código.

Desta maneira, a forma de uso no texto deste projeto é adequar as implementações originais para que exista plena compatibilidade com o sistema nativo.

Por fim, as ferramentas criadas são comandos completamente independentes para que seja possível atender usuários que optarem por usar apenas uma das duas funcionalidades conforme for mais adequado e vantajoso para seu trabalho. Assim, viabiliza-se também de forma natural a pesquisa por perguntas cujas respostas são esclarecimentos de dúvidas teóricas que não necessariamente contém código apresentado e também uma ferramenta de criação rápida de *snippets* com base no código do usuário.

3.2 Tecnologias utilizadas

3.2.1 Framework de Extensões - VS Code

A primeira parte deste desenvolvimento foi a pesquisa para entender o funcionamento do ambiente de extensões da plataforma. Para atingir o mínimo de entendimento necessário para iniciar o trabalho, foram seguidos dois passos: o primeiro sendo o básico de entendimento da criação de um novo componente simples para o *VS Code* - (MICROSOFT INC., 2019e) e o segundo uma abordagem um pouco mais avançada com interação com sites através do uso de *webviews* - (MICROSOFT INC., 2019d).

Em uma abordagem inicial do projeto foi considerado o uso de *iframes* (MDN WEB DOCS, 2019) por meio da *API* de *webviews* apresentada no tutorial que foi utilizado como base para o início do desenvolvimento. Esta abordagem se mostrou muito pouco adequada por ser muito limitada no modo de interação da plataforma *VS Code* com páginas exibidas diretamente através de *iframes* em sua interface e o péssimo desempenho resultante do encapsulamento da página no editor de texto.

Adotou-se, então, outra alternativa para a coleta de dados: a busca através da *API* do fórum *Stack Overflow*. Com este método, foi implementado em *TypeScript* no código do *Templater* um algoritmo de busca de texto simples nos títulos de perguntas da base do fórum que apresenta para o usuário uma lista dos primeiros cinquenta resultados com respostas aceitas (perguntas que já completaram o seu fluxo no fórum) para que este possa visualizar o conteúdo do título que lhe parecer mais adequado para sanar sua dúvida. Desta forma, fundamentalmente foi implementado um buscador interno ao editor de texto para conteúdo do *Stack Overflow* que pode ser usado para leitura de respostas no próprio ambiente de trabalho do usuário.

Respostas obtidas que contenham código de exemplo para uma situação de questionamento sobre implementação em linguagem específica são candidatas à criação de um *snippet* de código para uso recorrente no editor. Para este caso foi desenvolvida uma solução de geração automatizada no padrão de *snippets* do *VS Code* com o conteúdo selecionado pelo usuário. Desta forma, possibilita-se a rápida criação e edição do código necessário para que o usuário tenha seu *snippet* personalizado e pronto para uso assim que encontrar o que precisa.

O desenvolvimento de extensões para *VS Code* conta com um *framework* específico que cria vínculos entre os comportamentos criados e chamadas de comando da interface do editor de texto. Esse tipo de ligação é chamada de **contribuição** neste contexto. A Figura 3.1 mostra os comandos contribuídos e a importação do *framework* e dos códigos desenvolvidos para estes comandos.

Figura 3.1: Código da extensão *Templater* contribuindo comandos

```

TS extension.ts ×
rafaelpetersen > templater > src > TS extension.ts > activate
1  import * as vscode from 'vscode';
2  import { multipick } from './multistepinput';
3  import * as snippet from './snippet';
4
5  export function activate(context: vscode.ExtensionContext) {
6      context.subscriptions.push(
7          vscode.commands.registerCommand('searchPage.start', () => {
8              const answer = multipick();
9          }),
10         vscode.commands.registerCommand('saveSnippet.edit', () => {
11             const editor = vscode.window.activeTextEditor;
12             if (vscode.window.activeTextEditor && editor !== undefined && editor.document.getText(editor.selection)) {
13                 let text = editor.document.getText(editor.selection);
14                 snippet.appendAndOpenFile(text);
15             } else {
16                 vscode.window.showInformationMessage("Select code for snippet")
17             }
18             //snippet.log();
19         })
20     )
21 }

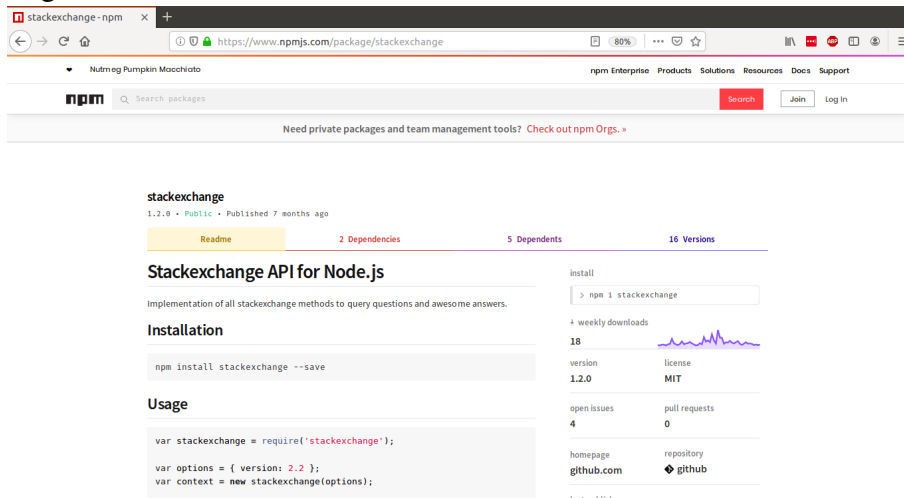
```

Fonte: Autor

3.2.2 NPM - módulos utilizados

Além do código original criado para esta aplicação, foram utilizados pacotes de código disponibilizados no *NPM*. *NPM* é um repositório de pacotes que são, de forma geral, códigos para resolução de cenários simples como eliminação de texto codificado para HTML e ferramentas de integração com *APIs*. A Figura 3.2 apresenta a interface do *NPM* em um dos módulos que foram utilizados.

Figura 3.2: NPM - Módulos da comunidade de desenvolvimento node



Fonte: Autor

Para o desenvolvimento desta ferramenta foram utilizados três pacotes além dos que são fundamentais para a inclusão das dependências do *framework*. Os pacotes utilizados foram *appdata-path* (SAMBORSKI, 2019), *striptags* (NORRIS, 2019) e *stackexchange* (SWAAGMAN, 2019). Todos os módulos utilizados, bem como o código desenvolvido estão sob a licença *MIT* (MASSACHUSETTS INSTITUTE OF TECHNOLOGY, 1988) e portanto podem ser livremente utilizados, editados e integrados a projetos por qualquer portador de uma cópia dos projetos.

O módulo *striptags* foi utilizado para remover elementos de HTML dos dados coletados no fórum. O texto obtido utiliza elementos de *markdown* e algumas *tags* para identificar *hiperlinks* e áreas de código que prejudicam a representação como texto puro no editor. Para o usuário final o ideal é ter um conteúdo livre deste tipo de elemento para criação de seus *snippets*. A ferramenta é utilizada no contexto da exibição para o usuário apenas e se encarrega destes ajustes através do comando *striptags*, que remove todas as ocorrências destes marcadores.

O segundo módulo utilizado é o *appdata-path*. Este é responsável por verificar qual é o sistema operacional do usuário e retornar o caminho para os arquivos de configuração de programas *default* para o mesmo. A intenção do uso deste módulo é compatibilizar com qualquer sistema operacional o uso da ferramenta, já que os arquivos de *snippets* são salvos na pasta de configurações do *VS Code*, que varia de localização para cada sistema. A utilização deste módulo se dá no momento em que é necessário salvar um novo *snippet*, sendo o responsável por encontrar o caminho até a pasta em que o documento deve estar para que possa ser usado como fonte de *snippets* quando este for

solicitado posteriormente pelo usuário.

Para realizar as buscas de dados remotos na base do fórum escolhido a própria organização responsável fornece *endpoints* para obtenção de dados através de acesso *HTTP*. Para abstrair a construção de endereços de busca foi utilizado o módulo *stackexchange* que se encarrega de montar as *URLs* de acordo com o padrão descrito na documentação da plataforma. Este módulo é utilizado para obter o conteúdo puro das respostas às pesquisas necessárias à aplicação.

3.2.3 Typescript

TypeScript foi a linguagem escolhida para o desenvolvimento da aplicação por ser o padrão preferencial para o desenvolvimento de extensões para o editor de texto. Esta linguagem é uma variação de JavaScript desenvolvida pela empresa *Microsoft* para garantir regras mais restritivas de tipagem de variáveis e funções. Esse padrão de linguagem é transpilado (SENGSTACKE, 2019) para JavaScript e visa reforçar a tipagem e oferecer uma série de recursos conhecidos de orientação a objetos, tais como a definição de classes e de funções anônimas. O esforço de uso é voltado para manutenibilidade e organização do código além de melhorar significativamente os testes unitários do código desenvolvido.

Para este projeto a escolha desta linguagem foi guiada pela vasta maioria de exemplos e larga utilização da linguagem neste tipo de desenvolvimento bem como a melhoria de qualidade e legibilidade que estão atreladas ao uso da mesma. Dois dos módulos utilizados já contavam com um arquivo de tipagem do projeto definido pelo esforço da comunidade de desenvolvedores Typescript para converter projetos puramente desenvolvidos em JavaScript para contar com um esquema de tipagem de variáveis, funções e seus argumentos. O *Templater* está disponível em um repositório aberto e conta com uma página de orientações (HOME. . . , 2019) e é uma forma de adaptar sem alterações no código original para que possa ser usado sem violar as restrições de Typescript. Para o módulo que não contava com a adaptação, foram aplicadas tipagens para as funções utilizadas no código desenvolvido diretamente, considerando que se utilizam apenas alguns dos muitos recursos oferecidos pelo projeto.

No tutorial de desenvolvimento de extensões fornecido pelo *VS Code* consta um parágrafo que cita JavaScript como uma opção alternativa para desenvolvimento e que todos os tutoriais e documentação serão providos em TypeScript devido à melhor experiência de desenvolvimento (MICROSOFT INC., 2019f).

4 TEMPLATER - APLICAÇÃO DESENVOLVIDA

Neste capítulo é feita uma apresentação dos resultados deste desenvolvimento, verificando os passos fundamentais e interface da aplicação desenvolvida em cada seção que segue descrevendo a funcionalidade implementada e forma de uso.

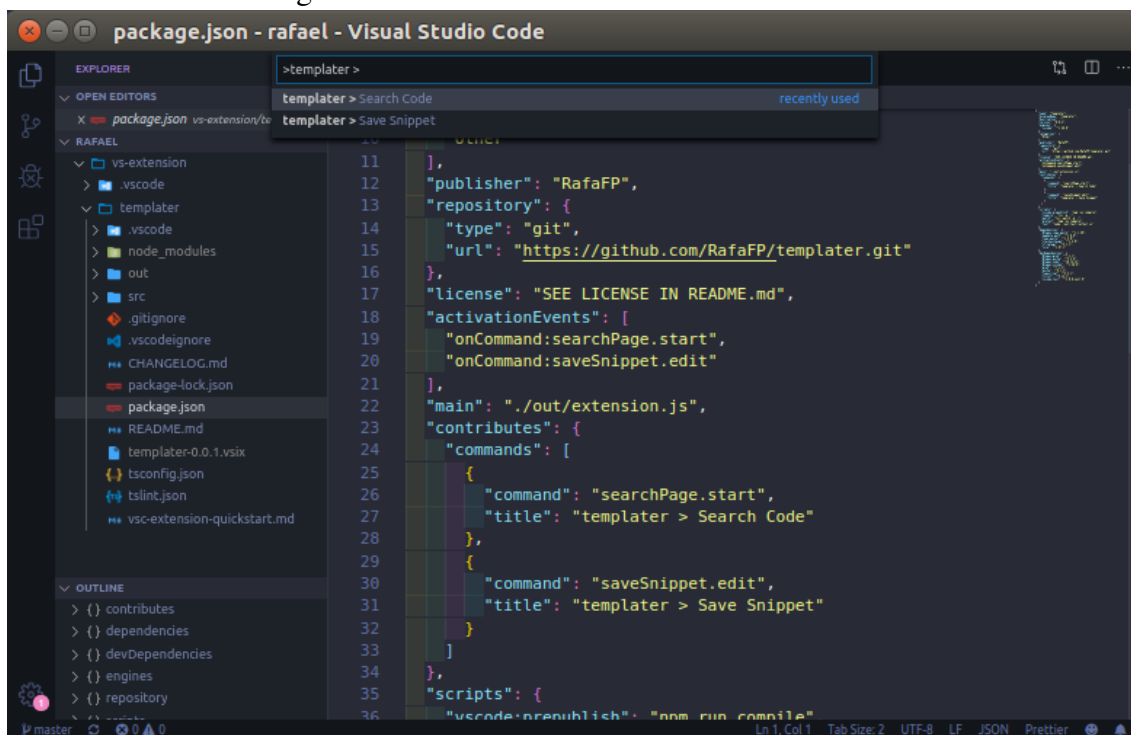
4.1 Comandos contribuídos

Conforme citado na seção 3.1, existem comandos contribuídos por uma extensão que ficam disponíveis na interface de comandos do editor de texto. A extensão contribui dois comandos para o usuário: *Search Code* que é responsável pela busca e apresentação de lista de opções e *Save Snippet* que cria um novo *snippet* de código a partir da seleção de texto do usuário.

Search Code conta com o apoio dos pacotes de busca na API do fórum *Stack Overflow* para gerar uma lista de perguntas apresentada ao usuário. Após a seleção, este mesmo comando realiza a extração dos dados da pergunta e resposta, formata adequadamente e abre uma nova janela de texto com o conteúdo obtido. *Save Snippet* utiliza o módulo de localização de pastas para criar um arquivo de *snippets* na pasta correta ou localizar o arquivo já criado e insere uma nova instância no padrão correto para que o usuário realize pequenos passos de customização (nomeação e ajustes do código salvo) para deixar o código pronto para uso.

Na Figura 4.1, é possível ver no menu de comandos do *VS Code* os dois comandos apresentados pela extensão. São comandos independentes, ou seja, um usuário pode realizar uma pesquisa sem criar um *snippet* caso deseje apenas esclarecer sua dúvida, da mesma forma um *snippet* pode ser criado partindo de código original criado pelo usuário sem necessidade de pesquisa.

Figura 4.1: Comandos contribuídos na extensão



Fonte: Autor

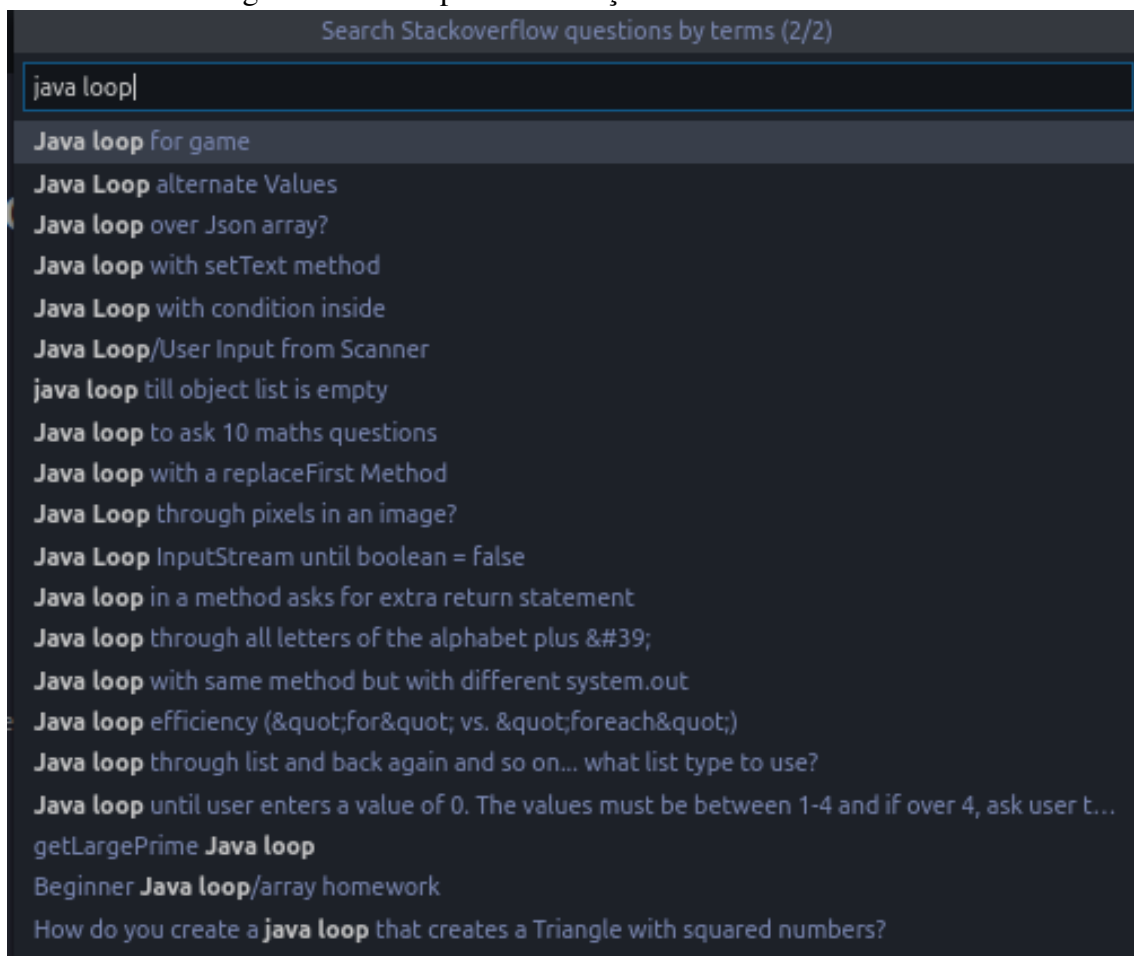
4.2 Como utilizar a aplicação

4.2.1 Busca de código

A primeira funcionalidade da extensão desenvolvida trata-se da ferramenta de busca de código no *Stack Overflow* que é acessada pelo usuário no menu de comandos. Assim que o usuário escreve os termos de pesquisa é realizada uma busca pelas 50 primeiras perguntas que já foram respondidas e os títulos destas são apresentados em uma lista para que o usuário selecione a que lhe parecer mais adequada. Conforme apresentado no cenário da Figura 4.2 o usuário recebe a lista de respostas contendo os termos pesquisados e pode usar o campo de pesquisa para buscar por um termo chave na lista de respostas.

É importante citar que os resultados apresentados são ordenados de acordo com a data da última atualização. Este foi o critério escolhido para oferecer ao usuário os resultados mais recentes por assumir que estes estarão mais bem contextualizados em um ambiente mutável como é a área do desenvolvimento de código.

Figura 4.2: Exemplo de utilização do comando de busca



Fonte: Autor

Quando o usuário seleciona uma das perguntas ele é direcionado a uma nova aba do editor contendo o resultado obtido pela ferramenta, apresentando toda a pergunta original como foi feita no fórum e a resposta selecionada pelo autor da pergunta como solução para o problema. Para simplificação do uso, todo o conteúdo é convertido em forma de texto puro, sem o *markdown* apresentado na versão original do *Stack Overflow*, como pode ser visto na Figura 4.3 apresentada anteriormente. Neste momento, a busca está concluída e o usuário pode coletar as informações que precisa, fazer uma cópia do código apresentado, se houver, e aplicar o que aprendeu no trabalho que está desenvolvendo.

Figura 4.3: Exemplo de resposta da busca após seleção

```

1 QUESTION: #####
2
3 Java loop with setText method
4
5 I want to make a loading screen for my application, but the JLabel which shows how much loaded
6 doesn't working properly.
7
8 I created a JDialog and JLabel. I want to update JLabel text via a Thread.
9
10 JDialog loadingScreen = new JDialog(JFrame null);
11 JLabel loaded = new JLabel();
12 loadingScreen.add(loaded);
13 loading.add(time);
14 loading.setSize(new Dimension(200,100));
15
16 Thread t = new Thread() -> {
17     loading.setVisible(true);
18     while (true) {
19         if (i3 &lt; Math.pow(characters.size(), Integer.parseInt(length.getText()))) {
20             time.setText(i3 + " / " + Math.pow(characters.size(), Integer.parseInt(length.getText(
21             ())), //doesn't work
22             System.out.println("hi"); //works
23         } else break;
24     }
25 }
26
27 ...
28 t.start();
29
30 ACCEPTED ANSWER: #####
31
32 Don't update Swing components via a Thread. Instead, use <a href="https://docs.oracle.com/javase/
33 8/docs/api/javaw/swing/SwingUtilities.html#invokeLater-java.lang.
34 noreferrer">SwingUtilities.invokeLater</a>, like so:

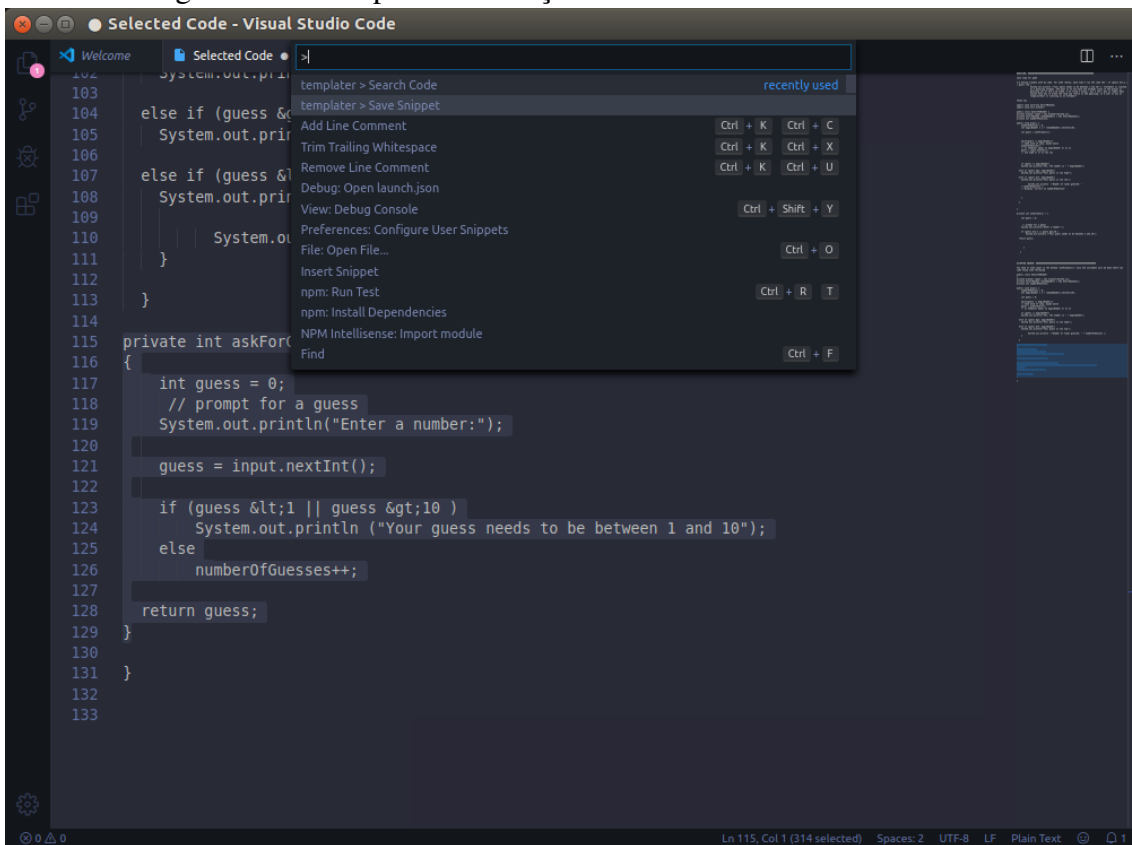
```

Fonte: Autor

4.2.2 Como salvar um *snippet*

O comando de armazenamento chamado *Save Snippet* é utilizado através da mesma interface de comandos do *VS Code* de uma forma diferente da ferramenta apresentada anteriormente. Este comando interage diretamente com o texto selecionado pelo usuário, ou seja, pode ser utilizado em qualquer texto sendo visualizado através do editor para criar um *snippet* de código a partir da seleção de texto atual. Pode-se ver na Figura 4.4 um usuário ativando o comando de armazenamento sobre um texto que está selecionado no editor que aparece ao fundo da interface de comandos.

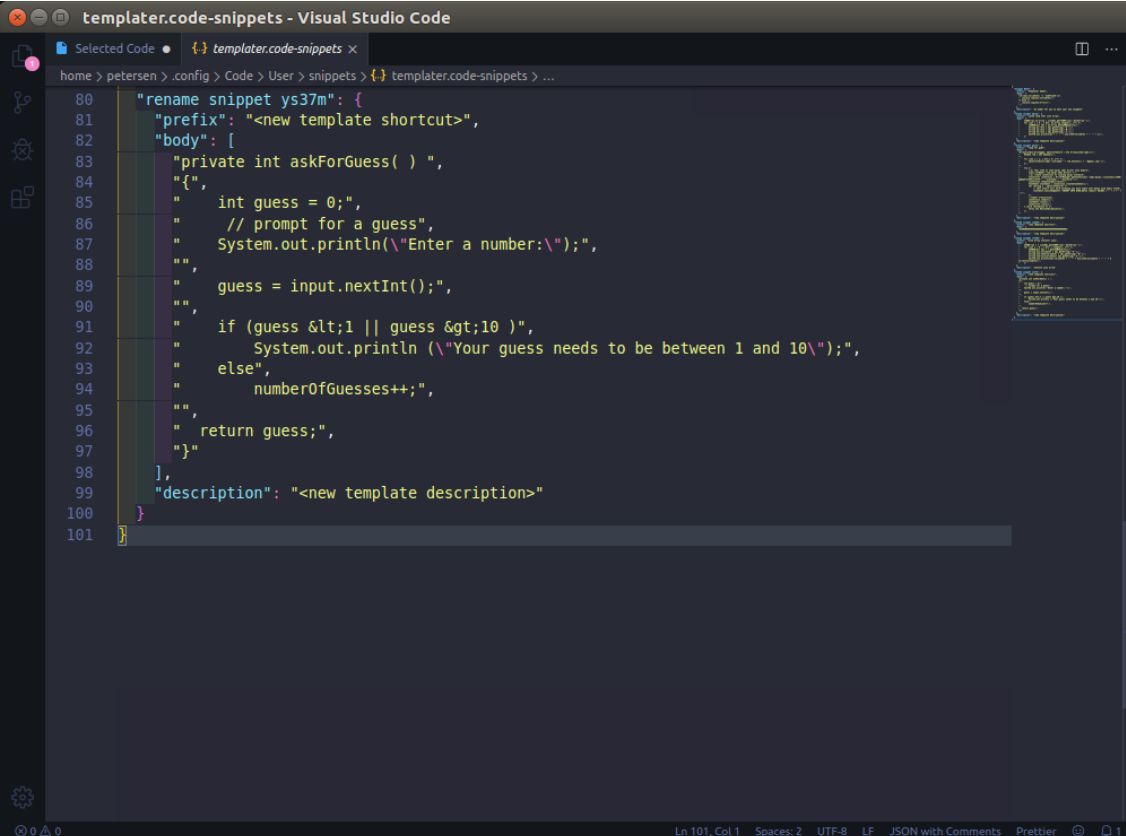
Figura 4.4: Exemplo de utilização do comando de armazenamento



Fonte: Autor

Na Figura 4.5 é apresentado o arquivo de *snippets* do *Templater*, que é aberto automaticamente com o novo trecho de código salvo no formato padrão com *placeholders* indicando para o usuário onde ele deve realizar as customizações para uso do novo código salvo. Neste ponto, é esperado que o usuário modifique o campo com chave *prefix* para nomear o *snippet* (define o nome que é usado para acessar este) e também o campo de chave *description* para adicionar algum indicativo que torne mais fácil lembrar a função do novo trecho salvo no futuro. Além disto, nesse momento, o usuário tem a oportunidade de modificar o código para que se adeque a suas preferências e criar *placeholders* conforme discutido na seção que trata de *snippets* deste trabalho.

Figura 4.5: Resultado do comando de armazenamento



```

80  "rename snippet ys37m": {
81    "prefix": "<new template shortcut>",
82    "body": [
83      "private int askForGuess( ) ",
84      "{",
85      "    int guess = 0;",
86      "    // prompt for a guess",
87      "    System.out.println(\"Enter a number:\");",
88      "",
89      "    guess = input.nextInt();",
90      "",
91      "    if (guess <1 || guess >10 )",
92      "        System.out.println (\"Your guess needs to be between 1 and 10\");",
93      "    else",
94      "        numberOfGuesses++;",
95      "",
96      "    return guess;",
97      "}"
98    ],
99    "description": "<new template description>"
100 }
101 ]

```

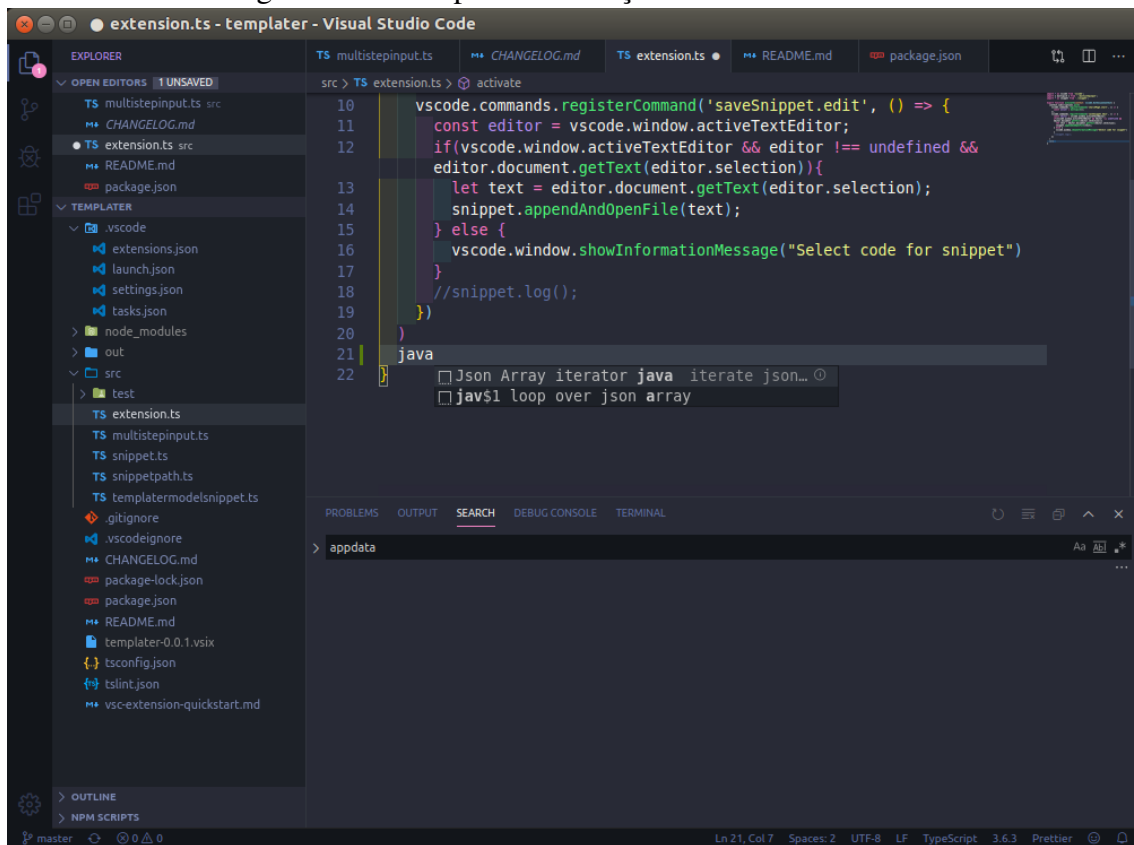
Fonte: Autor

Isto conclui o funcionamento da segunda funcionalidade implementada para o *Templater* que, da mesma forma que a primeira, pode ser utilizada livremente em qualquer ambiente de edição do *VS Code* sem necessidade de vínculo com a pesquisa, viabilizando, portanto, que um usuário salve seus próprios trechos de código recorrentes para facilitar a repetição destes em outros projetos.

4.2.3 Como utilizar um *snippet* salvo

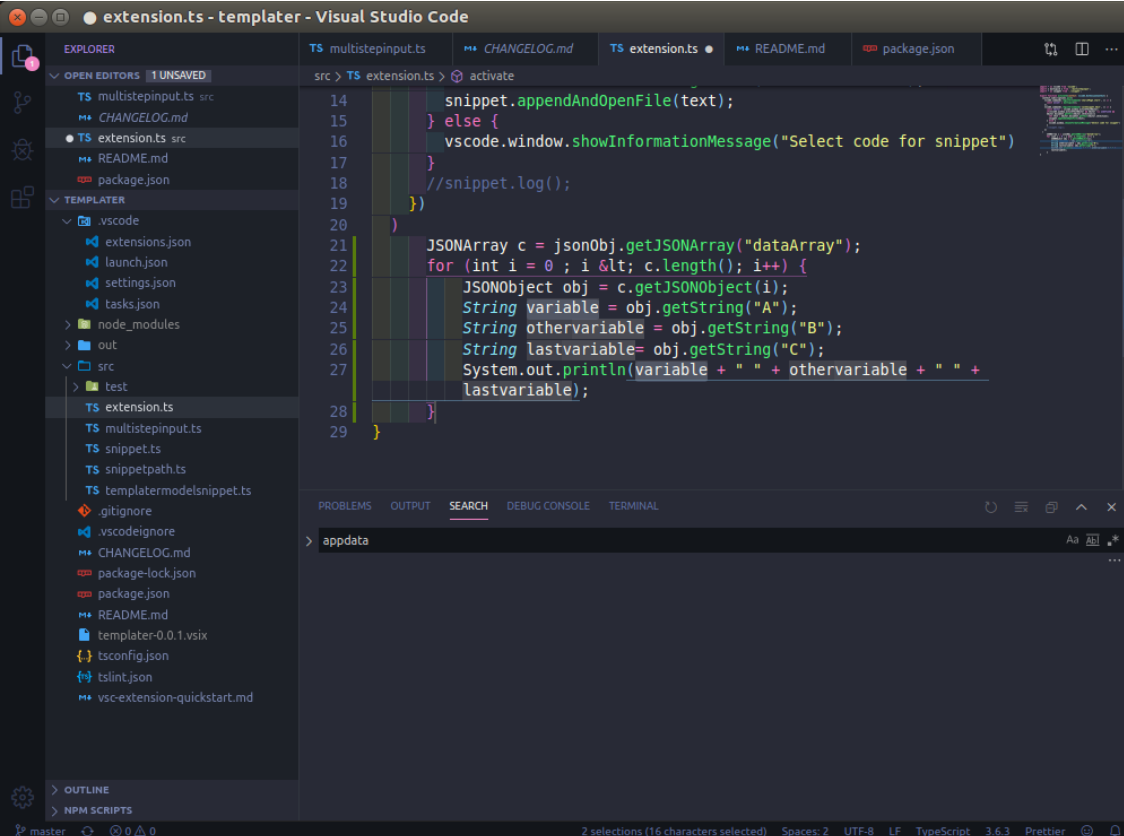
IntelliSense é o nome dado ao conjunto de ferramentas de sugestão de texto e assistência ao desenvolvimento de forma geral do *VS Code*. É através deste conjunto de ferramentas que o nosso *snippet* salvo torna-se acessível para aplicação no código desenvolvido. A Figura 4.6 mostra como é apresentada a lista de sugestões do *IntelliSense*: quando o usuário pressiona as teclas *HOME* + barra de espaço (*control* + barra de espaço em um MacOS) o editor de código vai apresentar uma lista dos *snippets* aplicáveis que realiza filtragem baseada no texto que o usuário escreve após iniciar o comando. Esta lista permite alternar entre as opções com uso das setas direcionais para seleção do *snippet*.

Figura 4.6: Exemplo de utilização do comando de uso



Fonte: Autor

Quando decidir por um *snippet* para aplicar, o usuário só precisa confirmar a seleção pressionando a tecla *enter* e é apresentado o trecho de código na posição e indentação atual do cursor como é possível verificar na Figura 4.7. Pode-se perceber também nesta figura que os textos que anteriormente haviam sido marcado com as expressões do tipo `_${1:nome}` estão destacados, significando que estão prontos para a substituição por um nome que sirva ao contexto em que estão sendo inseridos. A primeira marcação é chamada "*variable*" e quando o usuário começar a digitar, todas as ocorrências são simultaneamente apagadas e substituídas pela nova entrada. Para iterar entre as diferentes marcações, o usuário utiliza a tecla *tab*, podendo assim adequar de forma rápida todas as variáveis ao novo contexto de uso do *snippet*.

Figura 4.7: Resultado da aplicação do *snippet*

```
src > TS extension.ts > activate
14 snippet.appendAndOpenFile(text);
15 } else {
16     vscode.window.showInformationMessage("Select code for snippet")
17 }
18 //snippet.log();
19 })
20 )
21
22 JSONArray c = jsonObj.getJSONArray("dataArray");
23 for (int i = 0 ; i <t; c.length(); i++) {
24     JSONObject obj = c.getJSONObject(i);
25     String variable = obj.getString("A");
26     String othervariable = obj.getString("B");
27     String lastvariable= obj.getString("C");
28     System.out.println(variable + " " + othervariable + " " +
29     lastvariable);
30 }
```

PROBLEMS OUTPUT SEARCH DEBUG CONSOLE TERMINAL

> appdata

Fonte: Autor

Isto encerra a seção de uso da aplicação apresentando um comportamento nativo do editor de código (o *IntelliSense*) e dois comportamentos criados pela aplicação, dos quais um tem interação com o primeiro por meio da criação de *snippets* de código (ou texto em geral) para apresentação na sua lista de sugestões.

As funcionalidades de busca e armazenamento de exemplos foram devidamente apresentadas e, no próximo capítulo, apresentam as percepções dos usuários que se dispuseram a testar esta aplicação e considerações finais.

5 AVALIAÇÃO COM USUÁRIOS

A aplicação *Templater* está disponível na loja de extensões do *VS Code* e disponível gratuitamente na tela de extensões do editor, conforme visto na Figura 2.2, para uso desde o início do período de avaliação da ferramenta. No momento, após o experimento, a extensão já foi adquirida vinte e cinco vezes por usuários diferentes e instalada em seus ambientes de desenvolvimento. Este capítulo abordará os resultados obtidos seguindo um sistema conhecido por Escala de Usabilidade do Sistema e apresenta os cálculos correspondentes para cada resposta, compondo o escore de usabilidade, além de explorar as respostas descritivas dos usuários de forma crítica sobre o comportamento do sistema e expectativas destes.

5.1 Metodologia

Para a avaliação foi utilizada uma Escala de Usabilidade de Sistema *Likert* (LIKERT, 1932) onde o usuário escolhe entre "discordo plenamente", "discordo parcialmente", "indiferente", "concordo parcialmente", "concordo plenamente" em doze perguntas objetivas e também responde três perguntas dissertativas para avaliação da sua percepção do sistema utilizado. Esta escala foi escolhida por estar de acordo e ser amplamente utilizada como método de avaliação de aplicações pela indústria de software e dada sua capacidade de gerar resultados tangíveis e mensuráveis com uma amostra de resultados pequena gerando uma métrica aceitável da usabilidade da ferramenta avaliada.

Na avaliação foram utilizadas perguntas positivas, em que a concordância significa que o sistema apresenta-se utilizável e perguntas negativas, onde a conclusão de "concordo plenamente" significa que o usuário teve muitas dificuldades ou não gostou da forma que o sistema apresentou uma funcionalidade. Tais métricas são somadas usando de forma positiva ou negativa utilizando o método deste sistema de avaliação para gerar resultados para cada questionamento e finalmente uma pontuação de usabilidade geral da extensão desenvolvida. Este método deve gerar uma métrica da usabilidade com base em três pilares: efetividade, eficiência e satisfação. Respectivamente está tratando da habilidade de completar o objetivo, esforço e recursos necessários para atingir o objetivo e satisfação do usuário com a experiência de uso. As perguntas são apresentadas homogeneamente sem uma definição explícita do pilar que estão visando avaliar para tornar a avaliação fluida e natural.

5.2 Aplicação do método de avaliação

Dos usuários que baixaram a extensão na loja, dezesseis foram recrutados para fazer parte do experimento através da sua divulgação no mural de mensagens da empresa em que o autor trabalha e convite a conhecidos que atuam em áreas de tecnologia ou pretendem atuar no futuro. O grupo de avaliação foi escolhido buscando o perfil de usuário que utiliza ferramentas de edição de código no seu cotidiano e desenvolve código nestas ferramentas, portanto um grupo similar ao que se espera que utilize fóruns de soluções de código como o que foi selecionado para o desenvolvimento do *Templater*.

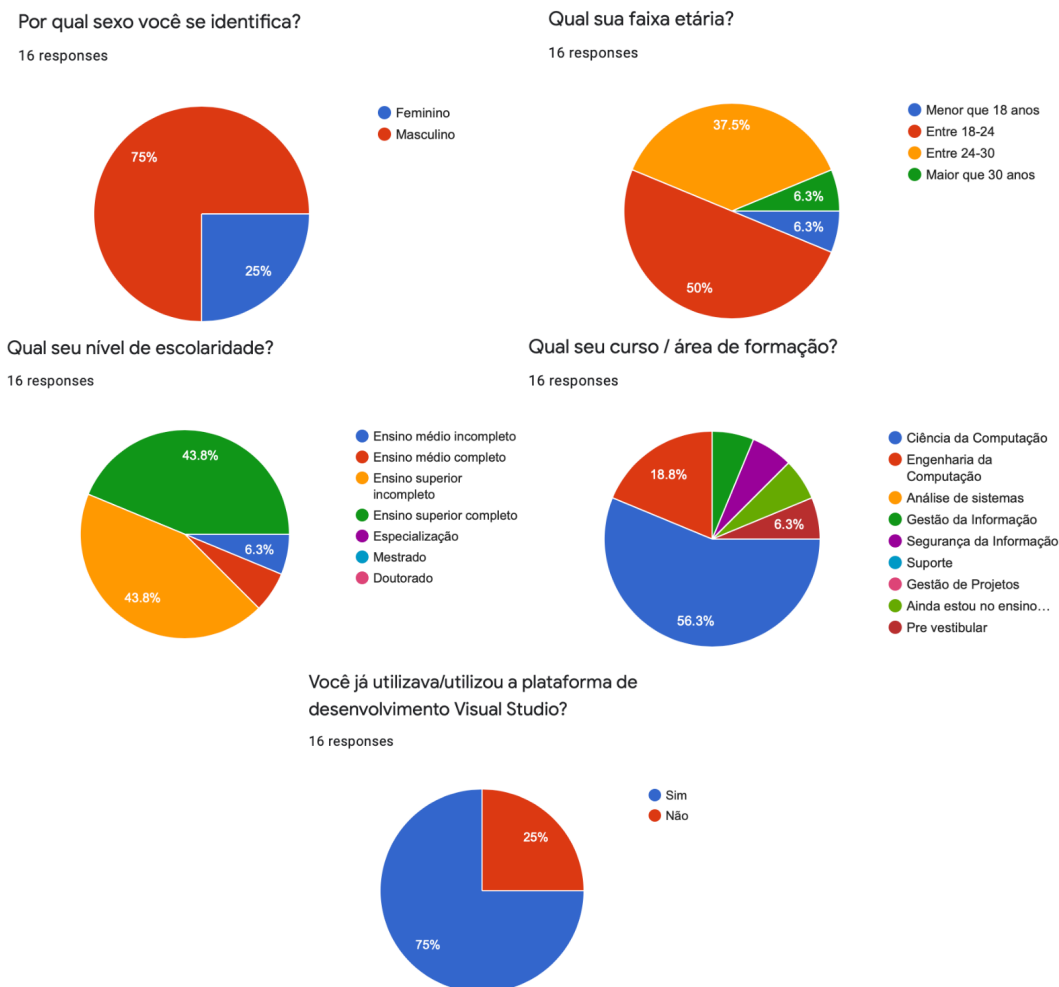
Os usuários receberam instruções para obter a extensão, o link para o questionário que está no apêndice A e um manual de instruções que deve ser utilizado caso não consiga entender o uso da ferramenta de forma intuitiva com o que a interface oferece no ambiente de desenvolvimento. Sendo assim, a página do *Templater* na loja não oferece qualquer tipo de manual de uso do produto para que seja possível avaliar também este fator na análise do produto oferecido para os usuários, de forma similar ao que é feito com aplicações comerciais, em que o usuário é apresentado com a interface do produto e pode optar por seguir um guia ou manual de uso caso tenha dificuldades. A forma de apresentação do formulário é um questionário criado através do **Google Formulários** e foi respondido remotamente em período livre conforme a disponibilidade dos voluntários do teste.

A tarefa designada para os usuários era um uso completo da aplicação como descrito anteriormente: realizar uma busca de código, selecionar uma resposta adequada, armazenar um *snippet* com base na resposta selecionada, realizar as edições necessárias no mesmo e então aplicar este trecho em qualquer editor para validar o resultado salvo.

5.3 Participantes do experimento

Os participantes deste experimento foram majoritariamente homens da faixa etária de dezoito a trinta anos que cursaram ou estão cursando o ensino superior neste momento, todos na grande área da informática, em sua maioria do curso de Ciência da Computação. Dados percentuais sobre os participantes são apresentados na Figura 5.1 e em sua grande maioria estavam familiarizados com a plataforma.

Figura 5.1: Gráficos de caracterização dos participantes do experimento



Fonte: Autor

5.4 Resultados obtidos das questões objetivas

O método de cálculo de resultados é conforme apresentado pelo sistema de avaliação selecionado (BARBOZA, 2019), em que os valores das respostas positivas dos usuários tem seu valor subtraído de um e os valores das respostas negativas são subtraídos do valor cinco, que seria a pontuação máxima. O resultado dessa operação é uma lista dos doze valores de pontuação (um para cada questão) que são então somados e multiplicados por 2,08 para que o resultado seja normalizado para um número percentual que nos dá um resultado individual da usabilidade percebida pelos participantes da avaliação. Escores menores que 60 são considerados ruins, uma usabilidade baixa e não seria possível utilizar a ferramenta no seu cotidiano. Escores superiores a 80 são considerados positivos, uso viável da ferramenta para auxiliar o trabalho dos usuários. Resultados entre estas faixas

são considerados medianos, a extensão não chama muita atenção mas é utilizável de forma geral e não se espera que integre de forma consistente as atividades diárias do usuário ou não atendeu completamente suas necessidades.

A seguir são apresentadas as pontuações percentuais ordenadas de forma crescente por resultado calculado de cada usuário que participou do teste, conforme dados na Tabela 5.1.

Tabela 5.1: Pontuação percentual por participante do experimento

Número do usuário	Pontuação percentual
1	12%
2	46%
3	49%
4	58%
5	60%
6	64%
7	68%
8	72%
9	83%
10	87%
11	89%
12	89%
13	91%
14	92%
15	92%
16	99%

Sendo assim, pode-se assumir, eliminando-se o maior valor e o menor valor obtidos no experimento para eliminar arbitrariamente os pontos discrepantes da curva e então fazendo a média dos resultados restantes obtém-se uma média final de 74%, o que significa que de forma geral essa extensão pode ser incorporada parcialmente no trabalho diário do público alvo deste produto. Analisando-se a tabela é possível também perceber que para metade dos usuários avaliadores do experimento a pontuação foi superior ao limiar de usabilidade que se definiu, dessa forma, para uma parcela significativa dos avaliadores, esta aplicação é utilizável e até desejável para uso contínuo. Apenas um quarto deste grupo de voluntários considerou a extensão não utilizável e mostrou-se insatisfeito com o experimento que realizou. Seguem os gráficos que representam os resultados das perguntas negativas no Gráfico 5.2 e positivas do experimento no Gráfico 5.3, que apresentaram de forma compilada as respostas de cada usuário ao experimento apresentado.

Seguem as questões relacionadas ao Gráfico 5.2.

Questão 03: Achei a extensão complexa e desorganizada.

Questão 05: Eu acho que precisaria de ajuda de manual de uso para usar a extensão.

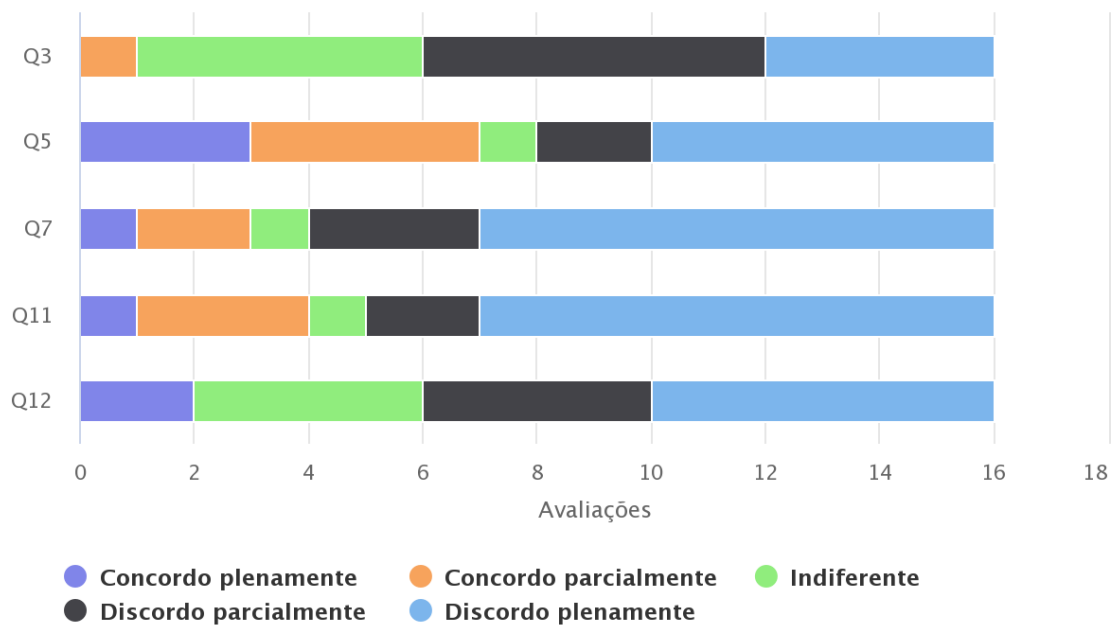
Questão 07: Eu precisei aprender várias coisas novas antes de conseguir usar a extensão.

Questão 11: Perdi mais tempo tentando entender a extensão do que de fato a utilizando.

Questão 12: Perdi mais tempo tentando entender a extensão do que levaria se pesquisasse o que buscava manualmente.

Figura 5.2: Gráficos de resultados votados nas perguntas negativas

Questões Negativas



Highcharts.com

Fonte: Autor

A seguir estão citadas as questões referentes ao Gráfico 5.3.

Questão 01: Eu acho que ter uma extensão que facilite a busca por códigos similares é útil durante o desenvolvimento.

Questão 02: Eu acho que gostaria de usar essa extensão com frequência.

Questão 04: Achei a extensão fácil e intuitiva.

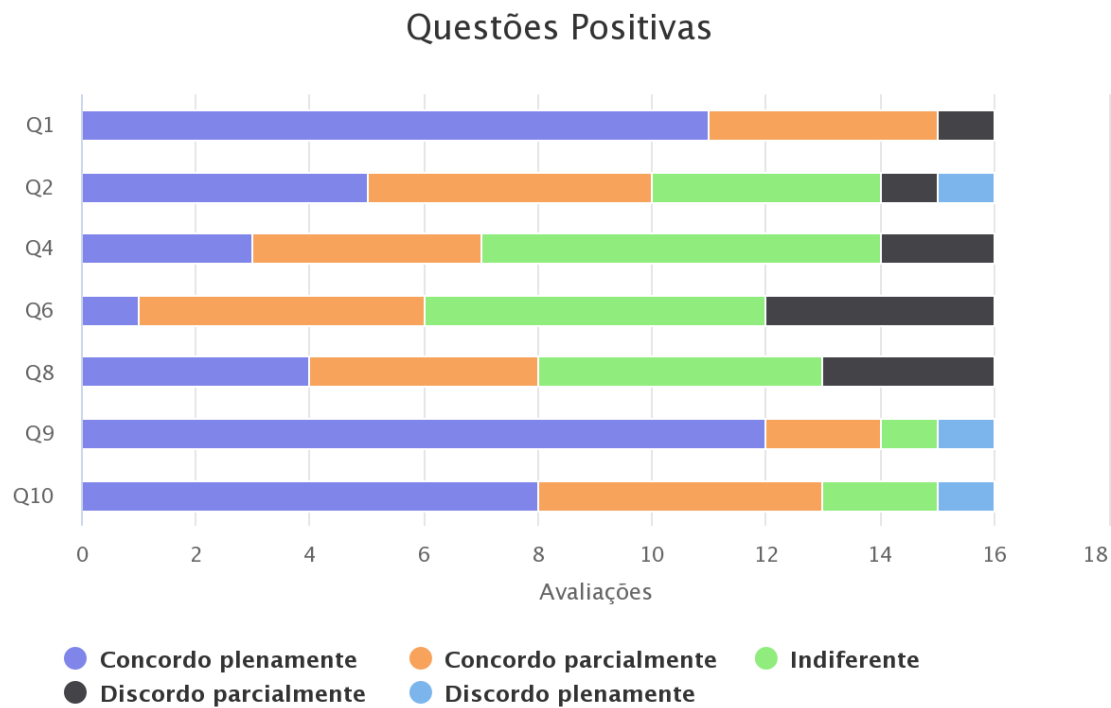
Questão 06: Eu economizei tempo de desenvolvimento utilizando a extensão.

Questão 08: Achei o sistema de buscas fácil de manusear.

Questão 09: Eu entendi como funcionava salvar um exemplo de código.

Questão 10: Encontrei o exemplo que eu precisava.

Figura 5.3: Gráficos de resultados votados nas perguntas positivas



Fonte: Autor

A seguir serão avaliados os resultados de algumas das respostas dissertativas que apontaram os pontos de destaque, tanto positivos quanto negativos, para os avaliadores do produto.

Os voluntários deste teste foram questionados com área de texto livre "*Quais as vantagens que encontraste utilizando a extensão Templater?*", e os resultados obtidos são apresentados na Figura 5.4.

De forma geral os usuários responderam positivamente sobre os aspectos que eram alvo do trabalho desenvolvido: não causar uma troca de foco do usuário saindo do seu ambiente de desenvolvimento e a percepção da facilidade para salvar os *Snippets*, com o auxílio da criação automática de arquivo e formatação na estrutura esperada pelo ambiente. Os usuários tiveram uma boa reação ao uso do fórum escolhido e percepção da possibilidade de reutilização dos trechos de código. Pelo menos um dos usuários que respondeu as perguntas dissertativas percebeu a possibilidade de utilização da ferramenta de armazenamento em qualquer arquivo.

Considerando as respostas destes usuários, afirma-se que as funcionalidades e objetivos básicos da extensão foram atingidos com sucesso. O mínimo produto viável foi concluído e entregue com sucesso, dado este *feedback*.

Figura 5.4: Respostas sobre as vantagens do uso do *Templater*

Quais as vantagens que encontraste utilizando a extensão *Templater*?

5 responses

Foi mais fácil não ter que deixar o editor e criar outra aba para fazer a busca. Além disso com a ferramenta para salvar snippets não vou precisar repetir a busca. A criação de um snippet não é algo que eu faria normalmente e em outro momento teria que encontrar novamente o exemplo ou buscar no meu código onde utilizei.

Busca direto no Stack Overflow

Possibilidade de salvar snippets de código através de um buffer editável. A extensão permite salvar snippets diretamente a partir de qualquer arquivo, de uma forma mais simples do que a fornecida nativamente pelo VS Code.

Salvar código e usar o código é realmente simples e fácil de usar. O fato da geração de um arquivo com os templates também é bem bacana, pois facilita o compartilhamento de snippets entre usuários caso necessário.

As pesquisas no StackOverflow estarem integradas com a IDE é algo bem interessante e que pode poupar tempo do desenvolvedor.

Mais vantajoso buscar o que se quer na própria plataforma de desenvolvimento, sem ter que tirar o foco pra entrar em um site de busca.

Fonte: Autor

O segundo questionamento com área de texto livre "*Quais as desvantagens que encontraste utilizando a extensão *Templater*?*", na Figura 5.5, é a contrapartida negativa do cenário anterior e abre espaço para que os usuários expressem os aspectos mais fracos da extensão.

As maiores desvantagens apontadas pelos usuários dizem respeito a opções que trariam mais conforto e facilidade de interação com a ferramenta. Um ponto recorrente nas respostas é a limitação da pesquisa. Conforme apontado em 3.1 e 3.2.2 a API oferecida pelo *Stack Overflow* é limitada e tem menos poder que um provedor de buscas (como o *Google*) e acaba ficando presa a buscas sequenciais na versão utilizada para este projeto. Entende-se que é uma limitação e pode levar o usuário a não encontrar um resultado devido à busca não ser exata com os termos do título da pergunta. A ferramenta também não utiliza o corpo das perguntas nas buscas.

Foi apontada a falta de um comando de edição dos trechos armazenados, um ponto de melhoria que se detectou durante o desenvolvimento e foi registrado como ponto de melhoria futura.

Além disso houve dificuldades relacionadas à falta de atributos mais elaborados de interface, como o retorno à etapa anterior durante uma busca e a apresentação de uma pré-visualização ou miniatura dos resultados, bem como a concatenação dos resultados

na apresentação dos dados quando múltiplas pesquisas são realizadas. Todos estes pontos foram considerados potenciais etapas da melhoria de interface.

Figura 5.5: Respostas sobre as desvantagens do uso do *Templater*

E quais as desvantagens que encontraste utilizando a extensão Templater?

5 responses

Não tem uma forma fácil de abrir o arquivo de snippets para fazer alterações em um snippet que eu já salvei, como não sabia onde encontrar eu criei outro snippet vazio e então removi ele para poder editar o snippet que eu queria mudar.

Não há preview do que se está buscando

A busca no StackOverflow não é muito precisa, e os resultados nem sempre batem com o resultado da busca diretamente no site.

Pesquisa no StackOverflow é bem bacana, mas a usabilidade poderia ser melhor. Em vez de sempre abrir as respostas no mesmo arquivo, cada pergunta/resposta poderia ser aberto em um arquivo diferente, ficando mais organizado. Também falta a opção (ou ao menos não encontrei) de voltar atrás e corrigir o termo pesquisado, é necessário fechar a pesquisa por completo e reiniciar o processo. Mas são pequenos detalhes apenas.

Os exemplos infelizmente são limitados

Fonte: Autor

Para o último questionamento com área de texto livre "*Há algo que você gostaria de mudar? O quê?*", apresentados na Figura 5.6, abre-se espaço para sugestões de mudanças que os testadores entendem que melhorariam sua experiência com o produto.

Figura 5.6: Respostas sobre as mudanças propostas para o *Templater*

Há algo que você gostaria de mudar? O quê?

5 responses

Adicionar um botão de edição de snippet.

Não usaria como templater, mas apenas como buscador. Talvez seria melhor se tivesse alguma forma de apenas visualizar a busca.

Senti falta da possibilidade de parametrizar snippets, de forma a permitir que sejam criados snippets genéricos a partir de trechos de código específicos (como geralmente é o caso no StackOverflow).

Um manual seria útil, assim como possibilidade de opções ao clicar com o botão direito do mouse. O manual poderia ser no próprio README da extensão. O link para a página original do StackOverflow também seria útil caso o usuário desejasse entrar na página e interagir comentando algo, questionando alguma coisa ou então vendo outras respostas.

A interface

Fonte: Autor

Deste questionamento reforçaram os pontos já detectados na pergunta sobre os pontos negativos. A principal melhoria apontada é relacionada à visualização dos dados. A falta de interações mais simples e diretas após a conclusão do uso da ferramenta, para editar os *snippets*. Foi citada novamente a questão de visualização de uma miniatura da resposta antes da seleção. Um dos usuários apontou que teve de fato necessidade de utilizar o manual da ferramenta. Estes pontos estão de acordo com o que foi detectado nas questões objetivas no que diz respeito aos resultados das questões direcionadas a usabilidade do programa.

6 CONCLUSÃO

Este trabalho apresentou a extensão *Templater* desenvolvida para uso no editor de texto *VS Code* com o objetivo de viabilizar a pesquisa de perguntas relacionadas à área da computação no fórum *Stack Overflow* e também oferecer uma forma facilitada de armazenamento de *snippets*, trechos de código, obtidos das respostas pesquisadas ou mesmo qualquer trecho de código selecionado pelo usuário no editor. Tanto o ambiente de edição de código quanto o fórum escolhido têm reconhecimento entre os profissionais da área e são ferramentas relevantes utilizadas com frequência por desenvolvedores de *software* de forma geral. A intenção destas escolhas foi utilizar ambientes reais de produção para que o *Templater* possa de fato ser integrada ao fluxo de trabalho de seus usuários.

Os resultados indicaram que o *Templater* atinge os objetivos de forma satisfatória, os usuários conseguem completar as tarefas propostas, ou seja, utilizar todo o fluxo da ferramenta sem problemas e percebem benefícios da adoção do *Templater* de forma geral. Os pontos negativos foram avaliados para melhorias futuras visando a consolidação da ferramenta como um produto sólido para o auxílio na atividade de desenvolvimento de código. A grande maioria dos usuários avaliou a ferramenta positivamente, o que aponta a possibilidade do seu uso e a demanda por melhorias é entendida como uma colaboração importante para a viabilização da distribuição e divulgação do *Templater*.

O projeto foi considerado concluído em sua versão inicial e foi publicado na loja de extensões do editor *VS Code* de forma pública e gratuita. Para as próximas etapas de implementação de melhorias esta ferramenta continuará disponível neste formato, sob uma licença de uso permissiva e o código será liberado de forma irrestrita para customização e contribuições. A expectativa desta ação é incentivar a melhoria colaborativa e contínua dos meios de trabalho dos profissionais que possam se beneficiar deste tipo de ferramenta.

6.1 Dificuldades encontradas

A principal dificuldade foi citada na Subseção 3.2.1. Em sua primeira abordagem o projeto visava oferecer acesso à página completa do fórum escolhido para que fosse possível ao usuário adicionar outras opções de fóruns às fontes de pesquisa simplesmente acessando as páginas dos mesmos. Esta implementação se mostrou inviável devido às limitações da plataforma *VS Code* no que diz respeito à manipulação de dados em páginas

exibidas na sua interface. Apesar de oferecer suporte para a visualização de *frames* com o conteúdo da página a ferramenta não suporta nenhum tipo de edição dos dados, nem mesmo a seleção de texto para a cópia. De acordo com a documentação do editor, não foi vista necessidade de implementação destas capacidades para suprir as necessidades atuais da comunidade.

É válido mencionar também que o uso da linguagem *TypeScript* torna o desenvolvimento muito menos flexível do que seria com a linguagem para qual é transpilada, *JavaScript*, que é muito mais permissiva quanto aos tipos de dados e métodos utilizados no programa. Esta barreira foi considerada aceitável para o projeto tendo em vista que traz uma série de garantias de consistência dos dados e também torna o código mais legível e bem estruturado devido às regras que impõe ao código desenvolvido.

6.2 Melhorias para o *Templater*

Com base na análise do produto criado e na avaliação dos usuários so *Templater* seguintes características foram identificadas:

1. **Centralizador:** Não tira o foco do usuário, mantendo-o dentro do ambiente.
2. **Eficiente:** Interface para salvar trechos de código rapidamente com um único comando.
3. **Fácil de usar:** Após aprendizado das funcionalidades é muito simples de aplicar suas funções.
4. **Pesquisa limitada:** A busca está restrita pelas limitações da *API* fornecida pelo fórum
5. **Portabilidade razoável:** É possível salvar o arquivo de *snippets* e mover para outro computador, porém não é oferecido comando para fazer isto.
6. **Interface pobre:** Não apresenta miniaturas dos resultados, falta um comando para edição dos trechos salvos.

Com esta enumeração de pontos descritivos do produto *Templater* é possível identificar facilmente propostas de melhorias futuras. Para o ponto de número 3 pretende-se apresentar na página da loja onde o projeto foi disponibilizado um tutorial com imagens detalhando e exemplificando o uso da ferramenta, bem como um comando para mostrar novamente o manual em caso de dúvida será adicionado às funcionalidades da ferramenta. O ponto de número 4 requer maior trabalho e será resolvido aplicando através de código

na extensão a combinação dos termos de pesquisa em diferentes ordens e subconjuntos dos termos para atingir uma quantidade maior de resultados para a pesquisa e será avaliada a aplicação de sinônimos para alguns dos termos utilizados, ou seja, a pesquisa do usuário resultará em uma série de pesquisas similares buscando chegar o mais próximo possível do resultado desejado e ao mesmo tempo oferecer mais opções de resultados.

O quinto ponto da lista requer a atribuição de uma nova contribuição na extensão oferecendo exportação e importação de arquivos de *snippet* para que o usuário possa migrar entre suas estações de trabalho.

As considerações dos usuários sobre apresentação de miniaturas ou resumos das respostas para as opções retornadas das buscas, comando para edição de *snippets* e opções de correção dos termos sem cancelamento da pesquisa serão incluídas na etapa de refinamento da interface.

Aplicadas estas correções e melhorias o produto será considerado concluído em sua segunda versão para a realização de uma nova rodada de avaliações com usuário.

6.3 Trabalhos futuros

Existe um ponto muito pouco explorado no desenvolvimento da aplicação *Templater* que tem espaço para inclusão na mesma. Se trata da edição colaborativa, que nesse trabalho está contemplada na disponibilização do código fonte da ferramenta e distribuição do produto de forma gratuita.

A proposta para o continuidade de trabalhos neste tipo de ferramenta é a criação de uma base compartilhada distribuída de *snippets*, com a visão de que o futuro da criação de código é de cooperação e compartilhamento entre os desenvolvedores.

Enquanto a implementação de métodos de importação e exportação da base de trechos de código criada pelos usuários do *Templater* está listada nas melhorias, um projeto mais abrangente que armazene as bases de conhecimento e as torne compartilháveis e colaborativas é o tipo de trabalho que direciona nossa área para o comportamento de colaboração idealizado. Este tipo de implementação abre espaço para que o *Templater* seja usado como base de conhecimentos para padronização de código para programadores trabalhando em projetos conjuntos e também como um guia de referência para novos desenvolvedores buscando iniciar sua carreira em uma área cuja base compartilhada já existe de forma aberta. Dessa forma damos mais passos em direção a um ambiente mais aberto que se desenvolve com mais qualidade através da colaboração do grupo.

REFERÊNCIAS

BARBOZA, A. **Medindo a usabilidade do seu produto com System Usability Scale (SUS)**. 2019. <<https://medium.com/design-contaazul/medindo-a-usabilidade-do-seu-produto-com-system-usability-scale-sus-3956612d9229>>. Acessado em 04 de dezembro de 2019.

CREVECOEUR, G. **GitHub - gcrev93/code-stackoverflow: Search StackOverflow from Visual Studio Code**. 2019. <https://github.com/gcrev93/code-stackoverflow>. Acessado em 22 de dezembro de 2019.

H, N. **How do I compare strings in Java? - Stack Overflow**. 2009. <<https://stackoverflow.com/questions/513832/how-do-i-compare-strings-in-java>>. Acessado em 08 de novembro de 2019.

HOME | DefinitelyTyped. 2019. <<http://definitelytyped.org/>>. Acessado em 17 de novembro de 2019.

LIKERT, R. **A Technique for the Measurement of Attitudes, Edições 136-165**. 1932. <https://medium.com/design-contaazul/medindo-a-usabilidade-do-seu-produto-com-system-usability-scale-sus-3956612d9229>. Acessado em 04 de dezembro de 2019.

MASSACHUSETTS INSTITUTE OF TECHNOLOGY. **The MIT License | Open Source Initiative**. 1988. <<https://opensource.org/licenses/MIT>>. Acessado em 08 de novembro de 2019.

MDN WEB DOCS. **<iframe>: The Inline Frame element - HTML: Hypertext Markup Language | MDN**. 2019. <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/iframe>. Acessado em 22 de dezembro de 2019.

MICROSOFT INC. **TypeScript - JavaScript that scales**. 2012. <<https://www.typescriptlang.org/>>. Acessado em 09 de novembro de 2019.

MICROSOFT INC. **Microsoft**. 2019. <<https://www.microsoft.com/pt-br/>>. Acessado em 08 de novembro de 2019.

MICROSOFT INC. **Snippets in Visual Studio Code**. 2019. <<https://code.visualstudio.com/docs/editor/userdefinedsnippets>>. Acessado em 16 de novembro de 2019.

MICROSOFT INC. **Visual Studio Code - Code Editing. Redefined**. 2019. <<https://code.visualstudio.com/>>. Acessado em 08 de novembro de 2019.

MICROSOFT INC. **Webview API | Visual Studio Code**. 2019. <<https://code.visualstudio.com/api/extension-guides/webview>>. Acessado em 16 de novembro de 2019.

MICROSOFT INC. **Your First Extension | Visual Studio Code**. 2019. <<https://code.visualstudio.com/api/get-started/your-first-extension>>. Acessado em 16 de novembro de 2019.

MICROSOFT INC. **Your First Extension | Visual Studio Code**. 2019. <<https://code.visualstudio.com/api/get-started/your-first-extension#using-javascript>>. Acessado em 17 de novembro de 2019.

MISHRA, O. A. **Stack Overflow = Programmers best friend**. 2019. <<https://codeburst.io/stack-overflow-programmers-best-friend-ef77e4d142c4>>. Acessado em 09 de novembro de 2019.

NORRIS, E. **striptags - npm**. 2019. <<https://www.npmjs.com/package/striptags>>. Acessado em 16 de novembro de 2019.

PAPA, J. **GitHub - johnpapa/vscode-angular-snippets: Angular Snippets for VS Code**. 2019. <https://github.com/johnpapa/vscode-angular-snippets>. Acessado em 22 de dezembro de 2019.

SAMBORSKI, C. **appdata-path - npm**. 2019. <<https://www.npmjs.com/package/appdata-path>>. Acessado em 16 de novembro de 2019.

SENGSTACKE, P. **JavaScript Transpilers: What They Are Why We Need Them — Scotch.io**. 2019. <https://scotch.io/tutorials/javascript-transpilers-what-they-are-why-we-need-them>. Acessado em 22 de dezembro de 2019.

STACK EXCHANGE. **Stack Exchange**. 2009. <<https://stackexchange.com/tour>>. Acessado em 09 de novembro de 2019.

SWAAGMAN, M. **stackexchange - npm**. 2019. <<https://www.npmjs.com/package/stackexchange>>. Acessado em 16 de novembro de 2019.

APÊNDICE A — QUESTIONÁRIO APLICADO NOS TESTES COM USUÁRIOS

Neste apêndice encontra-se o formulário feito com os usuários, em formato de texto e imagens.

Pesquisa de satisfação sobre Templater

Este formulário tem como objetivo analisar as experiências dos usuários a respeito da extensão Templater implementada para um trabalho de conclusão do curso de Ciência da Computação e já disponibilizada na loja do Visual Studio, plataforma para a qual ele foi desenvolvido.

O formulário será analisado de forma anônima e tem duração média de 10 minutos.

Perguntas iniciais

Por qual sexo você se identifica?

- Feminino
- Masculino

Qual sua faixa etária?

- Menor que 18 anos
- Entre 18-24
- Entre 24-30
- Maior que 30 anos

Qual seu nível de escolaridade?

- Ensino médio incompleto
- Ensino médio completo
- Ensino superior incompleto
- Ensino superior completo
- Especialização
- Mestrado
- Doutorado

10. Encontrei o exemplo que eu precisava. *

	1	2	3	4	5	
Discordo plenamente	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Concordo plenamente

11. Perdi mais tempo tentando entender a extensão do que de fato a utilizando. *

	1	2	3	4	5	
Discordo plenamente	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Concordo plenamente

12. Perdi mais tempo tentando entender a extensão do que levaria se pesquisasse o que buscava manualmente. *

	1	2	3	4	5	
Discordo plenamente	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Concordo plenamente

Sua opinião

Sua opinião é muito importante para que a extensão Templater sempre alcance seu objetivo de facilitar o desenvolvedor durante uma implementação.

Quais as vantagens que encontraste utilizando a extensão Templater?

E quais as desvantagens que encontraste utilizando a extensão Templater?

Há algo que você gostaria de mudar? O quê?