



Universidade Federal de Pernambuco

Curso de Ciência da Computação

**WRITEME 2.0: Generalização e validação de uma ferramenta
de auxílio à escrita de READMEs baseada em dados abertos**

Trabalho de Conclusão de Curso de Graduação

por

Victor Miranda de Melo

Orientador: Prof. Leopoldo Motta Teixeira

Recife, Maio / 2023

Victor Miranda de Melo

**WRITEME 2.0: Generalização e validação de uma ferramenta de auxílio à
escrita de READMEs baseada em dados abertos**

Monografia apresentada ao Curso de Ciência da Computação como requisito parcial para a obtenção do Título de Bacharel em Ciência da Computação, Centro de Informática da Universidade Federal de Pernambuco.

Orientador: Prof. Leopoldo Motta Teixeira

Recife

2023

Ficha de identificação da obra elaborada pelo autor,
através do programa de geração automática do SIB/UFPE

Melo, Victor Miranda de.

WRITEME 2.0: generalização e validação de uma ferramenta de auxílio à escrita de READMEs baseada em dados abertos / Victor Miranda de Melo. - Recife, 2023.

32 : il., tab.

Orientador(a): Leopoldo Motta Teixeira

Trabalho de Conclusão de Curso (Graduação) - Universidade Federal de Pernambuco, Centro de Informática, Ciências da Computação - Bacharelado, 2023.

1. Documentação de repositórios. 2. README. 3. GitHub. I. Teixeira, Leopoldo Motta. (Orientação). II. Título.

000 CDD (22.ed.)

Agradecimentos

Agradeço primeiramente à minha mãe, Carmem, por todo apoio, incentivo feito desde os meus primeiros passos até hoje. Por cada sacrifício feito para que eu não passasse nenhuma necessidade. Hoje, consigo retribuir um pouco disso com esta graduação que tanto sonhava.

Agradeço aos meus companheiros de graduação: Hilton, Renan, Bruno, Ricardo e Nicola. Por terem feito a trajetória mais leve. Comemoramos muitos sucessos de cada um, fizemos piadas dos nossos fracassos, mas nunca tiramos o sorriso do rosto. Agora, vamos para mais uma comemoração.

À Diana, pelos incentivos, por ter acreditado em mim, mostrando que eu sou capaz. Obrigado.

Ao meu orientador, professor Leopoldo, demais professores e funcionários do Centro de Informática, que me ajudaram direta ou indiretamente a chegar no final desta graduação. Obrigado.

Finalmente, agradeço aos grupos: Porteiros, Condomínio e Bolinhas. Pelos anos de amizade e tudo que aprendemos juntos.

Último Gás.

*Nós só podemos ver um pouco do futuro,
mas o suficiente para perceber que há muito a fazer.*

Alan Turing

RESUMO

Uma forma comum de documentação de software são os arquivos README (leia-me, em português) comumente disponibilizados em repositórios de código. Essa documentação facilita a utilização do software por seus usuários e também a contribuição por desenvolvedores que desejam ajudar na sua implementação.

O arquivo serve como uma página de boas vindas e, quando mal escrito, pode dificultar a utilização do software. O trabalho que idealizou o WRITEME propôs um modelo de ferramenta baseada em dados abertos, que teria caráter genérico e dinâmico para auxiliar desenvolvedores a escrever READMEs, recomendando seções para a escrita da documentação de seus projetos. Entretanto, sua implementação não foi finalizada, fazendo com que em seu estado atual esteja restrita a duas linguagens de programação.

Este projeto continua a implementação da integração entre o back-end e o front-end da ferramenta, enfim trazendo seu caráter dinâmico e genérico. Isso amplia sua utilização e facilita a realização de mais testes para validação, realizando uma nova pesquisa sobre a experiência de uso.

Palavras-chave: README, Documentação de Repositórios, GitHub

ABSTRACT

A common type of software documentation are README files, usually displayed on the main page of software repositories. Such documentation facilitates the software's utilization by its users and the contributions from developers who wish to help on its implementation.

The file works as a welcome page and, if badly written, can make it difficult to use the software. The work that idealised WRITEME suggested a tool based on open data, which would be generic and dynamic in order to help developers write READMEs by recommending sections to their project's documentation. Its implementation, however, wasn't finished, since its current state is restricted to only two programming languages.

This project aims to resume the implementation of the tool's back-end and front-end integration, at last bringing its dynamic and generic aspect. This will expand the tool's utilization and favor the realization of more validation tests, creating a new research on user experience.

Keywords: README, Repository Documentation, GitHub

LISTA DE FIGURAS

Figura 1	Exemplo de um arquivo README.	12
Figura 2	Ilustração da ferramenta em uso. À esquerda, seções recomendadas pelo WRITEME e à direita, seções selecionadas pelo usuário	16
Figura 3	Fluxo de utilização	17
Figura 4	Arquitetura final com serviços da AWS	18
Figura 5	Respostas à pergunta: “Quanto tempo de experiência profissional com desenvolvimento de software você tem?”	24
Figura 6	Respostas à pergunta: “Quão importante você considera que é ter um bom README para o sucesso de um projeto?”	24
Figura 7	Respostas à pergunta: “Quais fontes você utilizou na época para saber que tipo de conteúdo deveria constar no README antigo?”	25
Figura 8	Respostas à pergunta: “Você considera o README como um bom README?” entre a versão anterior e final	26
Figura 9	Respostas à pergunta: “Neste novo README, de onde vieram as seções utilizadas?”	26
Figura 10	Respostas à pergunta: “Em comparação com o README anterior, essa nova versão é:”	27
Figura 11	Comando de entrada utilizado no ChatGPT	28
Figura 12	Respostas à pergunta: “Em comparação com o README anterior, essa nova versão é:”	29

LISTA DE TABELAS

Tabela 1	Média de dias entre alterações nos arquivos README.	21
Tabela 2	Notas médias da importância dos arquivos README agrupadas por tempo de experiência dos participantes.....	23
Tabela 3	Respostas à pergunta: “Qual a linguagem de programação do projeto escolhido?”	25
Tabela 4	Comparação entre as quantidades de seções e subseções dos arquivos antes e após o uso do WRITEME	26

LISTA DE SIGLAS

API	Application Programming Interface
AWS	Amazon Web Services
CPU	Central Processing Unit
CSS	Cascading Style Sheets
HTML	HyperText Markup Language
IA	Inteligência Artificial
JSON	JavaScript Object Notation
URL	Uniform Resource Locators

SUMÁRIO

1	INTRODUÇÃO	10
2	CONTEXTO	12
2.1	README	12
2.2	A FERRAMENTA WRITEME.....	13
2.3	LIMITAÇÕES DO WRITEME	16
3	SOLUÇÃO PROPOSTA	17
3.1	FLUXO DE UTILIZAÇÃO	17
3.2	ARQUITETURA SERVERLESS	17
3.2.1	REACTJS	18
3.2.2	LAMBDA.....	19
3.2.3	DYNAMODB	19
3.2.4	BATCH	19
3.2.5	DOCKER.....	20
3.2.6	S3	20
3.3	ATUALIZAÇÕES DAS SEÇÕES RECOMENDADAS	20
4	METODOLOGIA	22
4.1	QUESTIONÁRIO.....	22
5	ANÁLISE DE RESULTADOS	23
5.1	ESTUDO COMPLEMENTAR.....	28
6	CONCLUSÃO	30
7	TRABALHOS FUTUROS	31

1 INTRODUÇÃO

A documentação de software pode ser definida como um conjunto de manuais gerais e técnicos, podendo ser organizado em forma de textos, diagramas, fluxogramas, dentre outros [1]. Esse material pode ser utilizado por profissionais envolvidos no projeto, independente da área de atuação, servindo para apoiá-los em todas as etapas de desenvolvimento [2].

Em repositórios de código aberto é comum encontrarmos esse tipo de documentação escrita em arquivos README. Nele, espera-se que os responsáveis comuniquem para os interessados algumas informações, como: o que o projeto faz, para que é útil, como utilizar, onde obter ajuda e quem contribui para o projeto [3].

A importância de um arquivo README deve ser compreendida. Ele é a primeira coisa que os interessados veem ao chegar ao seu projeto no Github¹ e uma importante fonte de informações sobre o projeto. Portanto, um arquivo README bem apresentado pode ajudar a construir confiança no seu projeto e atrair mais colaboradores e usuários [4]. A primeira impressão passada pela apresentação pode ser transmitida para o restante do projeto por associação [5].

Visando auxiliar desenvolvedores na escrita de READMEs, foi criada a ferramenta WRITEME [6]. Em sua versão inicial, o WRITEME recomenda seções frequentemente encontradas em repositórios famosos de acordo com a linguagem de programação utilizada. Até então, essa era uma ferramenta estática, que funcionava apenas com duas linguagens processadas previamente: Swift e Python. No capítulo 2, detalhamos como funciona a ferramenta.

Este trabalho tem como objetivo tornar o WRITEME mais abrangente, funcionando independente da linguagem de programação escolhida pelo usuário. Além disso, faremos com que a ferramenta funcione de forma autônoma, para qualquer linguagem de programação listada no GitHub.

Neste documento apresentamos, no capítulo 2, o contexto no qual o trabalho se encontra, o que são arquivos README, detalhamos como funcionava o WRITEME e o que precisava ser melhorado na ferramenta. No capítulo 3, apresentamos a solução proposta, detalhando a nova arquitetura e como os módulos e serviços se relacionam. No

¹<https://github.com>

capítulo 4, detalhamos o estudo de caso e resultados obtidos. Por fim, no capítulo 6, analisamos os resultados obtidos no estudo de caso e possíveis trabalhos futuros.

2 CONTEXTO

Este capítulo apresenta conceitos sobre o que é um arquivo README, a ferramenta WRITEME e problemas apresentados pelo próprio criador da ferramenta [6].

2.1 README

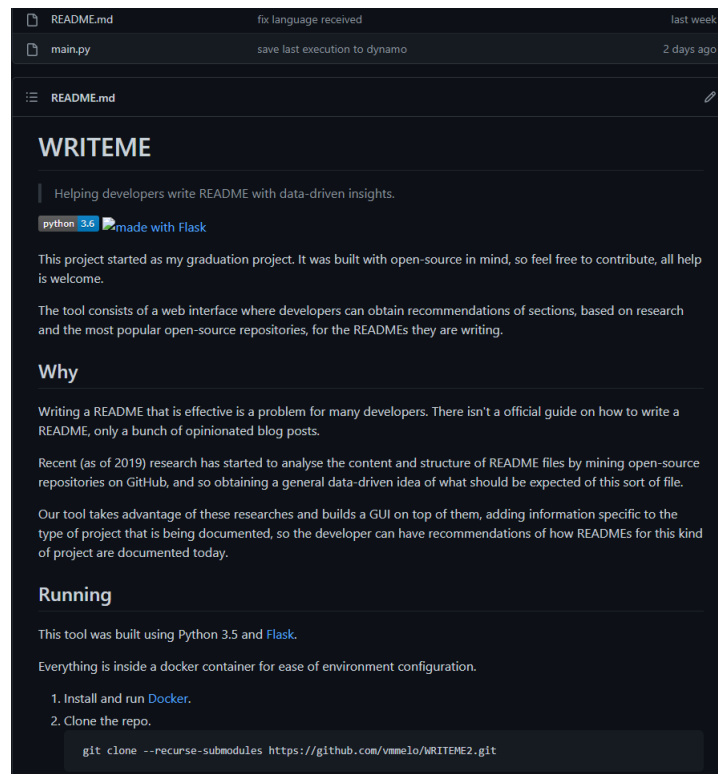


Figura 1: Exemplo de um arquivo README.
Fonte: <https://github.com/hpbl/WRITEME>.

README é um termo escrito na língua inglesa que significa “Leia-me”, um pedido para que os usuários leiam o arquivo de documentação antes de usar o projeto. Em plataformas de hospedagem de código-fonte e arquivos com controle de versão como o Github, o conteúdo do README é exibido em destaque na página inicial do projeto, sendo uma espécie de porta de entrada para o mesmo.

Pode-se observar na Figura 1 que o arquivo README é capaz de fornecer informações importantes sobre projetos de software, biblioteca, pacotes ou aplicativo. É um dos primeiros arquivos que alguém costuma encontrar ao baixar ou clonar um projeto, e é uma fonte de informações comum para aqueles que procuram entender o que o

projeto faz e como usá-lo.

Um arquivo README normalmente incluirá informações sobre o objetivo do projeto, porquê o projeto é útil, detalhes sobre contribuidores, como contribuir para o projeto e quem mantém o projeto². Além disso, pode conter informações sobre os requisitos do sistema, links para documentação adicional, exemplos de uso, entre outros. Esse conteúdo pode variar de acordo com as necessidades do projeto e da linguagem de programação utilizada.

2.2 A FERRAMENTA WRITEME

O WRITEME é uma ferramenta web projetada para recomendar seções de arquivos README, a fim de auxiliar desenvolvedores ou responsáveis por repositórios de software a documentá-los. Nela, o usuário pode escolher entre as linguagens de programação de acordo com o projeto que deseja documentar.

A partir da escolha do usuário, o back-end faz uma busca na API do Github, dos repositórios de código aberto mais famosos da plataforma, usando como critério a quantidade de estrelas recebidas por usuários. Em seguida, são obtidos os READMEs e o conteúdo deles é extraído e classificado através de scripts do classificador [7]. Então, é exportado um arquivo no formato JSON, contendo as seções classificadas e outro arquivo que contém uma árvore, representando o aninhamento das seções, onde cada nó é uma seção e os seus filhos são as sub-seções.

O bloco de código 2.1 exibe um trecho do arquivo de formato JSON. Nele, está uma lista de objetos que representam as seções. No exemplo, pode-se observar duas seções do repositório `LaravelShoppingcart.md` do usuário `Crinsane`. Cada elemento da lista contém o identificador do arquivo e da seção, nome do arquivo README, nível da seção, título e as classificações extraídas pelo classificador [7].

²<https://docs.github.com/pt/repositories/managing-your-repositorys-settings-and-features/customizing-your-repository/about-readmes>

```

1  [
2    // ...
3    {
4      "file_id": "2",
5      "section_id": "2",
6      "readme_file_name": "Crinsane.LaravelShoppingcart.md",
7      "heading_level": 2,
8      "title": "Installation",
9      "section_codes": [
10         3
11      ]
12    },
13    {
14      "file_id": "2",
15      "section_id": "3",
16      "readme_file_name": "Crinsane.LaravelShoppingcart.md",
17      "heading_level": 2,
18      "title": "Overview",
19      "section_codes": [
20         1,
21         3,
22         6
23      ]
24    },
25    // ...
26  ]

```

Código 2.1: Seções extraídas e classificadas de projetos famosos para linguagem PHP

No bloco de código 2.2 está exibido um trecho do conteúdo do arquivo JSON responsável hierarquizar as seções classificadas como uma árvore. No exemplo, pode-se observar a hierarquia entre as seções pertencentes ao repositório `LaravelShoppingcart.md` do usuário `Crinsane`. A seção `Database` (nível 2) é um nó, contendo o filho `Configuration` (nível 3), que não contém filhos. Mais abaixo, está a seção `Exceptions` (nível 2), sem filhos.

```

1 {
2   "Crinsane.LaravelShoppingcart.md": [
3     // ...
4     {
5       "level": 2,
6       "id": "database",
7       "name": "Database",
8       "children": [
9         {
10          "level": 3,
11          "id": "configuration",
12          "name": "Configuration",
13          "children": []
14        },
15        // ...
16      ]
17    },
18    {
19      "level": 2,
20      "id": "exceptions",
21      "name": "Exceptions",
22      "children": []
23    },
24    // ...
25  ]
26 }

```

Código 2.2: Árvores das seções para linguagem PHP

Os arquivos gerados pelo back-end são lidos pelo front-end, ilustrado na Figura 2, que monta e exibe ao lado esquerdo as sugestões das seções para o usuário escolher as que se encaixam com seu projeto. Ao selecionar uma seção, ela é adicionada a um bloco no lado direito, junto com todas as outras seções selecionadas e o usuário pode salvar o arquivo `README.md` gerado. A ferramenta sugere que o usuário edite o arquivo gerado como achar necessário, para documentar seu projeto.

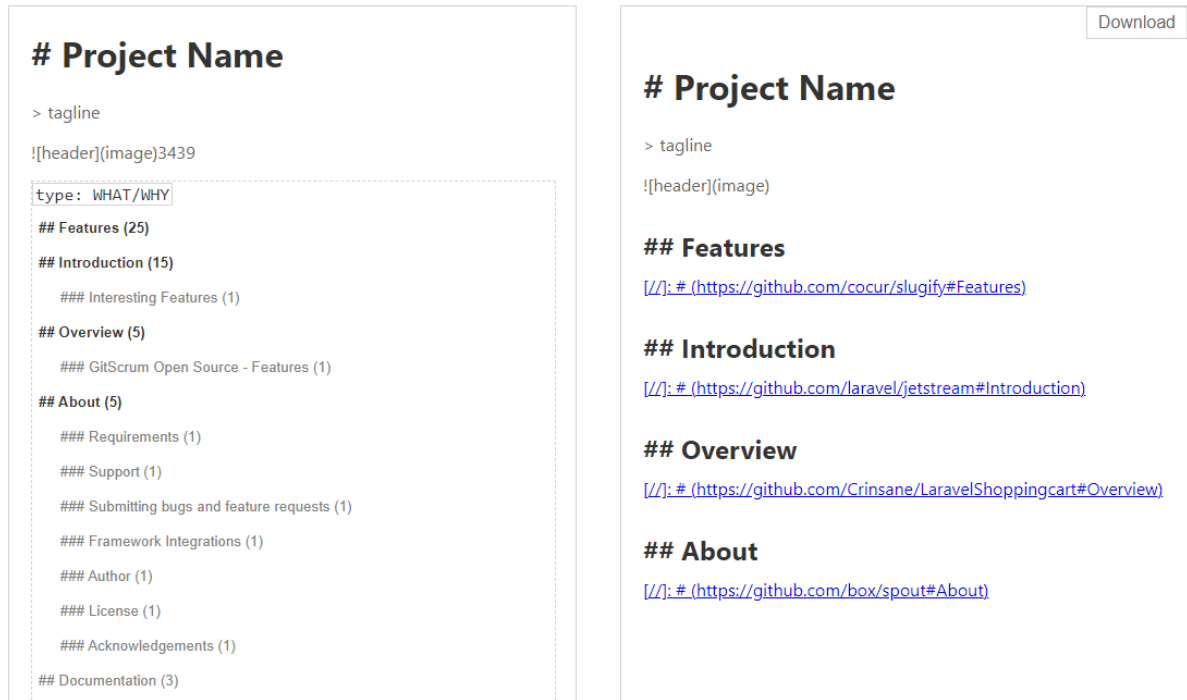


Figura 2: Ilustração da ferramenta em uso. À esquerda, seções recomendadas pelo WRITEME e à direita, seções selecionadas pelo usuário

2.3 LIMITAÇÕES DO WRITEME

A arquitetura proposta inicialmente previa que todos os módulos, desde a aquisição dos READMEs famosos até a geração dos seus resultados em arquivo, estivessem integrados de forma que o processo de recomendação das seções funcionasse para qualquer linguagem de programação suportada pelo GitHub. Porém, sua implementação não foi concluída, resultando na restrição a apenas duas linguagens de programação processadas previamente: Swift e Python. O processamento é feito de forma manual, executando cada um dos scripts, salvando os resultados para o front-end lê-los. O criador da ferramenta propõe que seja analisado o conteúdo dos arquivos escritos com o auxílio da ferramenta e sugere que novos testes sejam feitos com um número maior de usuários.

3 SOLUÇÃO PROPOSTA

Neste capítulo, explicamos o fluxo de utilização e as mudanças feitas no back-end, visando solucionar os problemas descritos na seção 2.3.

3.1 FLUXO DE UTILIZAÇÃO

Para eliminar a restrição de linguagens e automatizar o seu processamento, adicionamos uma solicitação do front-end para o back-end executar todos os scripts de extração e classificação das seções, e salvando os resultados em arquivo. A Figura 3 ilustra o fluxo de utilização da aplicação.

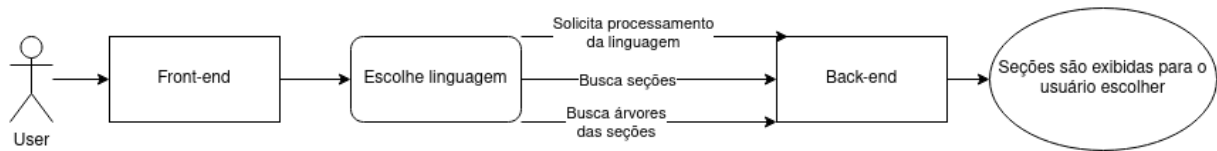


Figura 3: Fluxo de utilização

Na arquitetura proposta, o usuário acessa a interface do WRITEME e escolhe a linguagem de programação referente ao projeto que deseja escrever a documentação. A interface, por sua vez, irá se comunicar com o back-end da aplicação para obter as seções recomendadas para aquela linguagem e exibir na tela. Nesse momento, o usuário poderá escolher as seções que preferir, analisando o que se encaixa melhor com seu projeto.

Extraímos todas as linguagens de programação listadas no filtro de linguagem do Github e as inserimos em um campo de seleção no front-end da aplicação. Com isso, ela suporta qualquer linguagem que esteja listada no Github atualmente.

3.2 ARQUITETURA SERVERLESS

Buscando a redução de custos, planejamos uma arquitetura de computação serverless³ (computação sem servidor ou computação em nuvem). Esse tipo de arquitetura também tem a vantagem de dispensar a manutenção do servidor, que fica sob responsabilidade do provedor. A redução de custos é justificada pelo fato do provedor alocar os

³<https://aws.amazon.com/pt/serverless>

recursos das máquinas sob demanda, permitindo que o cliente seja cobrado apenas pelo que foi utilizado de fato. A Figura 4 mostra a arquitetura do back-end da aplicação separada em serviços da AWS. As seções a seguir estão divididas em cada serviço apresentado na imagem e contêm uma breve descrição de como funciona e como é utilizado dentro da arquitetura final da ferramenta.

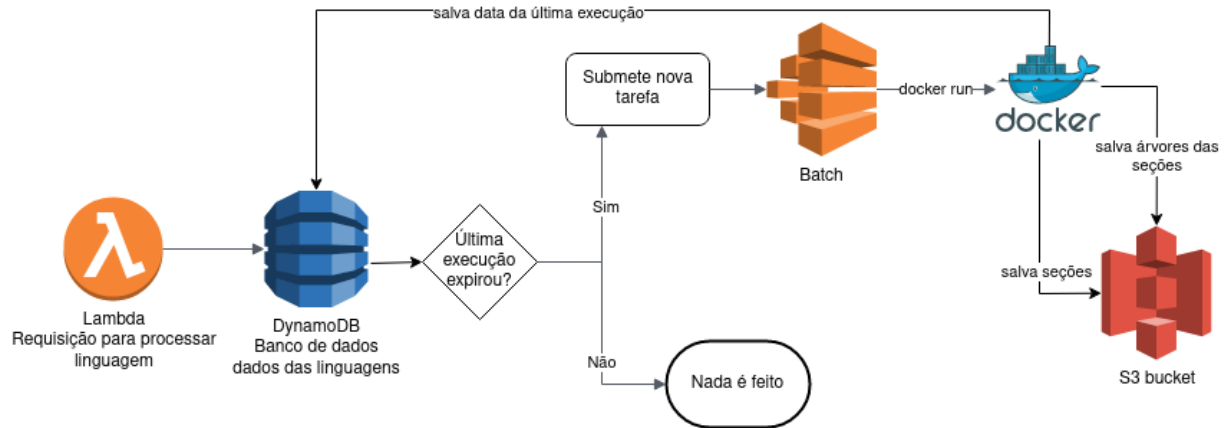


Figura 4: Arquitetura final com serviços da AWS

3.2.1 REACTJS

Mantivemos a estrutura inicial do front-end do WRITEME, que foi feito com o framework React,⁴ usando HTML, CSS e JavaScript. Porém, foram feitas alterações no fluxo percorrido:

- Na página inicial, foi adicionado um campo de seleção para o usuário escolher a linguagem de programação do seu projeto.
- Após o usuário selecionar a linguagem, acontece um redirecionamento para uma nova tela onde serão exibidas as seções recomendadas.
- O front-end faz uma requisição para o Lambda processar uma linguagem e outra para recuperar os arquivos salvos no S3, contendo os dados das seções recomendadas para a linguagem.

⁴<https://pt-br.reactjs.org/>

3.2.2 LAMBDA

O Lambda⁵ é um serviço que executa códigos nas linguagens de programação suportadas, sem precisar se preocupar com escalabilidade ou gerenciamento de infraestrutura. A cobrança do serviço é feita pela quantidade de recurso utilizada (CPU e memória) pelo tempo de execução.

Utilizamos este serviço para receber requisições de processamento de uma linguagem do WRITEME. O código se comunica com o DynamoDB para verificar se a linguagem já foi processada anteriormente e se o resultado do processamento ainda está válido. Se estiver fora dessas duas condições, o Lambda vai submeter uma nova tarefa na fila do Batch.

3.2.3 DYNAMODB

O DynamoDB⁶ é um serviço de banco de dados NoSQL rápido e flexível e de alta disponibilidade. Sua cobrança é feita de acordo com o fluxo de leitura e escrita de dados no banco.

Este serviço foi utilizado para armazenar os dados das validades de processamento por linguagem de programação e a data da última vez que a linguagem foi processada pelo WRITEME.

3.2.4 BATCH

O Batch⁷ é um serviço de processamento em lote, muito utilizado para aprendizagem de máquina. Com ele, é possível reduzir custos definindo os requisitos necessários de recurso da máquina. A cobrança é feita com o cálculo da quantidade de recurso utilizado vezes o tempo de execução.

Nas configurações do Batch, definimos as especificações da máquina necessárias para o processamento de WRITEME e a imagem Docker⁸ que é utilizada. Portanto, ao receber uma nova tarefa em sua fila, o serviço irá alocar os recursos definidos em alguma máquina disponível e processar a tarefa.

⁵<https://aws.amazon.com/pt/lambda>

⁶<https://aws.amazon.com/pt/dynamodb>

⁷<https://aws.amazon.com/pt/batch>

⁸<https://www.docker.com/>

3.2.5 DOCKER

Utilizamos também uma imagem Docker, criada para o WRITEME. Essa imagem é basicamente um conjunto de instruções para criação de um container que será executado na máquina do servidor.

Na imagem criada, usamos como base o Python3, instalando todas as bibliotecas necessárias para o classificador e algumas outras funções auxiliares, que servem para salvar os resultados extraídos pelo classificador no S3. Ao final do processo, também é salva a data do processamento da linguagem no DynamoDB.

3.2.6 S3

O S3⁹ é um serviço de armazenamento de arquivos em qualquer volume nos chamados “buckets”, que funciona como um diretório de arquivos na nuvem. Sua cobrança é feita pelo espaço utilizado de armazenamento e a transferência de dados (leitura e escrita).

Esse, por sua vez, foi utilizado para armazenar o resultado dos processamentos das linguagens, contendo as seções e árvores em arquivos no formato JSON. Os arquivos podem ser lidos pelo diretamente front-end acessando a URL pública do bucket.

3.3 ATUALIZAÇÕES DAS SEÇÕES RECOMENDADAS

Com o objetivo de manter os exemplos das seções recomendadas atualizado sem trazer maiores custos de processamento, optamos por definir uma data de expiração para as execuções anteriores. Foi realizado um estudo sobre a quantidade de dias para a validade dos arquivos gerados no processamento de cada linguagem de programação. Para isso, foram selecionadas as 25 linguagens classificadas como “Populares” no campo de filtro de linguagens do Github. Para cada uma delas, foi realizada uma busca, filtrando pelos repositórios que utilizam a linguagem e selecionando os 50 repositórios mais famosos, usando como critério a quantidade de estrelas recebidas por usuários.

Dentre os repositórios com mais estrelas, foram encontrados todos os commits do arquivo `README.md` e calculada a média aritmética da diferença de dias entre eles.

Para as linguagens de programação que não estão entre as populares do Github, foi feita uma nova média aritmética entre as médias das linguagens populares e definido

⁹<https://aws.amazon.com/pt/s3>

como o valor “default” (padrão, em português). A Tabela 1 mostra os resultados obtidos pelo estudo.

#	Linguagem	Média de dias
1	c	58
2	c#	75
3	c++	60
4	coffeescript	75
5	css	92
6	dart	35
7	default	66
8	dm	88
9	elixir	59
10	go	43
11	groovy	96
12	html	96
13	java	66
14	javascript	54
15	kotlin	51
16	objective-c	81
17	perl	80
18	php	122
19	powershell	63
20	python	59
21	ruby	62
22	rust	32
23	scala	68
24	shell	44
25	swift	54
26	typescript	38

Tabela 1: Média de dias entre alterações nos arquivos README.

4 METODOLOGIA

Realizamos uma pesquisa com o objetivo de avaliar a eficácia da ferramenta, analisando o impacto na qualidade do conteúdo de arquivos README gerados com seu auxílio. Para isso, compartilhamos o questionário com um conjunto de usuários proveniente das redes de contato profissional, pessoal e acadêmica dos autores, considerados parte do público alvo do WRITEME. Neste capítulo descrevemos a metodologia aplicada no questionário, detalhamos a pesquisa e os resultados obtidos a partir dela.

4.1 QUESTIONÁRIO

Disponibilizamos um questionário para usuários no qual a proposta era comparar a qualidade do conteúdo de arquivos README de sua autoria com um novo arquivo gerado com auxílio do WRITEME.

Inicialmente, os usuários escolheram o repositório que seria o objeto de estudo da pesquisa. A partir dessa escolha, os usuários responderam perguntas referentes ao README escrito previamente.

Após responder as perguntas iniciais, os participantes foram instruídos a selecionar no WRITEME a linguagem de programação de seu projeto escolhido e para receber as recomendações da ferramenta para aquela linguagem.

Em seguida foi solicitado que os usuários analisassem quais das seções recomendadas eram relevantes para o seu projeto. Então, eles puderam produzir seus conteúdos a partir das seções escolhidas e dos textos de referência anexados a cada uma das seções recomendadas.

Na última etapa do questionário, os participantes responderam perguntas referentes ao conteúdo da nova versão do arquivo que foi escrito com auxílio da ferramenta. Esta etapa visou avaliar a aplicabilidade do WRITEME no processo de redação dos arquivos README, a partir da percepção do usuário sobre o resultado final.

Todas as perguntas foram respondidas através de um formulário online que contou com a participação de 20 usuários.

5 ANÁLISE DE RESULTADOS

Esta seção detalha cada pergunta do formulário, resultados e análises das respostas obtidas no formulário.

Na Figura 5, pode-se observar que 4 usuários possuem mais de cinco anos de experiência profissional com desenvolvimento de software, enquanto as categorias "até dois anos" e "entre 3 e 5" possuem 8 participantes cada uma.

Fizemos uma pergunta do tipo escala linear na Figura 6, onde 60% dos usuários consideram extremamente importante ter um arquivo README bem escrito e responderam com o valor máximo, 5. Enquanto isso, 30% das pessoas responderam com o valor 4, considerando muito importante. Por último, 10% consideraram "importante" ter um bom arquivo README. Nenhum participante achou que o README não é importante para o sucesso do projeto.

Ao relacionar o tempo de experiência com a opinião sobre a importância do README, obtivemos as médias de cada grupo, detalhadas na Tabela 2. Os participantes com menos tempo de experiência profissional em programação apresentaram uma média mais próxima da nota 4 e os demais grupos apresentaram uma média mais próximas da nota máxima.

Tempo de experiência	Média das notas
0-2 anos	4,125
3-5 anos	4,875
Mais de 5 anos	4,5

Tabela 2: Notas médias da importância dos arquivos README agrupadas por tempo de experiência dos participantes

Os participantes escolheram linguagens de programação variadas, totalizando 11 linguagens diferentes. A mais utilizada no questionário foi JavaScript, contando com 5 escolhas. A Tabela 3 mostra a quantidade de usuários que escolheram cada uma delas.

Perguntamos as fontes utilizadas pelo participante na época que criou o arquivo README antes das alterações. Na Figura 7 podemos ver que a alternativa mais escolhida foram os projetos famosos do GitHub, marcada por metade dos usuários. Esse pode ser considerado um indicador positivo para a validação do WRITEME, pois a ferramenta extrai as seções recomendadas dos projetos famosos da linguagem escolhida. A imagem

20 respostas

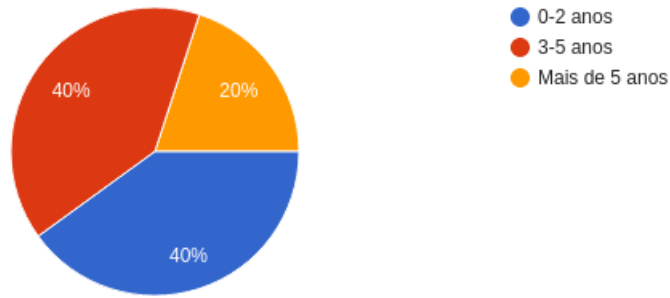


Figura 5: Respostas à pergunta: “Quanto tempo de experiência profissional com desenvolvimento de software você tem?”

20 respostas

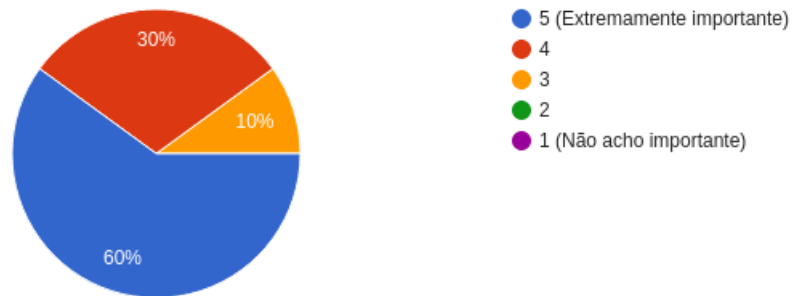


Figura 6: Respostas à pergunta: “Quão importante você considera que é ter um bom README para o sucesso de um projeto?”

mostra também que é comum também utilizar como fonte projetos de outras linguagens de programação.

Na última etapa do questionário, os usuários responderam as perguntas referentes ao conteúdo do arquivo reescrito com auxílio do WRITEME. Na Figura 9, apenas 1 usuário não utilizou nenhuma das recomendações da ferramenta para reescrever seu novo arquivo README, enquanto a maioria utilizou apenas as seções recomendadas.

Comparando os valores exibidos no gráfico da ??, houve um aumento considerável na quantidade de participantes que classificaram como “bom” o conteúdo do seu README após as alterações.

A Figura 10 detalha que 70% dos participantes tiveram a percepção de que a nova versão do arquivo ficou melhor que a anterior. Por outro lado, 25% não julgou que houve

Linguagem	Quantos usuários escolheram
C	1
C#	1
C++	1
Dart	1
GDScript	1
Java	2
JavaScript	5
PHP	1
Python	4
Swift	2
TypeScript	1

Tabela 3: Respostas à pergunta: “Qual a linguagem de programação do projeto escolhido?”

20 respostas

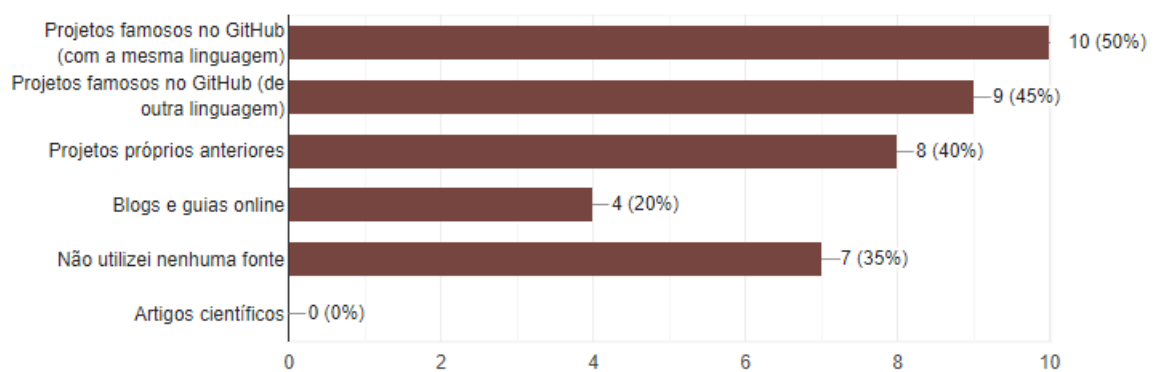


Figura 7: Respostas à pergunta: “Quais fontes você utilizou na época para saber que tipo de conteúdo deveria constar no README antigo?”

uma melhora significativa no material.

No total, 5 participantes compartilharam os seus repositórios utilizados para responder o questionário e foi possível comparar a estrutura dos arquivos antes e depois da utilização do WRITEME, usando como referência a quantidade de seções e subseções. Essa comparação está exibida na Tabela 4, onde conseguimos notar um aumento na quantidade de seções utilizadas em todos os casos.

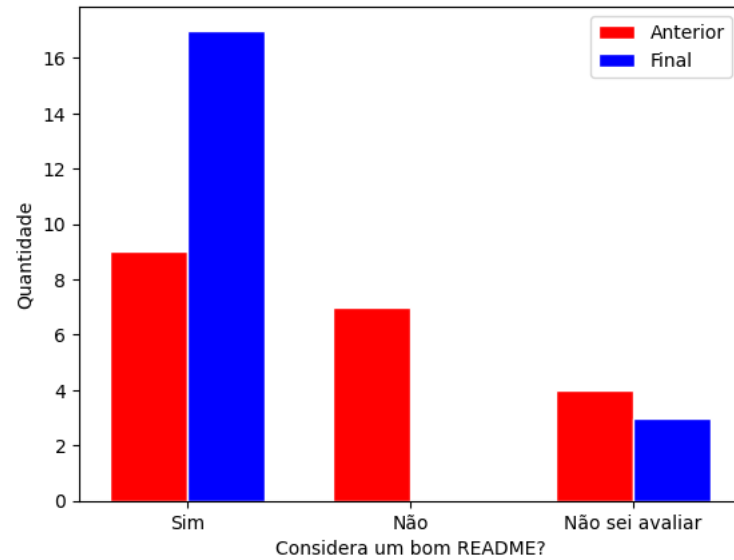


Figura 8: Respostas à pergunta: “Você considera o README como um bom README?” entre a versão anterior e final

20 respostas

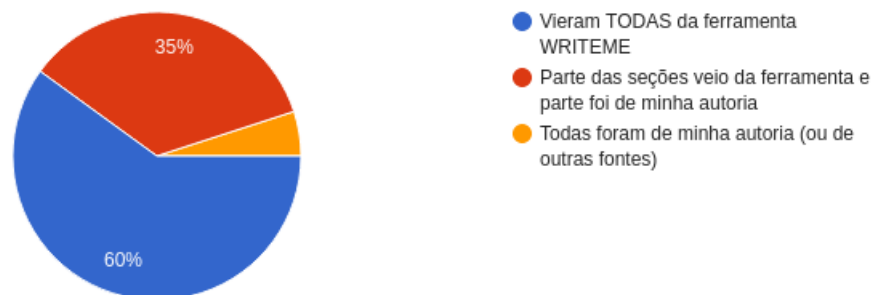


Figura 9: Respostas à pergunta: “Neste novo README, de onde vieram as seções utilizadas?”

Participante	Versão anterior		Versão final	
	Seções	Subseções	Seções	Subseções
Participante A	4	0	6	4
Participante B	4	0	8	4
Participante C	3	0	5	0
Participante D	3	11	6	11
Participante E	2	0	4	0

Tabela 4: Comparação entre as quantidades de seções e subseções dos arquivos antes e após o uso do WRITEME

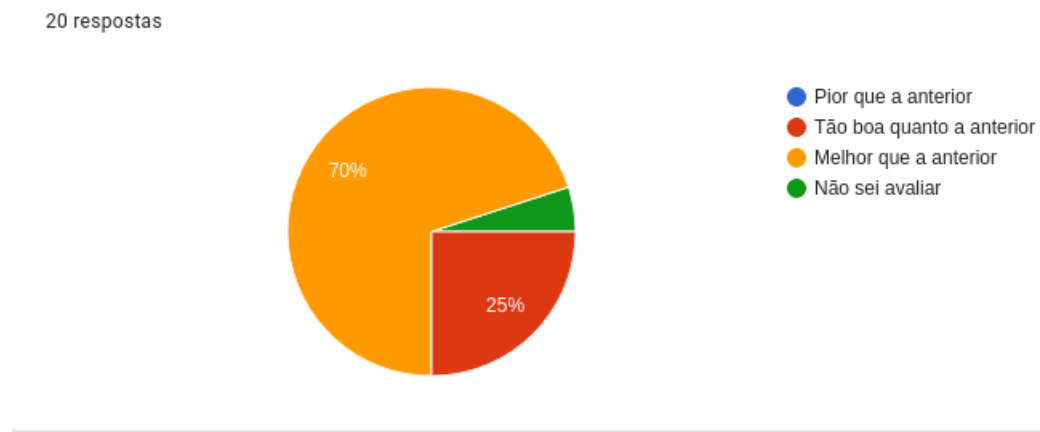


Figura 10: Respostas à pergunta: “Em comparação com o README anterior, essa nova versão é:”

5.1 ESTUDO COMPLEMENTAR

Dentre os 20 participantes, 8 concordaram em compartilhar seus arquivos e com isso foi possível realizar um estudo, utilizando inteligência artificial para classificar o conteúdo do arquivo. Essa classificação não foi integrada com o WRITEME e tem caráter genérico.

Para o estudo, utilizamos o ChatGPT,¹⁰ um modelo de linguagem artificial desenvolvido pela empresa OpenAI, baseado na arquitetura GPT-3.5, com técnicas de aprendizagem por reforço com feedback humano [8]. Este modelo apresenta uma habilidade considerável de responder perguntas sobre uma grande variedade de assuntos.

Solicitamos que o ChatGPT classificasse o conteúdo do arquivo README como bom, regular ou ruim e em seguida, inserimos o conteúdo de cada arquivo. Como resultado, a IA considerou que 7 deles estavam bem escritos. Segundo a IA, o arquivo restante foi classificado como regular pelo fato de não apresentar detalhes da tecnologia utilizada, como o sistema funciona e como contribuir para o projeto. A Figura 11 mostra um trecho da entrada enviada para o ChatGPT, contendo o comando para classificar o arquivo README e, logo abaixo o seu conteúdo. Já na Figura 12, exibimos a análise escrita pela IA.

```
quero que você classifique como bom, regular ou ruim. o conteúdo deste arquivo README
# Wine Pairing
An app that finds wines that go well with a food.

## Goals
This project was developed to research and study about accessibility in mobile apps.

## Overview
With VoiceOver on      | With VoiceControl on
```

Figura 11: Comando de entrada utilizado no ChatGPT

Comparando as análises dos próprios participantes com a do ChatGPT, houve uma similaridade em 7 das 8 respostas, onde ambos classificaram o conteúdo como “bom”. No caso de divergência, o participante avaliou o conteúdo do seu arquivo como “bom” enquanto a IA avaliou como “regular”.

¹⁰<https://chat.openai.com/chat>



Eu classificaria o conteúdo deste arquivo README como bom. Ele fornece informações úteis sobre o projeto, incluindo seus objetivos, recursos e requisitos, bem como instruções claras para configurar e executar o aplicativo. O uso de imagens para ilustrar as funcionalidades do aplicativo é um ponto positivo, e a inclusão de informações sobre a autora e a licença utilizada é adequada. Além disso, o arquivo é bem organizado e fácil de ler.

Figura 12: Respostas à pergunta: “Em comparação com o README anterior, essa nova versão é:”

6 CONCLUSÃO

Neste trabalho, expandimos o uso do WRITEME para outras linguagens de programação e fizemos uma nova pesquisa com um número maior de pessoas que utilizaram a ferramenta com um aumento de 6 para 20 participantes. A partir da pesquisa de validação, pudemos observar que houve uma melhora significativa na qualidade dos arquivos gerados com auxílio da ferramenta, tanto pela percepção dos próprios usuários quanto pelo estudo aplicando inteligência artificial, detalhado na seção 5.1.

Coletamos opiniões dos participantes sobre a ferramenta e também analisamos o estado atual da mesma para projetar os próximos trabalhos que precisam ser feitos a fim de elevar sua qualidade.

Segundo os próprios usuários, o WRITEME recomendava uma quantidade grande de seções, nem sempre relevantes e isso acabava atrapalhando sua jornada. Os resultados da pesquisa apontaram também que alguns usuários não souberam avaliar a qualidade de seus arquivos README.

7 TRABALHOS FUTUROS

Identificamos que precisa ser feito um estudo sobre o número mínimo de ocorrências das seções entre os READMEs mais famosos para que elas sejam recomendadas. Para ajudar os usuários que não souberam avaliar o conteúdo final de seus arquivos, seria útil um auxílio da própria ferramenta para classificar o conteúdo gerado e apontar possíveis pontos de melhoria.

Seria interessante que a aplicação possuísse um editor de texto online para facilitar a jornada do usuário durante a manipulação das suas seções escolhidas. Com ele, seria possível adicionar, remover e reordenar as sugestões a qualquer momento e o usuário só precisaria baixar o arquivo ao final do processo.

REFERÊNCIAS

- [1] COELHO, H. S. Documentação de software: uma necessidade. *Texto Livre: linguagem e tecnologia*, Universidade Federal de Minas Gerais, v. 2, n. 1, p. 17–21, 2009.
- [2] PINTO, G. *Documentação de software: por que é tão importante e o que sabemos sobre ela?* Disponível em: <https://www.zup.com.br/blog/documentacao-de-software>, acessado em 30/03/2023.
- [3] GitHub, Inc. *About READMEs*. Disponível em: <https://docs.github.com/en/repositories/managing-your-repositorys-settings-and-features/customizing-your-repository/about-readmes>, acessado em 30/03/2023.
- [4] WANG, T.; WANG, S.; CHEN, T.-H. P. Study the correlation between the readme file of github projects and their popularity. *Available at SSRN 4281782*.
- [5] FOGEL, K. *Producing Open Source Software: How to Run a Successful Free Software Project*. [S.l.]: O'Reilly Media, 2005.
- [6] LEITE, H. P. B. *WRITEME: uma Ferramenta de Auxílio à Escrita de READMEs Baseada em Dados*. Dissertação (Bacharelado) — UFPE, 2019.
- [7] PRANA, G. A. A. et al. Categorizing the content of github readme files. *Empirical Software Engineering*, Springer, v. 24, p. 1296–1327, 2019.
- [8] OPENAI. *Introducing ChatGPT*. Disponível em: <https://openai.com/blog/chatgpt>, acessado em 08/04/2023.