Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Mineração de Questões Sobre o Uso de Expressões Lambda em Java 8

Thiago Penha Torreão

Monografia apresentada como requisito parcial
para conclusão do Bacharelado em Ciência da Computação

Orientador
Prof. Dr. Rodrigo Bonifácio de Almeida

Brasília
2018

Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Mineração de Questões Sobre o Uso de Expressões Lambda em Java 8

Thiago Penha Torreão

Monografia apresentada como requisito parcial
para conclusão do Bacharelado em Ciência da Computação

Prof. Dr. Rodrigo Bonifácio de Almeida (Orientador)
CIC/UnB

Prof. Dr. Fabiano Cavalcanti Fernandes
IFB

Prof.a Dr.a Edna Dias Canedo
CIC/UnB

Prof. Dr. Rodrigo Bonifácio de Almeida
Coordenador do Bacharelado em Ciência da Computação

Brasília, 23 de julho de 2018

Dedicatória

Dedico este trabalho à minha família que sempre me apoiou e aos meus amigos e colegas que estiveram presentes durante este longo período de estudo da minha vida. Dedico também aos professores que contribuíram para a minha formação acadêmica, em especial ao meu orientador, professor doutor Rodrigo Bonifácio de Almeida, que contribuiu para que eu tenha conseguido finalizar minha graduação com sucesso.

Agradecimentos

Agradeço à minha família, que me apoiou ao longo de toda a minha vida, e que me deu todo o suporte que eu precisei durante a minha formação acadêmica. Agradeço também aos meus colegas de curso, especialmente aos que ingressaram comigo em 2010, por toda a ajuda que me foi dada durante todos estes anos de curso. Por fim, agradeço ao meu orientador, professor doutor Rodrigo de Almeida Bonifácio, pelo auxilio que me foi dado durante os semestres finais do curso.

Resumo

Este trabalho apresenta um estudo sobre características presentes em perguntas relacionadas com o uso de expressões lambda em Java 8. Este estudo foi realizado através da mineração de perguntas e respostas do site *Stack Overflow*, de onde foram encontradas 1975 perguntas e 3974 respostas relacionadas com este tema. Estes dados foram usados para verificar como o interesse dos usuários do *Stack Overflow* em fazer estas perguntas variou ao longo do tempo e se essas perguntas costumam ser respondidas adequadamente. Foi feita também uma análise de sentimentos nestas perguntas e suas respostas para tentar verificar quais sentimentos são expressados pelos seus autores. Além disso foram procurados os principais tópicos abordados nestas perguntas e, para isso, foram selecionadas as 100 perguntas mais populares que foram lidas uma a uma. A leitura destas perguntas possibilitou não só ajudaram a identificar os tópicos mais populares, mais também pôde ser usada para complementar os resultados da análise de sentimentos.

Palavras-chave: Mineração de dados, Java 8, Expressões Lambda, Stack Overflow

Abstract

This work presents a study about features identified in questions about the usage of lambda expressions in Java 8. The study was done by means of mining questions and answers from the site Stack Overflow, from which were found 1975 questions and 3974 answers related to this theme. The aquired data was used to check how the interest in asking these questions changed with time, and whether this questions are answered successfully. Also, a sentiment analysis was done to try and see what sentiments are expressed by the authors of these questions and answers. Furthermore, the main topics addressed in these questions were looked for and to accomplish this, the 100 most popular questions were selected and read one by one. After reading these questions it was not only possible to better identify the most popular topics among the questions, it was also possible to complement the results of the sentiment analysis.

Keywords: Data Mining, Java 8, Lambda Expressions, Stack Overflow

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Abreviaturas e Siglas

API Interface de Programação de Aplicações.

CRISP-DM CRoss-Industry Standard Process for Data Mining.

JDK Java Development Kit.

MSR Mineração em Repositórios de Software.

NLTK Natural Language Toolkit.

VADER Valence Aware Dictionary for sEntiment Reasoning.

Capítulo 1

Introdução

1.1 Contexto

A linguagem de programação Java, por muito tempo, não dava suporte substancial à programação funcional. Quando era necessário uma abordagem mais funcional em programas Java, eram utilizadas as chamadas *Anonymous Inner Classes* (classes anônimas). Essas classes são semelhantes a classes locais, com a diferença que elas não possuem um nome[1]. Elas permitem o programador a declarar e instanciar a classe ao mesmo tempo e são usadas para deixar o código mais conciso, mas só podem ser utilizadas uma única vez. Embora as classes anônimas são mais concisas que classes nomeadas, elas ainda são muito extensas e podem não ser muito claras em implementações mais simples, por exemplo em interfaces que contêm apenas um único método.

Isso mudou em 2014, com o lançamento do Java Development Kit (JDK) 8[2]. Esta nova versão adicionou à linguagem, dentre outras coisas, o suporte a expressões lambda. Expressões lambda permitem o programador a escrever instâncias de classes com um único método mais compactamente, e a tratar funcionalidade como um método, ou tratar código como dados[3].

Contudo, as expressões lambda não foram adicionadas à linguagem Java simplesmente como uma alternativa ao uso de classes anônimas. Java 8 também trouxe maneiras diferentes de se poder fazer uso dessas expressões. Dentre estas maneiras destacam-se a Interface de Programação de Aplicações (API) *Stream* e os *Method references*. A API *Stream* é utilizada para dar suporte ao uso de operações com estilo mais funcional em *streams* de elementos, uma nova abstração de dados que pode ser criada a partir de *Collections*, *arrays*, ou de métodos específicos[4]. *Method references* funcionam como uma expressão lambda ainda mais compacta e fácil de ler que chama um método já existente[5].

Todas estas adições, torna mais factível o uso de abordagens mais próximas do paradigma funcional em programas Java, principalmente para a confecção de códigos mais

concisos e legíveis. Mas elas também incitam a pergunta: *"Como os programadores Java estão fazendo uso dessa característica que só foi adicionada à linguagem depois de tanto tempo?"*. Tendo isso em vista, este trabalho visa realizar um estudo empírico baseado na mineração de questões do site *Stack Overflow*, um popular fórum de perguntas e respostas sobre problemas relacionadas com desenvolvimento de softwares, para entender melhor como os programadores Java estão utilizando as expressões lambda.

1.2 Objetivos

1.2.1 Objetivos Gerais

Entendendo as novas possibilidades que as expressões lambda dão para a escrita de programas Java. Surge também a necessidade da realização pesquisas sobre esse tema. Sabendo disso, este trabalho tem como objetivo geral a obtenção de novas informações úteis sobre o uso de expressões lambda na linguagem de programação Java.

1.2.2 Objetivos Específicos

Sabendo da possibilidade de se utilizar dados coletados do site *Stack Overflow* para a obtenção de novas informações a respeito de temas relacionados com o desenvolvimento de software. Os objetivos específicos deste trabalho consistem em utilizar dados coletados desse site para responder às seguintes questões de pesquisa:

Q1: Que peculiaridades podem ser encontradas em perguntas relacionadas com o uso de expressões lambda em Java 8 no site *Stack Overflow*?

Q2: Quais são os sentimentos expressados em perguntas e respostas relacionadas com o uso de expressões lambda em Java 8?

Q3: Quais são os principais tópicos presentes nas perguntas relacionadas com o uso de expressões lambda em Java 8?

1.3 Organização do Trabalho

Este trabalho está dividido em 5 capítulos:

- **Capítulo 2:** Apresenta uma a base teórica sobre mineração de dados, análise de sentimentos, e mineração em repositórios de software.
- **Capítulo 3:** Faz uma descrição do estudo realizado para se atingir os objetivos deste trabalho.

- **Capítulo 4:** Apresenta os resultados obtidos e as análises realizadas sobre eles.
- **Capítulo 5:** Apresenta as conclusões gerais obtidas a partir deste trabalho.

Capítulo 2

Mineração de Dados

Este capítulo apresenta a teoria básica sobre mineração de dados e uma breve introdução aos conceitos de análise de sentimentos e mineração em repositórios de software. Além disso também são abordados alguns trabalhos relacionados a este.

2.1 O Processo de Mineração de Dados

De acordo com Witten et al.[6], mineração de dados é definida como o processo de descobrir padrões em dados. Esta ideia não é algo recente, seres humanos, ao longo da história, já vêm observando padrões em muitas situações diferentes. Exemplos dessas situações são: quando um caçador identifica padrões nos hábitos dos animais que caçam para facilitar as suas caças, ou quando um fazendeiro observa padrões no crescimento de suas plantações, para maximizar sua colheita. O diferencial da mineração de dados, quando comparada com essas situações, é que os dados utilizados são armazenados eletronicamente em computadores e o processo de descobrir os padrões deve ser automático ou, como é mais comum, semi-automático. Além disso, os padrões encontrados devem fornecer novos conhecimentos ou possibilitar tomadas de decisões rápidas e precisas.

O ciclo de vida de um processo de mineração de dados, de acordo com o modelo CRoss-Industry Standard Process for Data Mining (CRISP-DM), é dividido nas seguintes fases[7]:

1. ***Business understanding*:** Esta fase foca em entender os objetivos do processo que será realizado, usar este entendimento para convertê-lo em um processo de mineração de dados e desenvolver planos preeliminares para atingir os objetivos desejados.

2. ***Data understanding*:** Esta fase se inicia com a coleta dos dados. Depois disso é realizada uma exploração inicial nos dados para detectar padrões que possam revelar

informações escondidas, identificar problemas que possam afetar a qualidade dos dados e descobrir peculiaridades em geral sobre estes dados.

3. ***Data preparation*:** Esta fase consiste no processamento dos dados para que estes possam ser modelados. Isso inclui seleção, refinamento, construção, integração e formatação dos dados.

4. ***Modeling*:** Esta fase envolve o uso de técnicas de modelagem que criam modelos a partir dos dados processados. Isso inclui a seleção das técnicas apropriadas, a criação dos modelos e a avaliação dos modelos.

5. ***Evaluation*:** Nesta fase os resultados da mineração de dados são avaliados para determinar se os objetivos foram atingidos de forma satisfatória. Além disso o processo em si é revisto para conferir se nenhuma tarefa foi negligenciada.

6. ***Deployment*:** Esta fase envolve a organização e a apresentação do conhecimento adquirido a partir do processo. Dependendo do que motivou a existência desse processo, isso pode ser tão simples quanto a confecção de um relatório, ou tão complexo quanto a implementação de sistemas que realizem um processo de mineração repetível para uma empresa.

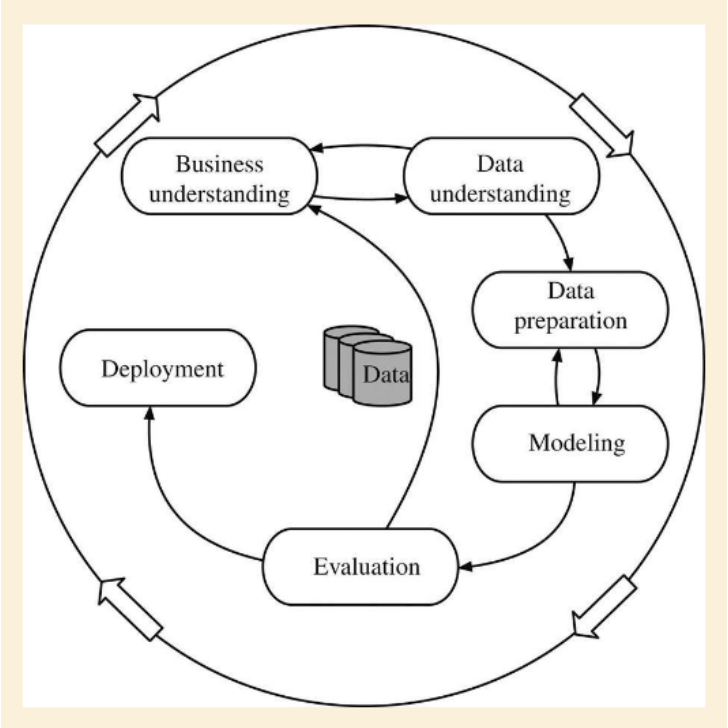

Figura 2.1: Ciclo de vida de um processo de mineração de dados, segundo o modelo CRISP-DM (Fonte: [6]).

2.1.1 Técnicas de Mineração de Dados

A mineração de dados envolve a extração de informação útil a partir de dados. Estes dados vêm na forma de um conjunto de instâncias, cada uma caracterizada por valores de atributos que medem diferentes aspectos delas[6]. Para modelar estes dados existem diversas técnicas e algoritmos que podem ser utilizados. Algumas técnicas bastante utilizadas são:

Classificação

Classificação é a técnica de mineração de dados mais utilizada[8]. Esta técnica usa uma abordagem de aprendizado de máquina onde um conjunto de exemplos já classificados é utilizado para desenvolver modelos que podem classificar outros exemplos. A classificação dos dados pode ser dividido em duas etapas, na primeira o algorítimo de classificação analisa os dados de treinamento e na segunda é verificada a precisão das regras de classificação geradas.

Associação

Regras de associação são semelhantes às regras de classificação, mas apresentam funcionalidades diferentes. Regras de associação podem prever qualquer atributo, não somente a classe como regras de classificação, e podem também prever mais de um atributo em um mesmo tempo. Regras de associação costumam ter apenas atributos não-numéricos[6].

Árvores de decisão

Árvores de decisão é um método de aprendizado de máquina bastante utilizado em mineração de dados. O seu uso consiste no teste de um atributo em particular em cada nó da árvore. Geralmente estes testes são entre um atributo e uma constante, mas também pode ser feito uma comparação entre atributos e, dependendo das comparações feitas em seus nós, cada folha da árvore pode ser uma classe de instâncias ou uma associação entre atributos. Além disso também é possível utilizar esse mesmo tipo de estrutura para realizar predições numéricas, neste caso cada folha representa a média de todos os valores que se aplicam a ela [6].

Clusterização

Clusterização é aplicada quando o objetivo não é separar as instâncias em classes, mas sim agrupá-las em *clusters*. Neste caso o resultado é apresentado na forma de um diagrama que mostra em qual *cluster* cada instância se encontra. Embora *clusters* já sejam uma

possível forma de representar os resultados da mineração, é comum que sejam criados um conjunto de regras ou uma árvore de decisão que alocam cada instância no *cluster* ao qual pertencem[6].

Regressão linear

A ideia da regressão linear é de expressar as classes como uma combinação linear dos atributos, utilizando pesos adequados em cada atributo. Esta técnica é naturalmente adequada a situações nas quais tanto os resultados quanto os atributos utilizados sejam todos numéricos.[6]

Técnicas de aprendizado de máquina

Um processo mineração de dados assemelha-se a um processo de aprendizado de máquina no aspecto de que ambos são atividades realizadas para se obter informações novas a partir de informações já disponíveis. Com isso, aplicações de mineração de dados podem se beneficiar do uso de várias técnicas de aprendizado de máquina. Alguns dos estilos de aprendizado que aparecem comumente em aplicações de mineração de dados são:[6]

- ***Classification learning*:** o programa visa aprender a classificar exemplos a partir de exemplos já classificados.
- ***Association learning*:** o que é procurado é qualquer associação entre atributos, não somente associações que predizem uma classe específica.
- ***Clustering*:** o objetivo não é prever classes mas agrupar as instâncias em *clusters*.
- ***Numeric prediction*:** o que é previsto não é uma categoria, mas sim um valor numérico.
- ***Instance-based learning*:** ao invés de criar regras, o trabalho é feito em cima das instâncias em si, comparando instâncias novas a instâncias que já existem para definir a qual classe a instância nova pertence.

Representação por probabilidade

Outro aspecto importante de ser considerado em processos de mineração de dados é a precisão dos resultados. É comum que os dados de onde são retiradas as informações contenham erros ou estejam incompletos, o que pode gerar resultados imprecisos. Por isso, muitas vezes estimar a probabilidade de uma instância pertencer a uma classe ou um *cluster* pode ser mais útil que simplesmente prever a qual classe ou *cluster* a instância pertence, por exemplo. Representando os resultados com probabilidades permite ranquear

as previsões, baseando-se nas chances estimadas de acertos e erros, para assim obter-se resultados ainda mais precisos no final da mineração.[6]

2.2 Análise de Sentimentos

A análise de sentimentos é uma das diversas abordagens da mineração de dados. Segundo Bing Liu[9], análise de sentimentos, que também pode ser chamada de mineração de opinião, é o estudo computacional de opiniões, sentimentos e emoções expressados em um texto. Ela é um problema desafiador sobre processamento de linguagem natural e mineração de textos, que apresenta grande crescimento tanto entre pesquisas científicas, quanto em aplicações na indústria. As áreas mais pesquisadas sobre esse tema são:

- **Classificação de sentimentos e de subjetividade:** É a área mais pesquisada sobre análise de sentimentos. Nela, a análise de sentimentos é tratada como um processo de classificação de texto. Os dois tópicos mais estudados nessa área são: classificar um texto opinativo como sendo positivo ou negativo; e classificar uma frase ou oração como sendo objetiva ou subjetiva e, caso seja subjetiva, classificá-la com expressando uma opinião positiva, negativa, ou neutra.
- **Análise de sentimentos baseada em características:** Nela, primeiro o modelo identifica um objeto sobre o qual as opiniões estão sendo expressadas. Este objeto pode ser um produto, um serviço, um indivíduo, uma organização etc.
- **Análise de sentimentos de frases comparativas:** Aqui, a análise de um objeto é feita através de frases que o comparem a um outro objeto similar. Por exemplo: "produto X é melhor que produto Y".
- **Procura e obtenção de opinião:** Esta área consiste em processos nos quais, dado um tema específico, é feita uma busca por documentos ou frases relevantes a esse tema, e a identificação e classificação destes documentos ou frases.
- **Detecção de spam e utilidade de opiniões:** Spam de opiniões refere-se a opiniões falsas usadas para enganar os leitores. Isso inclui opiniões positivas sobre objetos que não as merecem, para promovê-lo, ou opiniões negativas maliciosas para denegrir o objeto. Utilidade de opiniões refere-se à qualidade das opiniões.

Recentemente uma grande quantidade de métodos, técnicas e melhorias têm sido propostas para lidar com problemas de análises de sentimentos com diferentes objetivos e em diferentes níveis[10]. Para medir automaticamente os sentimentos presentes em um texto foram propostas medidas semânticas que medem subjetividade e opiniões em textos[11].

Estas medidas costumam medir a polaridade e a "força", ou intensidade, de um documento. A polaridade pode ser positiva ou negativa. A intensidade mostra o quanto o texto é positivo ou negativo em relação à um tópico. Por exemplo um texto com opinião positiva de intensidade 1 é mais positivo que um texto de opinião positiva com intensidade 0.5.

Algoritmos de análise de sentimentos podem apresentar uma abordagem baseada em aprendizado de máquina ou uma abordagem léxica[11]. No caso de aprendizado de máquina são utilizados textos de entrada para gerar classificadores capazes de classificar novos textos. A abordagem léxica é um problema de processamento de linguagem natural que utiliza listas, ou dicionários, de palavras e expressões relacionadas a sentimentos já conhecidas, suas polaridades, e a estrutura gramatical da linguagem para medir o sentimento expressado em um texto. Há também a possibilidade de uma abordagem híbrida, que combina recursos léxicos e de aprendizado de máquina[10].

2.3 Mineração em Repositórios de Software

Existem diversos repositórios de softwares comumente utilizados em projetos de desenvolvimento de softwares, por exemplo repositórios de controle de versões. Devido ao advento de sistemas de código aberto, o acesso a repositórios de sistemas de software de grande porte ficou mais fácil. De acordo com Ahmed E. Hassan[12], a Mineração em Repositórios de Software (MSR) é um campo da mineração de dados que envolve a mineração e a análise dos dados presentes em tais repositórios para descobrir informações interessantes sobre o desenvolvimento de sistemas de software. Alguns exemplos de repositórios de software que podem ser minerados são:

- **Repositórios de controle de versão:** Estes repositórios registram o histórico de desenvolvimento de um projeto. Eles guardam informações a respeito de todas as mudanças realizadas no código fonte, junto de metadados sobre essas mudanças, por exemplo o nome do autor da mudança e a hora em que ela foi feita. Exemplos desse tipo de repositório são o *Git* e o *Subversion*.

- **Repositórios de *bugs*:** Estes repositórios registram o histórico de relatórios de resolução de *bugs*, ou pedidos de correção de *bugs* feitos por usuários ou desenvolvedores de projetos grandes. Alguns exemplos desses repositórios são o *Bugzilla* e o *Jira*.

- **Comunicações arquivadas:** Estes repositórios registram discussões sobre vários aspectos de um projeto de desenvolvimento de software, ao longo do seu ciclo de vida. Exemplos desses repositórios são listas de e-mails e chats IRC.

- ***Logs* de implantação:** Estes repositórios registram informações sobre a implantação de uma aplicação, ou sobre múltiplas implantações de uma mesma aplicação. Por exemplo, *logs* de implantação podem registrar as mensagens de erros reportadas durante a implantação de uma aplicação em vários ambientes.
- **Repositórios de códigos:** Estes repositórios registram o código fonte de um grande número de projetos de desenvolvimento de software. Exemplos desses repositórios são o *Sourceforge.net* e o *Google code*.

Repositórios de software, segundo Hassan[12], costumam ser usados apenas para guardar registros do desenvolvimento de softwares e raramente são utilizados para ajudar em tomadas de decisões. Pesquisadores de MSR visam fazer com que esses repositórios deixem de ser apenas registros estáticos e passem a ter um papel mais dinâmico para guiar processos de tomadas de decisões em projetos de desenvolvimento de softwares.

A pesquisa sobre MSR foca primariamente em dois aspectos. O primeiro é a criação de técnicas para automatizar ou melhorar a extração de informação de repositórios. Isso é importante para facilitar que outros pesquisadores adotem técnicas de MSR. O segundo aspecto é a descoberta e validação de novas técnicas e abordagens para minerar informações importantes destes repositórios. Isso é importante para mostrar relevância de informações armazenadas em repositórios de softwares, o que encoraja a adoção de técnicas de MSR. Algumas contribuições que processos de MSR podem oferecer para os sistemas de desenvolvimento de softwares, ou para pesquisadores envolvem[12]:

- Um melhor entendimento dos sistemas de desenvolvimento
- Propagação de mudanças[1]
- Predição e identificação de *bugs*
- Melhor entendimento de dinâmicas de grupo
- Melhorar a experiência de usuários
- Reuso de código
- Automatização de estudos empíricos

2.4 Trabalhos Relacionados

Como mencionado no Capítulo 1, este trabalho tem como objetivo estudar o uso de expressões lambda em Java 8. A abordagem utilizada para isso foi a mineração de perguntas

[1]A propagação de mudanças é o processo de propagar mudanças realizadas em um código para outra entidades do sistema de desenvolvimento, a fim de garantir a consistência do sistema.

e respostas do site *Stack Overflow* relacionadas com o tema a ser estudado. Já existem estudos realizados a respeito do uso de expressões lambda em Java 8, assim como também existem estudos que utilizam os dados do site *Stack Overflow* para a obtenção de novas informações. Dentre estes estudos, é relevante citar:

- Um trabalho que busca caracterizar o uso de expressões lambda entre os projetos Java mais populares desenvolvidos pela comunidade do *GitHub*[13]. Mais especificamente, este trabalho focou em utilizar técnicas de MSR para investigar a refatoração de códigos legados para a adoção de expressões lambda em programas que migraram para o Java 8.
- Um estudo que utiliza a mineração de perguntas do *Stack Overflow* para entender qual o interesse de programadores pelo consumo de energia de suas aplicações e como eles lidam com problemas relacionados a esse consumo[14].
- Uma pesquisa na qual um dos estudos realizados utiliza dados do *Stack Overflow* para verificar quais problemas relacionados ao uso de APIs de criptografia na linguagem Java[15] são encontrados por desenvolvedores.

Capítulo 3

O Estudo

Este capítulo apresenta o estudo realizado. Nele, são abordados: a metodologia adotada para a realização do estudo, as questões de pesquisa que serão respondidas, a fonte dos dados utilizados, os procedimentos realizados para se obter os resultados desejados e as ferramentas que foram utilizadas durante a realização do estudo.

3.1 Metodologia

A metodologia adotada para a realização deste estudo foi baseada modelo de ciclo de vida de um processo de mineração de dados apresentado no Capítulo 2. Assim é possível organizar o processo realizado em 6 fases, que são:

1. **Compreensão do processo:** Nesta fase, foi identificado o tema da pesquisa, isto é, o uso de expressões lambda em Java 8, e como este tema seria pesquisado, através da mineração de perguntas do site *Stack Overflow*.
2. **Compreensão dos dados:** Nesta fase os dados utilizados foram coletados e estudados para identificar o que poderia ser descoberto a partir deles. Nesta fase, também foram decididas as questões de pesquisa a ser respondidas.
3. **Preparação dos dados:** Nesta fase os dados coletados foram processados a fim de prepará-los para a coleta de informações.
4. **Modelagem dos dados:** Nesta fase os dados foram manipulados e organizados de forma que possibilitasse a obtenção das informações desejadas.
5. **Avaliação dos resultados:** Nesta fase os resultados obtidos foram analisados. Esta fase serviu não só para se obter as conclusões tiradas dos resultados obtidos, mas também para decidir se estes eram satisfatórios o suficiente. Para esse trabalho,

foi decidido a realização da seleção e leitura de 100 perguntas para complementar os resultados já obtidos.

6. **Apresentação dos resultados:** Nesta fase foi realizada a confecção deste documento, a fim de relatar os resultados obtidos e as análises realizadas sobre eles.

3.2 Questões de Pesquisa

As questões de pesquisa cujas respostas representam os objetivos a serem alcançados por este estudo são:

Q1: Quais são as peculiaridades de perguntas relacionadas com o uso de expressões lambda em Java 8 no site *Stack Overflow*?

Q2: Quais são os sentimentos expressados em perguntas e respostas relacionadas com o uso de expressões lambda em Java 8?

Q3: Quais são os principais tópicos presentes em perguntas relacionadas com o uso de expressões lambda em Java 8?

3.3 A Fonte dos Dados

Como informado no Capítulo 1, este estudo é feito tendo como base a mineração de perguntas do site *Stack Overflow*[1]. Ele é um dos sites da rede *Stack Exchange*[2], uma rede com diversas comunidades de perguntas e respostas, cada uma voltada para um tópico diferente. O *Stack Overflow* é um fórum de perguntas e respostas voltado para a solução de problemas e dúvidas relacionados ao desenvolvimento de softwares.

O *Stack Overflow* é bem popular entre programadores, que o utilizam para obter ajuda com problemas encontrados durante o desenvolvimento de seus programas. Não só isso, ele também popular entre pesquisadores, que usam os dados presentes nesse site para a realização de suas pesquisas. Essa popularidade já influenciou, em particular, que ele fosse o assunto de duas *Mining Challenges* propostas pela *International Conference on Mining Software Repositories*, em 2013[16] e em 2015[17]. Esses desafios contribuíram para a confecção de diversas pesquisas, que foram publicadas em *The 10th Working Conference on Mining Software Repositories*[18] e em *The 12th Working Conference on Mining Software Repositories*[19], respectivamente.

[1]https://stackoverflow.com/
[2]https://stackexchange.com/

Como já dito, o *Stack Overflow* é um portal voltado para a realização de perguntas que envolvam o desenvolvimento de softwares. O site apresenta algumas regras quanto ao conteúdo presente nas perguntas que se adequam ao seu contexto. É indicada a postagem de perguntas que focam em um problema real que o autor tenha encontrado, incluindo detalhes de como ele tentou resolver este problema e o que exatamente ele está tentando fazer. Mais especificamente o site recomenda perguntas sobre[20]:

- problemas específicos sobre programação ou;
- algorítimos de software ou;
- técnicas de escrita de código ou;
- ferramentas de desenvolvimento de software.

Nem todas as perguntas adequadas ao contexto do site, porém, necessariamente possuem as características recomendadas. Não obstante, perguntas que não se adequam a esse contexto podem ser marcadas pelos usuários do site como sendo "*off topic*"e fechadas pela comunidade do *Stack Overflow*[21]. Em casos mais extremos, a pergunta pode até mesmo ser excluída permanentemente do site. Assim, para facilitar a decisão de uma pergunta ser adequada ou não, o site também informa quais características não são desejadas para as suas perguntas. O site contraindica a postagem de perguntas que sejam primariamente baseadas em opiniões, ou que são mais propícias a gerar discussões ao invés de respostas. Mais especificamente, não devem ser feitas perguntas que[20]:

- o autor não tenha tentado encontrar a resposta anteriormente ou;
- sejam sobre recomendações de serviços ou produtos ou;
- peçam por listas, enquetes, opiniões, discussões etc. ou;
- sejam sobre qualquer coisa não diretamente relacionada com a escrita de programas.

3.3.1 Os Dados Coletados

Os dados de todos os sites da rede *Stack Exchange*, incluindo os dados do *Stack Overflow*, estão disponíveis desde 2009 para download no *Internet Archive*[3], sob a licença *Creative Commons BY-SA 3.0.*[4][22].

Os dados coletados para a realização deste estudo consistem no *dump* de postagens do *Stack Overflow*, baixado no dia 11 de dezembro de 2017. Este *dump* de dados vêm na forma de um arquivo *xml* que contém um total de 38,394,917 postagens, que vão desde o

[3]https://archive.org/details/stackexchange
[4]https://creativecommons.org/licenses/by-sa/3.0/

dia 31 de julho de 2008 até o dia 3 de dezembro de 2017. Destas postagens 14,995,834 são perguntas e 23,399,083 são respostas.

As postagens contidas no *dump* de dados adquirido, são caracterizadas por diversos atributos qualitativos e quantitativos. Os atributos mais relevantes para o que foi realizado neste estudo são:

- ***Id*:** O número de identificação da postagem.
- ***PostTypeId*:** Um atributo usado para classificar as postagens em perguntas ou respostas.
- ***ParentId*:** Um atributo presente em respostas para identificar qual pergunta esta resposta tenta responder.
- ***AcceptedAnswerId*:** Um atributo presente nas perguntas para identificar a resposta aceita por ela, caso exista.
- ***CreationDate*:** A data de criação da postagem.
- ***Body*:** O corpo textual das postagens.
- ***Title*:** O título de uma pergunta.
- ***Tags*:** Marcadores especiais utilizadas para identificar os tópicos abordados nas perguntas. Cada pergunta possui pelo menos uma tag e pode conter até cinco tags.
- ***Score*:** A pontuação da da postagem. Este atributo está presente tanto nas perguntas quanto nas respostas. Ele é baseado nas quantidades de *upvotes* e *downvotes* dados para a postagem em questão por usuários do site.
- ***ViewCount*:** O número de visualizações da postagem. É baseado no número de vezes que a página de uma pergunta é acessada. Como as respostas ficam na página da pergunta a qual elas respondem, este atributo é dado apenas para as perguntas.
- ***AnswerCount*:** O número de respostas dadas para uma pergunta.
- ***CommentCount*:** O número de comentários presentes na postagem.
- ***FavoriteCount*:** O número de usuários que estão marcando a pergunta em questão como uma favorita. Marcar uma pergunta como favorita faz com que quem o fez seja notificado quando houver alguma alteração no estado da pergunta, por exemplo: quando esta é editada ou recebe uma nova resposta.

3.4 Procedimentos

O estudo foi iniciado com a definição dos objetivos gerais a serem atingidos e da abordagem a ser utilizada para se atingir tais objetivos. Após isso foi realizada a coleta do *dump* de dados das postagens do site *Stack Overflow* e, em seguida foi realizado uma análise inicial dos dados coletados. Esta análise serviu para identificar que informações poderias ser obtidas através dos dados coletados.

Em seguida foi realizada a preparação dos dados. Este processo começou com a seleção das perguntas relacionadas com o uso de expressões lambda em Java 8 e suas respostas para que estas formem a base de dados a ser analisada. A seleção destas perguntas foi feita através das tags presentes nelas, sendo que as perguntas selecionadas foram todas as que continham as tags *lambda* e *java-8*. Essa base de dados foi organizada em diversos arquivos *csv* que armazenam informações diferentes a respeito das postagens selecionadas. Todo o processamento feito nos dados foi realizado através de *scripts Python*.

Ao todo, foram selecionadas 1975 perguntas e 3974 respostas para formar a base de dados do estudo. Após a formação dessa base de dados, foi realizada a modelagem desses dados a fim de responder a cada uma das três questões de pesquisa.

Para responder à primeira questão de pesquisa, foram usados *scripts R* para analisar alguns atributos quantitativos e qualitativos presentes nas postagens da base de dados, como as datas de criação das postagens, as pontuações dadas às postagens, a quantidade de respostas das perguntas etc. As informações obtidas a partir disso foram organizadas em gráficos e tabelas para que elas possam ser analisadas.

Para responder à segunda questão de pesquisa foi a realizada a análise de sentimentos no corpo textual das postagens da base de dados. Esta análise foi feita de forma automática através de um *script Python* que utiliza a ferramenta de análise de sentimentos VADER em conjunto com o Natural Language Toolkit (NLTK). O resultado desta análise foi organizado em curvas de densidade e tabelas, através de *scripts R*, para que estes possam ser analisados.

Para responder à terceira questão de pesquisa foram analisadas as tags das perguntas, com a finalidade de identificar os tópicos mais presentes as perguntas selecionadas. Isso, porém, não foi suficiente para responder à questão de pesquisa satisfatoriamente. Então foi decido complementar essas informações utilizando a leitura de um "top 100"das perguntas da base de dados. O critério utilizado para escolher este top 100 foi a popularidade das perguntas. Estas perguntas foram lidas para que fosse possível confirmar quais são os tópicos mais populares dentre as perguntas relacionadas com o uso de expressões lambda em Java 8.

A seleção das 100 perguntas mais populares foi baseada principalmente na quantidade de visualizações das perguntas, mas também levando em consideração a pontuação dada

a elas e a quantidade de vezes que as perguntas foram favoritadas. A escolha destes critérios foi feita pois, como explicado por Nadi et al.[15], o número de visualizações está diretamente relacionado com a procura por soluções de problemas com o tema em questão. Além disso, essa contagem é feita automaticamente para todos os acessos feitos às perguntas. Assim, esse é o principal indicador da popularidade desse tema. A pontuação indica a qualidade da pergunta, entretanto, só pode ser dada por usuários cadastrados, e nem todos o fazem. Por isso, este é apenas o indicador secundário da popularidade das perguntas. A contagem de favoritos é ainda menos utilizada que a pontuação, já que só é utilizada quando usuários querem receber notificações sobre atualizações nas perguntas. Por isso este é o menor indicador da popularidade das perguntas, dentre os três utilizados.

Além de encontrar os tópicos mais populares, as 100 perguntas lidas também foram utilizadas para complementar os resultados da análise de sentimentos. Para isso, primeiro foi repetida a mesma análise de sentimentos automática que já tinha sido feita em todas as perguntas nessas 100 perguntas. Segundo, foi utilizada a leitura dessas perguntas para identificar de uma forma mais interpretativa os sentimentos expressados nelas, a fim de se obter não só as polaridades mas também exemplos de sentimentos expressos nessas perguntas.

3.5 Ferramentas Utilizadas

Para realizar os procedimentos descritos acima, as seguintes ferramentas foram utilizadas:

- O NLTK é uma plataforma de código aberto voltada para o desenvolvimento de programas *Python* que trabalham com processamento de linguagem natural. Esta ferramenta foi utilizada para processar o texto das perguntas a fim de realizar a análise de sentimentos.
- VADER (um acrônimo para Valence Aware Dictionary for sEntiment Reasoning) foi a ferramenta utilizada para realizar a análise de sentimentos em si. Esta ferramenta faz a análise baseada em seu próprio dicionário léxico combinado com um conjunto de regras que englobam convenções gramaticais e sintáticas utilizadas para expressar ou enfatizar a intensidade de sentimentos[23]. A ferramenta VADER já está incluída nas versões mais recentes do NLTK, para que ambos possam ser utilizados sem gerar problemas.

Capítulo 4

Resultados

Este capítulo apresenta os resultados obtidos e as análises realizadas sobre eles, organizados por questão de pesquisa.

4.1 Q1: Quais são as peculiaridades de perguntas relacionadas com o uso de expressões lambda em Java 8 no site *Stack Overflow*?

Para responder a esta pergunta foram analisados os atributos quantitativos e qualitativos das postagens que formam a base de dados do estudo. Essa análise foi dividida em duas etapas. A primeira refere-se à frequência de criação de perguntas relacionadas com expressões lambda e Java 8 ao longo do tempo. A segunda refere-se à comparação de dos atributos quantitativos entre as postagens da base de dados com e todas as outras postagens do site *Stack Overflow*.

4.1.1 Frequência de Criação das Perguntas ao Longo do Tempo

Essa análise foi feita com a intenção de verificar se o interesse dos usuários do site *Stack Overflow* em fazer perguntas sobre expressões lambda em Java 8 aumentou ou diminuiu com o passar do tempo. Para isso, a frequência de criação das postagens da base de dados foi organizada sob a forma de um histograma ilustrado pela Figura 4.1.

A partir da observação desse histograma, em particular a altura das barras referentes à criação de perguntas relacionadas com expressões lambda em Java 8, nota-se que, entre 2014 e 2016 a frequência de criação das perguntas da base de dados analisada oscilou consideravelmente. Depois de 2016, entretanto, nota-se que a altura das barras referentes à frequência das perguntas ficou frequência das perguntas se manteve estável.

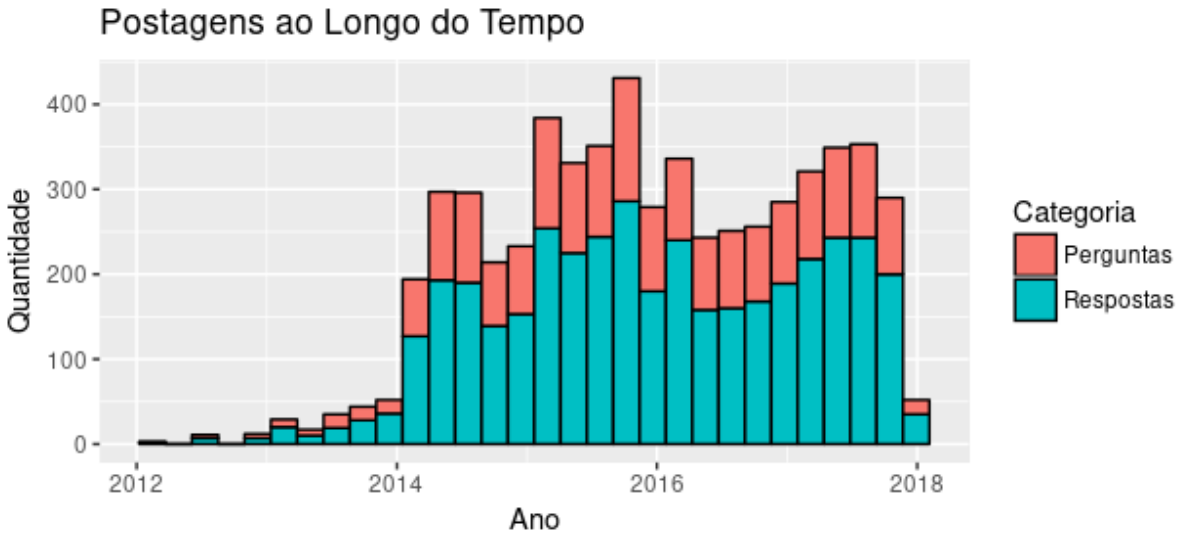

Figura 4.1: Frequência de criação de perguntas relacionadas com o uso de expressões lambda em Java 8, e de suas respostas, ao longo do tempo.

Assim como a frequência de criação de perguntas, a frequência de criação de respostas também oscilou bastante no intervalo de tempo entre 2014 e 1015. Contudo, a frequência de criação de respostas não se estabilizou após o ano de 2016, diferente do que aconteceu com as perguntas. Porém, não houveram reduções significativas nas alturas das barras referentes à criação de respostas. Pelo contrário, na maior parte do período no qual a frequência de perguntas se manteve estável, a frequência de respostas aumentou.

Outra coisa a se notar é que as barras referentes à criação das respostas são sempre maiores que as barras de criação das perguntas. Não só isso, também pode ser visto que as alturas das barras que representam a criação de respostas costumam ser por volta de duas vezes maiores que as alturas das barras referentes à criação de perguntas e, em alguns intervalos de tempo, a diferença entre a altura das barras é ainda maior. Isso indica que, em média, são criadas pelo menos duas questões para cada pergunta. Outro fator que reforça essa ideia é que os períodos nos quais as maiores barras de ambas as categorias aparecem coincidem.

Também é possível ver que a altura da última barra da direita é muito menor que todas as outras. Isso, porém, ocorre devido ao fato de que estes dados foram coletados em Dezembro de 2017, enquanto a faixa de tempo representada pela barra em questão começa ainda em 2017 mas vai até depois de 2018. Assim parte do período representado por esta barra está fora do escopo de dados coletados.

O que mais chama atenção na Figura 4.1, porém, é o fato de existirem perguntas com a tag *java-8* feitas antes da data de lançamento do Java 8. Ao todo, existem 108 perguntas feitas antes desta data. Isso é equivalente a apenas 5.47% das perguntas da

base de dados do estudo, porém algumas destas perguntas foram criadas no começo de 2012, pouco mais de dois anos antes do lançamento do Java 8.

Para explicar a existência dessas perguntas é importante saber que a data de lançamento do JDK 8, 18 de março de 2014, marca o dia em que Java 8 foi lançada para o público geral. A *Oracle* possibilita, através do *OpenJDK*, acesso a projetos da plataforma Java para desenvolvimento colaborativo[24]. Dentre os projetos disponibilizados está o *Project Lambda* que foi a força motriz por trás da criação do JDK 8[2]. Este projeto produziu o as implementações da *OpenJDK* relacionadas com adição de *closures* e características relacionadas na linguajem Java[25]. A pergunta mais antiga da base de dados coletadas para esse estudo é a pergunta: "*Are there any reasons why specifying the argument type is required in Java 8 lambda syntax?*"[1], feita em janeiro de 2012. Esta pergunta refere-se a uma dúvida acerca dos exemplos de sintaxe das expressões lambda contidos em um documento do *Project Lambda*[26].

Conclusões

A partir desta análise pode-se concluir, primeiro, que houve interesse em fazer perguntas sobre o uso de expressões lambida em Java 8 desde que as mesmas ainda estavam sendo implementadas. Este interesse, embora pequeno, foi aumentando a medida a data de lançamento do JDK 8 foi se aproximando.

Segundo, pode-se concluir que, após o lançamento oficial do JDK 8, o interesse em fazer perguntas sobre o uso de expressões lambda aumentou bastante. Em particular, entre os anos de 2014 e 2016 houveram alguns períodos nos quais foram feitas mais perguntas que os outros períodos. Depois de 2016 a frequência de criação de perguntas se estabilizou.

Por fim, pode-se concluir que a frequência de criação de respostas não acompanhou a variação do crescimento da frequência de criação das perguntas. Entretanto, isso se deve ao fato de que a frequência de criação de respostas aumentou mais do que a frequência de criação de perguntas. Não só isso, a taxa de respostas por perguntas se manteve em pelo menos duas respostas por pergunta na maioria dos períodos.

4.1.2 Comparando as Postagens da Base de Dados Com as Demais Postagens do *Stack Overflow*

Para poder comparar as postagens da base de dados com as demais perguntas do *Stack Overflow*, foi utilizado as médias dos valores dos atributos quantitativos das postagens de cada um destes grupos. Estes valores foram organizados em tabelas pertinentes às

[1] https://stackoverflow.com/questions/8907505

Tabela 4.1: Médias dos atributos quantitativos das perguntas.

Perguntas	Pontuação	Visualizações	Respostas	Comentários	Favoritos
Base de Dados	7.376	3552	2.012	3.016	2.035
Stack Overflow	1.898	1982	1.56	1.942	0.589

Tabela 4.2: Médias dos atributos quantitativos das respostas.

Respostas	Pontuação	Comentários
Base de Dados	6.921	1.832
Stack Overflow	2.648	1.421

perguntas e as respostas. Além disso, também foi calculada a taxa de satisfação das perguntas, baseando-se nas quantidades de respostas e de respostas aceitas.

Análise das Médias dos Atributos

Pela Tabela 4.1 pode-se ver que, em média, a pontuação atribuída às perguntas da base de dados é 3.89 vezes maior que a média do resto do site. A média de visualizações é 1.78 vezes maior que a média das outras perguntas. A média de respostas é 1.29 vezes maior que a média das outras perguntas. A média de comentários é 1.55 vezes maior que a média das outras perguntas. Por fim, a média de perguntas marcadas como favoritas é 3.46 vezes maior que a média das outras perguntas.

As respostas do site possuem atribuição de valores apenas para a pontuação e o número de comentários. Por isso a Tabela 4.2 mostra penas que, em média, a pontuação atribuída às respostas da base de dados é 2.61 vezes maior que a média do resto do site. E que média de comentários é 1.29 vezes maior que a média das outras perguntas.

Com estes dados é visto que tanto as perguntas quanto as respostas da base de dados estão acima da média do site, não importando qual atributo é considerado. Em especial estão os atributos de pontuação das perguntas e das respostas, e o número de favoritos das perguntas são bem maiores que a média do site. Estas pontuações indicam que a qualidade das perguntas relacionadas com o uso de expressões lambda e suas respostas é alta. O número de favoritos indica que usuários do site têm bastante interesse em acompanhar o desdobramento dessas perguntas. Vale ressaltar que estes atributos só são modificados por ações tomadas por usuários cadastrados no site. Portanto, estas perguntas aparentam ser de interesse especialmente para estes usuários.

Análise da Taxa de Satisfação das Perguntas

Para comparar a taxa de satisfação das perguntas da nase de dados com as demais perguntas do site, ambas foram classificadas de acordo com os seguintes critérios:

Tabela 4.3: Taxa de satisfação das perguntas.

Perguntas	Satisfeitas	Insatisfeitas	Ignoradas
Base de Dados	74.94%	19.90%	5.16%
Stack Overflow	53.57%	33.29%	13.14%

Perguntas Satisfeitas: são as perguntas que possuem um resposta marcada como resposta aceita.

Perguntas Insatisfeitas: são as perguntas que possuem respostas, mas não possuem nenhuma resposta aceita.

Perguntas Ignoradas: são as perguntas que não possuem nenhuma resposta.

Com as perguntas de cada um dos dois grupos a serem comparados classificadas em suas respectivas categorias, foi calculada a porcentagem de perguntas que fazem parte de cada uma destas categorias. Esses resultados estão registrados na Tabela 4.3. Esta tabela mostra que a taxa de perguntas satisfeitas da base de dados é consideravelmente maior que a taxa das outras perguntas do site. A diferença entre essas taxas é de 21.37%. Isso é um bom indicativo para as perguntas da base de dados, que mostra que essas perguntas costumam ser solucionadas com sucesso.

A taxa de perguntas insatisfeitas, entretanto, é menor na base de dados. A diferença entre os dois grupos para esta taxa é de 13.19%. Isso faz sentido devido ao fato de que quando um autor marca uma resposta dada a sua pergunta como aceita, essa pergunta passa a ser uma pergunta satisfeita. Isso significa que quanto mais a taxa de perguntas satisfeitas cresce, menor fica a taxa de perguntas insatisfeitas.

Por fim, taxa de perguntas ignoradas da base de dados também é menor que a taxa das outras perguntas. A diferença entre elas é de 7.98%. Isso tudo indica que perguntas sobre o uso de expressões lambda em Java 8 costumam ser, não só respondidas, mas respondidas com sucesso.

Conclusões

Com estes dados foi concluído que as perguntas sobre o uso de expressões lambda em Java 8 são bem vistas entre os usuários do *Stack Overflow*. Isso pôde ser concluído pois as médias de pontuação dada para as perguntas da base de dados e para as suas respostas estão bem acima da média calculada para as demais perguntas e respostas do site. Também foi possível concluir que essas perguntas são consideravelmente populares, em especial para usuários cadastrados no site. Isso pôde ser visto não só devido às altas médias de visualizações e de favoritos, mas também devido à alta taxa de perguntas que

são respondidas satisfatoriamente. Isso também indica que membros do site sabem como resolver perguntas com esse tema.

4.2 Q2: Quais são os sentimentos expressados em perguntas e respostas relacionadas com o uso de expressões lambda em Java 8?

Para responder a esta questão de pesquisa foram utilizados os resultados das análises de sentimentos realizada pela ferramenta VADER. Esses resultados são dados na forma de quatro valores atribuídos para cada postagem analisada. Deses valores, três correspondem à intensidade de cada polaridade de sentimento (positivo, neutro e negativo) expressada em uma postagem. Esses três valores são números reais que vão de 0 até 1,sendo que a soma deles é sempre 1. Desta forma, eles podem ser usados para representar a proporção de sentimentos de cada polaridade presentes na postagem.

O quarto valor, chamada pela ferramenta de "composto", corresponde ao veredicto dado para o sentimento expresso na postagem como um todo. Ele consiste de um número real, que vai de -1 a 1, calculado a partir de todos os valores obtidos na análise de uma postagem. Esse valor representa tanto a polaridade quanto a intensidade do sentimento, sendo que quanto mais próximo de -1 mais negativo é a postagem e quanto mais próximo de 1 mais positiva é a postagem.

A análise de sentimentos foi realizada separadamente para as perguntas e para as respostas. Além disso, a mesma análise de sentimentos foi repetida para analisar 100 perguntas mais populares dentre as perguntas da base de dados e em suas respostas. Essa segunda análise foi complementada com a leitura destas 100 perguntas.

4.2.1 Análise de Sentimentos da Base de Dados

Os resultados obtidos pela análise de sentimentos realizada nas postagens da base de dados foram utilizados para gerar as curvas de densidade presentes nas Figuras 4.2 a 4.4. Além disso, também foi obtido os dados de máximos, mínimos, médias e medianas dos valores obtidos. Esses dados estão organizados nas Tabelas 4.4 a 4.5.

Pela Figura 4.2 é visto que todas as perguntas apresentam intensidades de sentimentos com polaridade neutra muito maiores que sentimentos com polaridades positiva ou negativa. Isso mostra que as perguntas expressam poucos sentimentos, independente da polaridade. Contudo, vendo apenas as curvas de sentimentos positivos e negativos, percebe-se que quando sentimentos são encontrados nas perguntas, estes costumam ser

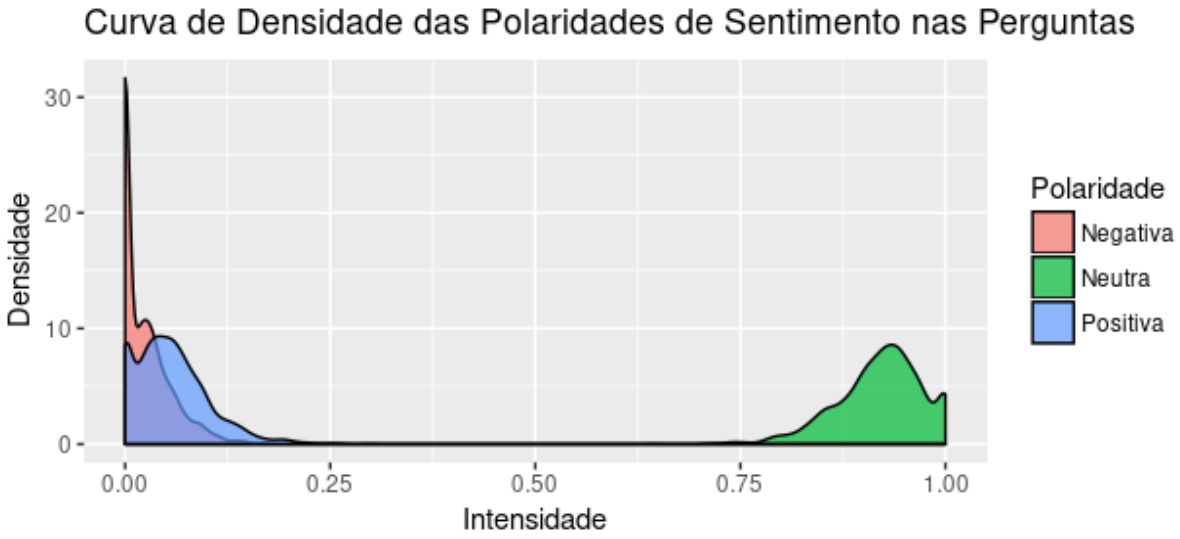

Figura 4.2: Curvas de densidade de cada polaridade de sentimento nas perguntas. Quanto mais a direita, maior é a intensidade.

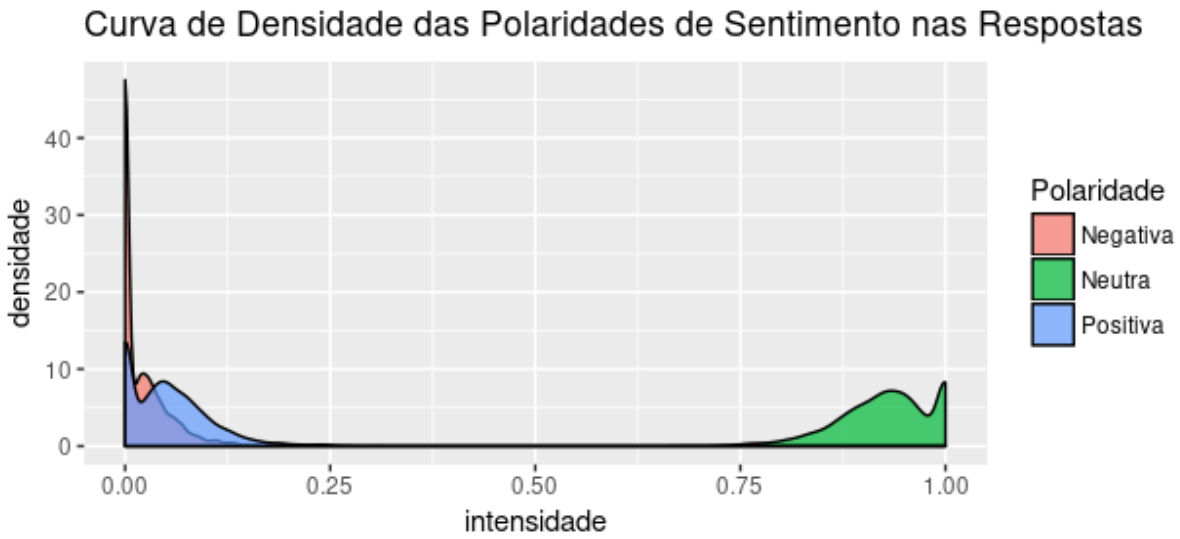

Figura 4.3: Curva de densidade de cada polaridade de sentimento nas respostas. Quanto mais a direita, maior é a intensidade.

positivos. A Figura 4.3 mostra que o mesmo ocorre com as respostas. A Figura 4.4 confirma que sentimentos negativos são os menos expressos, tanto nas perguntas como nas respostas. Esta figura também confirma que a maioria das respostas não expressam sentimentos com polaridade bem definida. Esta figura também mostra, entretanto, que existem mais perguntas com sentimentos positivos.

Os valores apresentados nas tabelas apenas confirmam o que já tinha sido visto nas curvas de densidade. Os máximos, médias e medianas das polaridades positiva e negativa

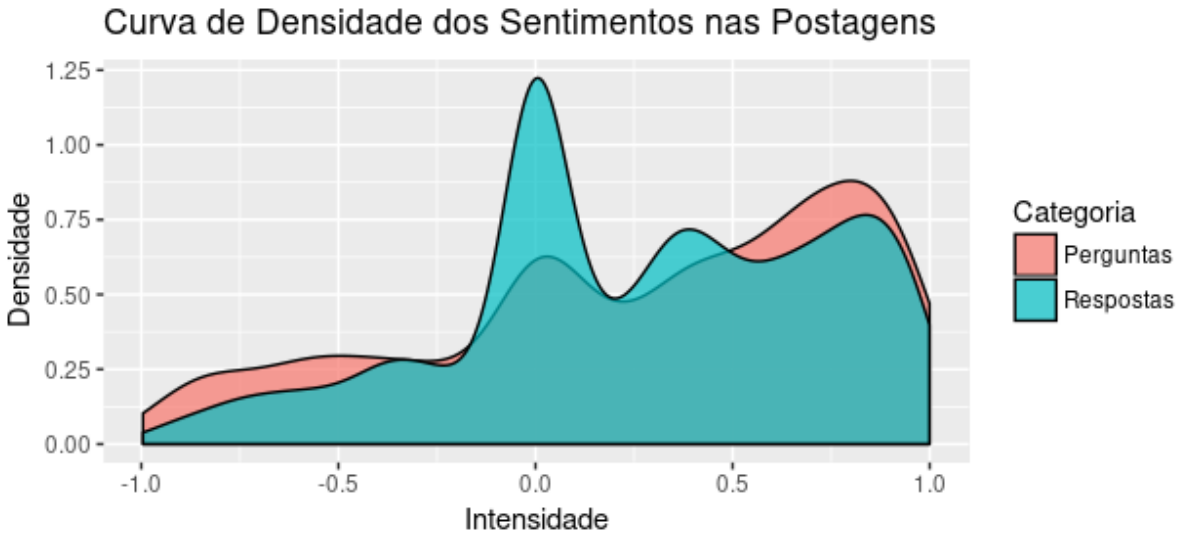

Figura 4.4: Curva de densidade da intensidade dos sentimentos nas perguntas e respostas. Quanto mais a direita, mais positivo é o sentimento. Quanto mais a esquerda, mais negativo.

Tabela 4.4: Dados sobre os sentimentos nas perguntas. Os valores Positivo, Neutro e Negativo variam de 0 a 1. O valor Composto varia de -1 a 1.

	Positivo	Neutro	Negativo	Composto
Máximos	0.2980	1.0000	0.1730	0.9976
Médias	0.0552	0.9209	0.0239	0.2671
Medianas	0.0500	0.9260	0.0150	0.3736
Mínimos	0.0000	0.6440	0.0000	-0.9921

foram muito baixos, enquanto o mínimo, a média e a mediana da polaridade neutra foram bem elevados. Estes valores também mostram que o mínimo da polaridade negativa nas perguntas é consideravelmente menor que nas respostas, o que pode ter influenciado a densidade de perguntas positivas vista na Figura 4.4.

Conclusões

A partir destes dados foi concluído que as postagens do site *Stack Overflow* costumam expressar poucos sentimentos, independente da polaridade. Isso pode ser resultado do caráter mais técnico das perguntas e das respostas presente neste site. Outro ponto importante de se notar é que, de acordo com Tourani et al.[11], ferramentas de análise de sentimentos automáticas apresentam precisão reduzida ao analisar textos referentes a desenvolvimento de softwares. Isso ocorre devido à presença de termos técnicos ambíguos e a dificuldade de distinguir textos neutros de textos positivos ou negativos. Isso reduz a credibilidade da análise realizada.

Tabela 4.5: Dados sobre os sentimentos nas respostas. Os valores Positivo, Neutro e Negativo variam de 0 a 1. O valor Composto varia de -1 a 1.

	Positivo	Neutro	Negativo	Composto
Máximo	0.3060	1.0000	0.2720	0.9996
Média	0.0511	0.9281	0.0207	0.2684
Mediana	0.0450	0.9330	0.0000	0.3058
Mínimo	0.0000	0.5980	0.0000	-0.9957

4.2.2 Análise de Sentimentos das 100 Perguntas Mais Populares

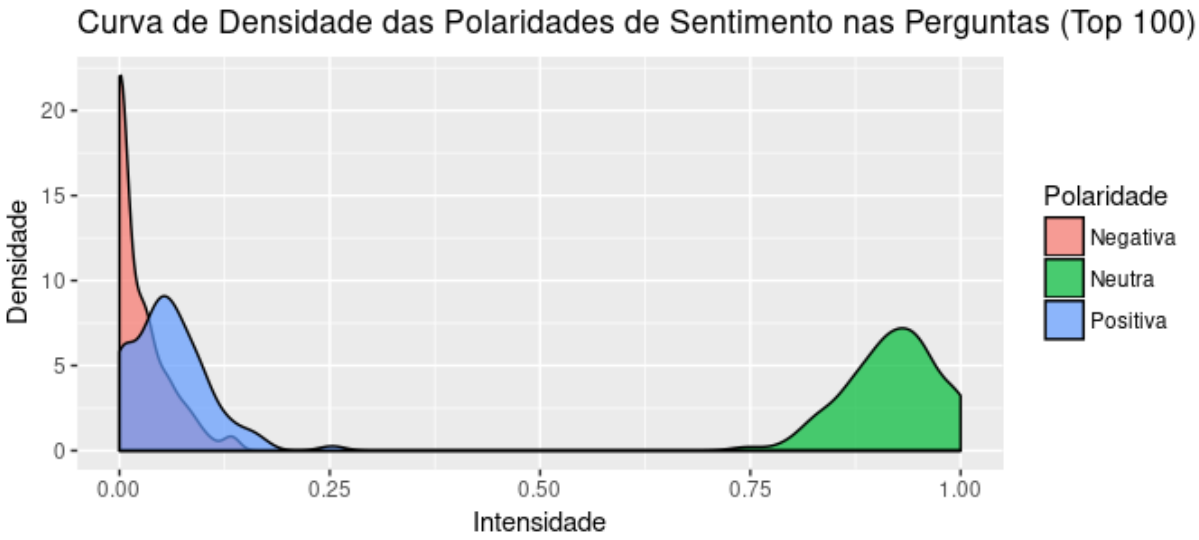

Figura 4.5: Curvas de densidade de cada polaridade de sentimento nas 100 perguntas mais populares. Quanto mais a direita, maior é a intensidade.

O resultado das análises de sentimentos automáticas feitas nas 100 perguntas mais populares e suas respostas foram usados para gerar as curvas de densidade das Figuras 4.5 a 4.7. Os valores de máximos, médias, medianas e mínimos foram utilizados para formar as Tabelas 4.6 a 4.7.

A Figura 4.5 e a Figura 4.6 apresentam resultados semelhantes aos resultados da análise realizada nas perguntas de toda a base de dados. Confirmando o viés neutro das postagens. A Figura 4.7 também confirma a predominância de respostas neutras. Entretanto, esta figura apresenta uma densidade ainda maior de postagens com sentimentos positivos para ambas as polaridades. Isso pode ser explicado observando os valores das tabelas. Os máximos da polaridade negativa em ambas as tabelas foram bem menores que os máximos obtidos para as mesmas polaridades em todas as postagens da base de dados. Outros valores de máximos das polaridades positivas também foram um pouco menor, mas a diferença entre os valores das polaridades negativas foi maior.

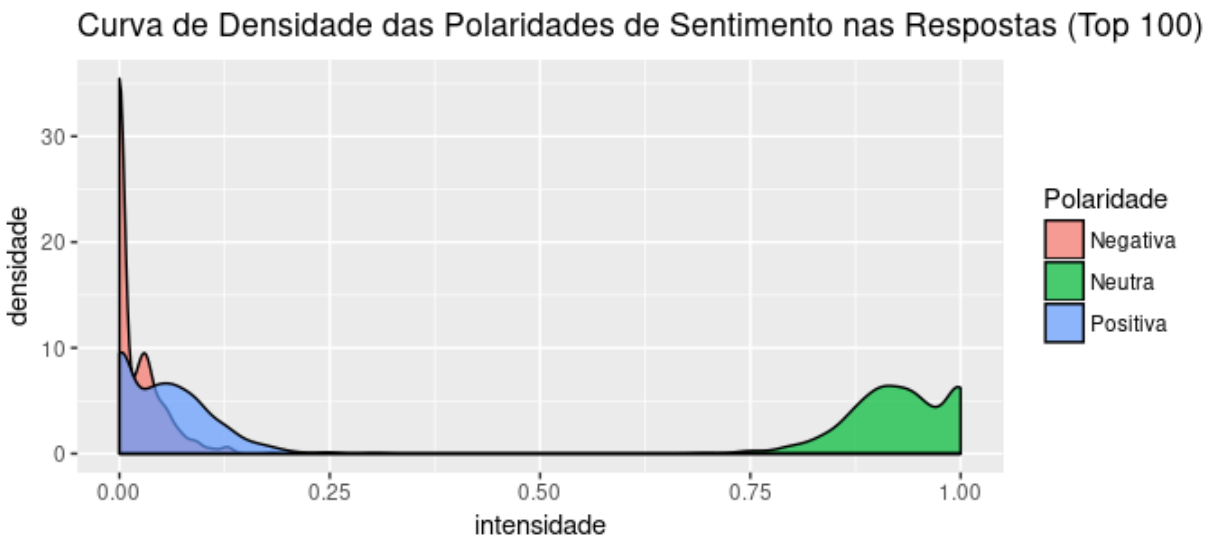

Figura 4.6: Curva de densidade de cada polaridade de sentimento nas respostas das 100 perguntas mais populares. Quanto mais a direita, maior é a intensidade.

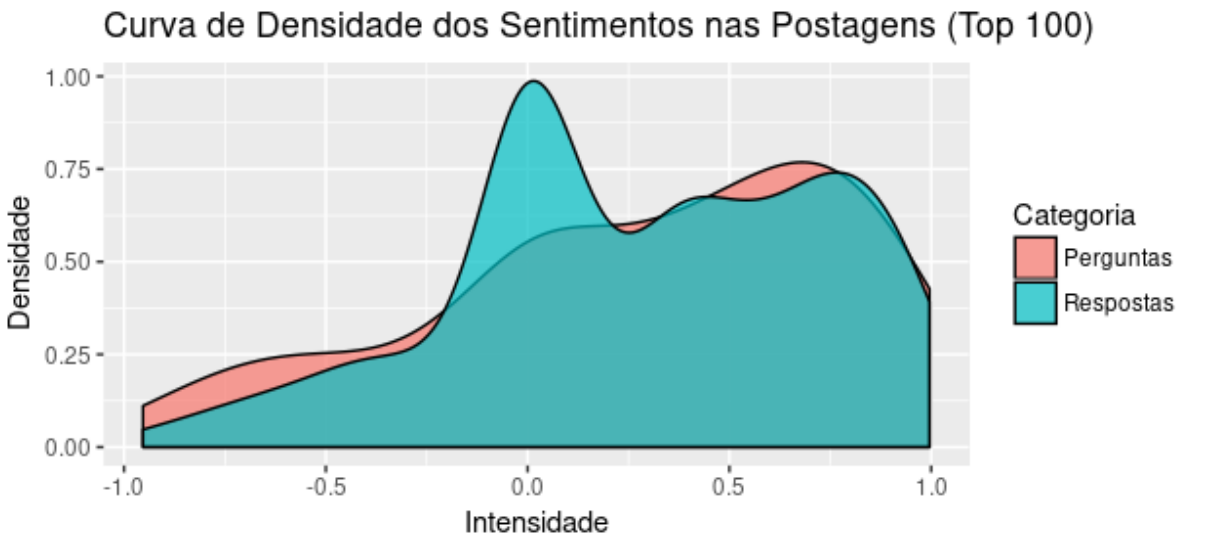

Figura 4.7: Curva de densidade da intensidade dos sentimentos nas 100 perguntas mais populares e suas respostas. Quanto mais a direita, mais positivo é o sentimento. Quanto mais a esquerda, mais negativo.

A partir da leitura das 100 melhores perguntas foi visto que as postagens são bem mais objetivas que subjetivas. Todas as perguntas lidas apresentam conteúdo bem técnico e focam na descrição de uma dúvida ou de um problema. As respostas, mais ainda, são bem diretas e visam sanar essas dúvidas ou resolver esses problemas. Isso restringe a expressão de sentimentos nas postagens do site.

O único sentimento que pôde ser identificados em algumas perguntas é a sensação de

Tabela 4.6: Dados sobre os sentimentos nas 100 perguntas mais populares. Os valores Positivo, Neutro e Negativo variam de 0 a 1. O valor Composto varia de -1 a 1.

	Positivo	Neutro	Negativo	Composto
Máximo	0.2530	1.0000	0.1330	0.9866
Média	0.0584	0.9183	0.0232	0.2780
Mediana	0.0560	0.9225	0.0040	0.3728
Mínimo	0.0000	0.7470	0.0000	-0.9531

Tabela 4.7: Dados sobre os sentimentos nas respostas das 100 perguntas mais populares. Os valores Positivo, Neutro e Negativo variam de 0 a 1. O valor Composto varia de -1 a 1.

	Positivo	Neutro	Negativo	Composto
Máximo	0.3040	1.0000	0.1820	0.9960
Média	0.0545	0.9262	0.0192	0.2832
Mediana	0.0480	0.9280	0.0000	0.3291
Mínimo	0.0000	0.6960	0.0000	-0.9524

frustração. Contudo, essa frustração foi detectada através da interpretação da situação em que os autores das dessas perguntas se encontravam, isto é, o fato de eles não estarem conseguindo resolver os problemas de seus programas. Logo, detectar este sentimento não é possível apenas analisando a composição textual da pergunta, é necessário compreender o contexto destas perguntas através de nuances deixadas pelo autor. Este resultado, porém, é obtido a partir de uma análise subjetiva, por isso ele está sujeito a ser inexato.

A pergunta que deixa esse sentimento de frustração mais aparente é a pergunta: "*Java 8: Mandatory checked exceptions handling in lambda expressions. Why mandatory, not optional?*"[2]. Nessa pergunta o autor explica que a empresa na qual ele trabalha iria migrar para o Java 8 e que, embora ele achasse as expressões lambda bem úteis, ele encontrou um problema ao tentar substituir uma interface bastante utilizada na empresa que aparenta não ter solução. Além disso, o autor da pergunta marcou uma resposta como aceita mas deixou um comentário nessa resposta indicando que, apesar de a solução apresentada funcionar, ele ainda não está totalmente satisfeito com a situação.

Outro exemplo desse sentimento está na pergunta "*Lambda Expression and generic method*"[3]. Nela o autor informa que ele já tentou resolver o problema apresentado de diversas maneiras diferentes e já tinha pesquisado na internet sobre possíveis soluções mas ainda assim não conseguiu resolver seu problema. No final do texto, ao invés de perguntar como resolver o problema, o autor pergunta se é possível resolvê-lo. Essa pergunta foi respondida com uma explicação do porquê não é possível utilizar expressões lambda do jeito que o autor da pergunta estava tentando utilizar.

[2]https://stackoverflow.com/questions/14039995
[3]https://stackoverflow.com/questions/22588518

Conclusões

A partir desta análise concluiu-se que as perguntas lidas são muito objetivas. Isso condiz com os resultados apresentados nas curvas de densidade de polaridades de sentimentos. Mas entra em conflito com a alta densidade de sentimentos positivos mostrada Figura 4.7. Uma possível causa desse viés mais objetivo são as políticas do próprio *Stack Overflow*, que estimulam a criação de perguntas mais práticas e objetivas, e desestimulam a criação de postagens opinativas.

Ainda assim foi possível notar que alguns autores de perguntas estavam se sentindo frustrados por não conseguirem resolver os problemas encontrados durante o desenvolvimento de seus programas. Isso, porém, foi inferido a partir da interpretação do contexto da criação das perguntas, o que deixa esta observação com um teor mais subjetivo e impreciso.

4.3 Q3: Quais são os principais tópicos presentes em perguntas relacionadas com o uso de expressões lambda em Java 8?

Para responder a esta questão foram realizadas duas tarefas. A primeira foi a listagem das tags mais recorrentes nas perguntas da base de dados da pesquisa. A segunda foi a leitura das 100 perguntas mais populares dentre as perguntas da base de dados coletada.

4.3.1 Análise das 25 Tags Mais Recorrentes nas Perguntas

Para inferir quais são os principais tópicos das peguntas relacionadas com expressões lambda e Java 8, foi feita a relação das 25 tags mais recorrentes nas perguntas da base de dados coletada. Nesta relação foram desconsideradas as tags *lambda*, *java-8*, pois estas estão presentes em todas as perguntas em questão. A Tabela 4.8 mostra essa relação das 25 tags mais recorrentes e o número de ocorrências de cada uma delas. A tabela também mostra a quantidade de perguntas possuem apenas as tags *lambda* e *java-8*.

A partir da Tabela 4.8, é possível ver que as 5 tags mais utilizadas nas perguntas da base de dados deste estudo foram:

1. A tag *java*, presente em 1642 perguntas (83.14% das perguntas). Isso, porém, não revela nenhuma informação nova sobre as perguntas, já que todas elas possuem a tag *java-8* e estão de alguma forma relacionadas com a linguagem Java.

Tabela 4.8: Número de ocorrências das tags mais recorrentes nas perguntas.

Tags	Base de Dados	Top 100
java	1642	95
java-stream	555	36
functional-programming	87	6
method-reference	82	2
collections	71	7
generics	62	1
list	49	3
functional-interface	41	2
optional	37	4
collectors	36	2
eclipse	35	2
foreach	28	4
predicate	27	2
intellij-idea	25	2
comparator	25	1
android	24	0
reflection	22	0
sorting	20	3
dictionary	20	2
filter	20	2
arrays	18	3
hashmap	18	0
serialization	16	1
closures	15	1
type-inference	14	0
Apenas *lambda* e *java-8*	68	0
Total	3057	181

2. A tag *java-stream*, presente em 555 perguntas (28.10% das perguntas). Esta tag é dada a perguntas relacionadas com o uso da API *Stream*, uma API adicionada ao Java 8 justamente para fazer uso das expressões lambda.

3. A tag *functional-programming*, presente em 87 perguntas (4.40% das perguntas). Esta tag é dada para perguntas relacionadas com programação funcional. O fato desta tag aparecer entre as tags com mais ocorrências nas perguntas relacionadas com expressões lambda não é uma surpresa, já que as estas foram adicionadas à linguagem java justamente para dar suporte à programação funcional.

4. A tag *method-reference*, presente em 82 perguntas (4.15% das perguntas). Esta tag é utilizada para marcar a pergunta com sendo relacionada com o uso de *Method references*, outra adição da versão 8 da linguagem Java para complementar o uso de expressões lambda.

5. A tag *collections* presente em 71 perguntas (3.59% das perguntas). Esta tag é dada para perguntas relacionadas com o uso da API *Collections*.

Conclusões

Com estes dados é possível concluir duas coisas. A primeira é que o tópico mais recorrente entre as perguntas do site *Stack Overflow* relacionadas com uso de expressões lambda em Java 8 é referente ao uso da API *Stream*. Mas isso só pode ser concluído devido ao elevado número de ocorrências da tag *java-stream*, que é mais de 6 vezes maior que o número de ocorrências da tag *functional-programming*. Não é possível concluir a popularidade de outros tópicos a partir das tags justamente pela segunda conclusão que pode-se tirar a partir desses dados: o uso das tags para identificar os tópicos de perguntas mais populares não apresentará um resultado muito preciso.

Dois fatores contribuem para esta conclusão. O primeiro é que existem perguntas cujas tags não representam todos os tópicos nelas abordados. Isso pode ser visto primeiramente pela tag *java*. Todas as perguntas da base de dados contêm a tag *java-8*, por tanto todas elas estão relacionadas com a linguagem Java. Entretanto, das 1975 perguntas utilizadas na pesquisa, somente 1642 contêm a tag *java*, deixando 333 perguntas sem esta tag. Outra observação que reforça essa ideia é que houveram 68 perguntas marcadas apenas com as tags *lambda* e *java-8*. Não só isso, houveram também 417 perguntas marcadas somente com as tags *lambda*, *java-8* e *java*. Isso dá, ao todo, 485 perguntas sem nenhuma indicação sobre um tópico específico além do fato de que elas são sobre o uso de expressões lambda em Java 8.

O segundo fator que contribui para a imprecisão das tags é que perguntas possuem múltiplas tags. Isso fica evidente devido ao total de perguntas listadas na tabela ser 3057,

um número que supera o total de perguntas da base de dados da pesquisa por 1082. Um bom exemplo disso é a tag *java-stream*, a mesma tag que representa o tópico mais recorrente nas perguntas. Esta tag aparece em 24 das 87 perguntas que contêm a tag *functional-programming*, em 3 das 87 perguntas que possuem a tag *method-reference* e em 40 das 71 perguntas que contêm a tag *collections*. Assim não é possível identificar se o principal tópico das perguntas que possuem a tag *collections*, por exemplo, é referente ao uso de *Collections* ou ao uso de *streams*.

4.3.2 Análise das 100 Perguntas Mais Populares

Para encontrar os tópicos mais populares das perguntas relacionadas com o uso de expressões lambda em Java 8 foram utilizadas as 100 perguntas mais populares selecionadas dentre as perguntas da base de dados da pesquisa. Com a leitura dessas perguntas e suas respostas foram definidos os tópicos principais de cada uma destas perguntas, sendo atribuído um único tópico para cada pergunta. Assim pôde-se categorizar as perguntas de acordo com o seu principal tópico.

Ao todo foi encontrado 9 tópicos diferentes, que são: perguntas sobre o uso de *streams*, perguntas conceituais, perguntas sobre refatoração de código, perguntas sobre casos específicos, perguntas sobre *arrays* ou *Collections*, perguntas sobre a sintaxe de expressões lambda, perguntas sobre tratamento de exceções, e perguntas sobre configurações de IDEs. A Tabela 4.9 mostra a relação destes tópicos, ordenada pelo número de perguntas referentes a cada um deles.

Para complementar esse resultado, também foram medidas as taxas de satisfação (Tabela 4.11) e as médias dos atributos quantitativos (Tabela 4.10) das perguntas de cada um destes tópicos. Além disso também foram conferidos os números de ocorrências das mesmas 25 tags vistas na seção anterior e o número de vezes que estas tags aparecem foram adicionados à Tabela 4.8 As descrições de cada uma destas 9 categorias são:

Perguntas sobre *Streams*

Este tópico foi dado a perguntas cujo principal tema era o uso da API *Stream*. Este foi o tópico mais encontrado dentre as 100 perguntas lidas, com 41 perguntas. O fato deste tópico ser o mais encontrado já era esperado, considerando que já foi visto através das tags. Além disso, é importante ressaltar que estas 41 perguntas não são as únicas perguntas relacionadas com o uso de *streams*, mas são as perguntas onde isso é o foco principal. A maioria destas perguntas envolvem operações com *Collections*.

Tabela 4.9: Relação de tópicos encontrados nas 100 perguntas mais populares e o número de perguntas e respostas para cada tópico.

Tópicos	Perguntas	Respostas
Streams	41	172
Conceituais	14	66
Refatoração	11	46
Casos Específicos	11	41
Arrays ou Collections	10	39
Sintaxe	6	29
Tratamento de Exceções	4	48
IDE	3	21
Total	100	462

Alguns exemplos de perguntas classificadas com este tópico são: *"How to map to multiple elements with Java 8 streams?"*[4] e *"Filter Java Stream to 1 and only 1 element"*[5], que remetem a dúvidas quanto ao uso adequado de métodos da API *Stream*.

Perguntas Conceituais

Este tópico foi dado a perguntas mais abertas, que não possuem um caso de uso específico em mente. Esta categoria contém 14 perguntas com um total de 66 respostas.

Alguns exemplos que podem ilustrar melhor as perguntas desta categoria são: *"Java8 Lambdas vs Anonymous classes"*[6], que pergunta quais são as vantagens e desvantagens entre uso de expressões lambda e o uso de classes anônimas, e *"Understanding Spliterator, Collector and Stream in Java 8"*[7], que pede para alguém explicar o que são um *Spliterator*, um *Collector* e a interface *Stream* e como utilizá-los.

Perguntas sobre Refatoração

Este tópico foi dado a perguntas que em alguma parte do seu conteúdo explicitamente mencionam a transição de um código que não utiliza expressões lambda para um que as utiliza. Foram selecionadas 11 perguntas que possuem 46 respostas nesta categoria. Vale destacar que este tópico possui as menores médias de visualizações, pontuação e favoritos por pergunta. É este tópico também que apresenta a menor taxa de satisfação, embora apenas 3 das 11 perguntas não possuam respostas aceitas. Outro fato interessante é que 6 destas perguntas envolvem o uso de *streams* para substituir laços de repetição.

[4]https://stackoverflow.com/questions/23620360
[5]https://stackoverflow.com/questions/22694884
[6]https://stackoverflow.com/questions/22637900
[7]https://stackoverflow.com/questions/19235606

Tabela 4.10: Médias dos atributos quantitativos das 100 perguntas mais populares, separadas por tópico. Os valores destacados são os maiores e menores valores de cada coluna.

Tópicos	Pontuação	Visualizações	Respostas	Comentários	Favoritos
Streams	49.85	47038	4.195	**1.659**	9.537
Conceituais	71.07	40212	4.714	4.357	16.93
Refatoração	**31.36**	**30982**	4.182	2.636	**8.545**
Casos Específicos	51.0	31290	**3.727**	3.091	10.55
Arrays/Collections	57.40	46036	3.90	2.1	10.60
Sintaxe	77.33	50140	4.833	1.667	14.50
Trat. de Exceções	**153.50**	**84609**	**12.0**	**5.75**	**47.75**
IDE	56	38036	7	4.667	9.667
Todas	57.65	43903	4.62	2.6	12.51

Tabela 4.11: Taxa de satisfação nas 100 perguntas mais populares, por tópico.

Tópicos	Satisfeitas	Insatisfeitas
Streams	87.8%	12.2%
Conceituais	92.86%	7.14%
Refatoração	72.73%	27.27%
Casos Específicos	81.82%	18.18%
Arrays ou Collections	100%	0%
Sintaxe	100%	0%
Tratamento de Exceções	100%	0%
IDE	100%	0%
Todas	89%	11%

Alguns exemplos dessas perguntas são: *"Java 8, Lambda : replace Anonymous inner class by lambda"*[8], que pede ajuda ao tentar refatorar um código que utiliza classes anônimas para um código que utiliza expressões lambda, e *"How to iterate nested lists with lambda streams?"*[9], que pergunta como transformar um código que utiliza laços de repetição *for* encadeados para iterar listas em um código que faz o mesmo usando a API *Stream*.

Perguntas sobre Casos Específicos

Este tópico foi dado a perguntas são feitas com o objetivo de resolver um problema bem específico descrito pelo autor, não se encaixando em nenhuma das outras categorias. Esta categoria contém 11 perguntas com 41 respostas, apresentando a menor média de respostas por pergunta.

Alguns exemplos de perguntas com este tópico são: *"How to serialize a lambda?"*[10], na qual o autor pergunta como consertar um código que lança uma exceção, e *"Returning a value from a method within a lambda expression"*[11], na qual o autor apresenta seu código que não retorna o que ele quer e pergunta como fazer para retornar o valor correto.

Perguntas sobre *Arrays* ou *Collections*

Este tópico foi dado a perguntas cujo principal tema é a manipulação de *arrays* ou *Collections*. Um fato importante de ser destacado sobre estas perguntas é que embora as perguntas em si não fazem menção ao uso de *streams*, 7 das 10 perguntas nesta categoria marcaram como aceitas respostas que utilizam *streams* para solucionar o problema proposto.

Alguns exemplos de perguntas com este tópico são: *"java: Arrays.sort() with lambda expression"*[12], na qual o autor pede para alguém apresentar a forma correta de ordenar *arrays* visto que o código dele não funciona, e *"Lambda expression to convert array/List of String to array/List of Integers"*[13], que pergunta como converter um *array* ou uma lista de *strings* para um array ou uma lista de inteiros, *floats* ou *doubles*.

[8] https://stackoverflow.com/questions/25270467
[9] https://stackoverflow.com/questions/29215375
[10] https://stackoverflow.com/questions/22807912
[11] https://stackoverflow.com/questions/24704791
[12] https://stackoverflow.com/questions/21970719
[13] https://stackoverflow.com/questions/23057549

Perguntas sobre Sintaxe

Este tópico foi dado a perguntas cujo principal objetivo é resolver um problema relacionado com a sintaxe das expressões lambda. Foram colocadas apenas 6 perguntas nesta categoria, mas que obtiveram 29 respostas.

Alguns exemplos destas perguntas são: *"Java 8 lambda Void argument"*[14], que pede para alguém mostrar uma alternativa de como escrever uma expressão sem causar erros e *"How do you assign a lambda to a variable in Java 8?"*[15], que apresenta dois trechos de códigos e pergunta por que só o segundo acusa erros de sintaxe.

Perguntas sobre Tratamento de Exceções

Este tópico foi dado a perguntas relacionadas com o lançamento de exceções em expressões lambda. Um fato importante de ser notado é que este é a categoria com o segundo menor número de perguntas, com apenas 4, mas que possui o terceiro maior número de respostas, 48. Não só isso, as médias de todos os atributos quantitativos das perguntas com esse tópico são maiores que os das perguntas de qualquer outro tópico.

Alguns exemplos dessas perguntas são: *"How can I throw CHECKED exceptions from inside Java 8 streams?"*[16], que pergunta como fazer para poder lançar uma exceção que normalmente não é possível em interfaces funcionais em Java 8, e *"Java 8: Lambda-Streams, Filter by Method with Exception"*[17], que pergunta como fazer para poder tratar uma exceção lançada por métodos chamados por expressões lambda.

Perguntas sobre Configuração de IDEs

Este tópico foi dado a perguntas relacionadas com a configuração de IDEs para o uso de expressões lambda. Foram selecionadas apenas três perguntas para esta categoria, mas elas obtiveram um total de 21 respostas. Além disso elas apresentam a segunda maior média de respostas e comentários por pergunta, embora também apresentem a terceira pior média de visualizações. As três perguntas desta categoria são: *"Java 8 Lambdas don't work, everything else from Java 8 works though"*[18], *"Java "lambda expressions not supported at this language level""*[19] e *"Java 8: Formatting lambda with newlines and indentation"*[20]. As duas primeiras perguntas citadas referem-se a IDEs (Eclipse e Intellij IDEA) que apresentam mensagens de erro afirmando que não dão suporte ao uso de

[14]https://stackoverflow.com/questions/29945627
[15]https://stackoverflow.com/questions/21920039
[16]https://stackoverflow.com/questions/27644361
[17]https://stackoverflow.com/questions/19757300
[18]https://stackoverflow.com/questions/22544064
[19]https://stackoverflow.com/questions/22703412
[20]https://stackoverflow.com/questions/24649971

expressões lambda, as respostas aceitas destas mostram como configurar as IDEs corretamente para resolver estes problemas. A terceira refere-se a como configurar qualquer IDE para melhorar a indentação automática nas expressões lambda.

Conclusões

A partir destes dados é possível concluir que o principal tópico das perguntas relacionadas com o uso de expressões lambda em Java 8 é o uso da API *Stream*. Este tópico é o principal foco do maior grupo de perguntas dentre as 100 perguntas selecionadas e, assim como foi visto nas tags, este tópico é o único presente em uma quantidade muito maior que os outros. Além disso, *stream* também é um tópico presente em muitas outras perguntas cujos principais focos sejam outros. Em especial perguntas que envolvem iteração de *arrays* ou *Collections* foram em grande parte respondidas com a utilização de *streams*.

As perguntas sobre refatoração de código são outro grupo de perguntas que costuma envolver o uso de *Streams*. Ao contrário do que era esperado, apenas uma das perguntas lidas envolvia a refatoração direta de classes anônimas para expressões lambda. Por volta de metade destas perguntas, entretanto, envolvia a substituição de laços de repetição pelo uso da API *Stream*. Outro ponto que podemos inferir destes dados é que, dentre as perguntas mais populares, este é um dos tópicos menos popular. Isso não foi concluído a partir do número de perguntas realizadas, mas sim pela baixa visualização e baixo número de respostas e favoritos das perguntas com este tópico. Uma possível razão para isso é que, como muitas das refatorações desejadas serem referentes ao uso de *streams*, a busca por perguntas sobre *streams* pode ser o suficiente para resolver as dúvidas de muitos desenvolvedores.

Outra conclusão que pode ser feita a partir desses dados é que as perguntas sobre tratamento de exceções, mesmo que poucas, são bastante populares. Não só a média de todos os deste grupo de perguntas são maiores que as dos demais tópicos, mas salvo a média de comentários, essas médias são muito maiores que as segundas maiores médias. Em particular a elevada média de visualizações leva à conclusão de que as perguntas deste grupo são acessadas com frequência. E as elevadas médias de pontuação de favoritos levam à conclusão de que estas perguntas são especialmente populares entre membros cadastrados no site *Stack Overflow*, já que somente membros dos site podem alterar a pontuação das perguntas ou favoritá-las.

Capítulo 5

Considerações Finais

Este estudo permitiu a identificação de alguns aspectos interessantes a respeito de características de perguntas feitas sobre o uso de expressões lambda em Java 8. As perguntas selecionadas para serem utilizadas nesta pesquisa formam uma parcela bem pequena do site *Stack Overflow*, o que faz sentido já que o site existe desde 2008 e a adicção de expressões lambda na linguagem Java só ocorreu em 2014. Perguntas sobre esse tema, entretanto, foram encontradas desde 2012, durante o período no qual essa nova característica da linguagem ainda estava sendo implementada. Não obstante, perguntas sobre esse tema só começaram a ser feitas com bastante frequência depois do lançamento do Java 8. Além disso, a frequência na qual estas perguntas são feitas se mantem estável, principalmente depois de 2016.

Outra característica interessante de se notar sobre as perguntas selecionadas, é que ela está acima de média do site em qualquer um dos atributos quantitativos que estas possuem. Além disso, a taxa de perguntas com esse tema respondidas de forma satisfatória é bem maior que a mesma taxa para o resto do site. Tudo isso indica que essas perguntas (e, por extensão a temática de expressões lambda em Java 8) são populares entre os usuários do site *Stack Overflow* e que estes usuários costumam conseguir respondê-las corretamente e acompanhar o desenrolar dessas perguntas.

A análise de sentimentos mostrou principalmente o caráter mais objetivo das perguntas e de suas respostas, já que não só as curvas de densidade referente às polaridades acusaram um predomínio de polaridade neutra, mas também as perguntas lidas apresentavam um viés bastante objetivo e técnico. Isso, porém, pode não ocorrer somente para perguntas com a temática abordada neste trabalho, pois o site possui políticas que incentivam a postagem de perguntas objetivas e desencoraja a postagem de perguntas mais subjetivas ou opinativas.

Mais uma característica que foi notada neste estudo é que não existe muito interesse em perguntas que envolvam refatoração de classes anônimas para expressões lambda. Ao

invés disso, a maioria das perguntas foram feitas para resolver problemas quanto ao uso de *stream*, geralmente envolvendo operações com *Collections* ou *arrays*. Até mesmo as perguntas que envolviam refatoração de códigos antigos eram, em sua maioria, referentes à substituição de laços de repetição entrelaçados para a utilização da API *Stream*, o que permitia que o código ficasse mais legível. Dentre os outros tópicos encontrados nas perguntas lidas, o que mais se destaca refere-se a perguntas sobre tratamento de exceções. Contudo, esse tópico não se destaca pela quantidade de perguntas feitas sobre ele, mas pelo grande interesse demonstrado pelos usuários do *Stack Overflow* nas poucas perguntas feitas sobre esse tópico.

Para trabalhos e contribuições futuras, seria interessante o desenvolvimento de um classificador capaz de identificar os principais tópicos de todas as perguntas relacionadas com o uso de expressões lambda em java 8. Esse classificador pode utilizar uma fração das perguntas, não necessariamente as perguntas mais populares, classificadas manualmente como base para realizar essa classificação.Com isso seria possível entender melhor quais são os tópicos mais presentes nessas perguntas, ou também poderiam ser identificadas mais peculiaridades desses tópicos, por exemplo, se o interesse em um tópico específico varia com o passar do tempo.

Uma outra possibilidade para uma contribuição futura é a utilização de outras fontes de dados para tentar confirmar se os tópicos identificados neste trabalho são uma causa de problemas comuns durante o desenvolvimento de aplicações Java que utilizem expressões lambda. Isso seria interessante em especial para a questão de tratamento de exceções, que foi identificada como um tópico bem procurado e acompanhado pelos usuários do *stack Overflow*.

Referências

[1] Oracle: *Anonymous classes (the java^{TM} tutorials > learning the java language > classes and objects)*. Online. `https://docs.oracle.com/javase/tutorial/java/javaOO/anonymousclasses.html`, Visitado em 20 de junho de 2018. 1

[2] Oracle: *Jdk 8*. Online. `http://openjdk.java.net/projects/jdk8/`, Visitado em 6 de julho de 2018. 1, 20

[3] Oracle: *Lambda expressions (the java^{TM} tutorials > learning the java language > classes and objects)*. Online. `https://docs.oracle.com/javase/tutorial/java/javaOO/lambdaexpressions.html`, Visitado em 20 de junho de 2018. 1

[4] Oracle: *java.util.stream (java platform se 8)*. Online. `https://docs.oracle.com/javase/8/docs/api/java/util/stream/package-summary.html`, Visitado em 20 de junho de 2018. 1

[5] Oracle: *Method references (the java^{TM} tutorials > learning the java language > classes and objects)*. Online. `https://docs.oracle.com/javase/tutorial/java/javaOO/methodreferences.html`, Visitado em 20 de junho de 2018. 1

[6] Witten, Ian H., Frank Eibe, Mark A. Hall e Christopher Pal: *Data mining: practical machine learning tools and techniques*. Morgan Kaufmann Series in Data Management Systems. Morgan Kaufmann, 4ª edição, 2016, ISBN 9780128043578. 4, 5, 6, 7, 8

[7] Shearer, Colin: *The crisp-dm model: The new blueprint for data mining*. Journal of Data Warehousing, 5(4):13–22, 2000. 4

[8] Ahmed, Abeer Badr El Din e Ibrahim Sayed Elaraby: *Data mining: A prediction for student's performance using classification method*. World Journal of Computer Application and Technology, 2:43–47, 2014. 6

[9] Liu, Bing: *Sentiment analysis and subjectivity*. Em Indurkhya, Nitin e Fred J. Damerau (editores): *Handbook of Natural Language Processing, Second Edition.*, páginas 627–666. Chapman and Hall/CRC, 2010, ISBN 978-1-4200-8592-1. `http://www.crcnetbase.com/doi/abs/10.1201/9781420085938-c26`. 8

[10] Madhoushi, Z., A. R. Hamdan e S. Zainudin: *Sentiment analysis techniques in recent works*. Em *2015 Science and Information Conference (SAI)*, páginas 288–291, July 2015. 8, 9

[11] Tourani, Parastou, Yujuan Jiang e Bram Adams: *Monitoring sentiment in open source mailing lists: exploratory study on the apache ecosystem.* Em Ng, Joanna, Jin Li e Ken Wong (editores): *Proceedings of 24th Annual International Conference on Computer Science and Software Engineering, CASCON 2014, Markham, Ontario, Canada, 3-5 November, 2014*, páginas 34–44. IBM / ACM, 2014. http://dl.acm.org/citation.cfm?id=2735528. 8, 9, 25

[12] Hassan, Ahmed E.: *The road ahead for mining software repositories.* Em *2008 Frontiers of Software Maintenance*, páginas 48–57, Sept 2008. 9, 10

[13] Tavares, Aline Laís Gomes e Filipe Cardoso Caldas: *Caracterizando a adoção de expressões lambda em código java legado.* Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação), Universidade de Brasília, Brasília, 2017. http://bdm.unb.br/handle/10483/18496. 11

[14] Pinto, Gustavo, Fernando Castor e Yu David Liu: *Mining questions about software energy consumption.* Em Devanbu, Premkumar T., Sung Kim e Martin Pinzger (editores): *11th Working Conference on Mining Software Repositories, MSR 2014, Proceedings, May 31 - June 1, 2014, Hyderabad, India*, páginas 22–31. ACM, 2014, ISBN 978-1-4503-2863-0. http://doi.acm.org/10.1145/2597073.2597110. 11

[15] Nadi, Sarah, Stefan Krüger, Mira Mezini e Eric Bodden: *"jumping through hoops": Why do java developers struggle with cryptography apis?* Em Jürjens, Jan e Kurt Schneider (editores): *Software Engineering 2017, Fachtagung des GI-Fachbereichs Softwaretechnik, 21.-24. Februar 2017, Hannover, Deutschland*, volume P-267 de *LNI*, página 57. GI, 2017, ISBN 978-3-88579-661-9. https://dl.gi.de/20.500.12116/1268. 11, 17

[16] *Mining challenge - msr 2015.* Online. http://2013.msrconf.org/challenge.php, Visitado em 11 de julho de 2018. 13

[17] *Mining challenge - msr 2013.* Online. http://2015.msrconf.org/challenge.php, Visitado em 11 de julho de 2018. 13

[18] *MSR '13: Proceedings of the 10th Working Conference on Mining Software Repositories*, Piscataway, NJ, USA, 2013. IEEE Press, ISBN 978-1-4673-2936-1. 13

[19] *MSR '15: Proceedings of the 12th Working Conference on Mining Software Repositories*, Piscataway, NJ, USA, 2015. IEEE Press, ISBN 978-0-7695-5594-2. 13

[20] Stack Overflow: *Welcome to stack overflow.* Online. https://stackoverflow.com/tour, Visitado em 9 de julho de 2018. 14

[21] Stack Overflow: *What topics can i ask about here?* Online. https://stackoverflow.com/help/on-topic, Visitado em 9 de julho de 2018. 14

[22] David Fullerton: *Stack exchange creative commons data now hosted by the internet archive.* Online. https://stackoverflow.blog/2014/01/23/stack-exchange-cc-data-now-hosted-by-the-internet-archive/, Visitado em 10 de dezembro de 2017. 14

[23] Hutto, Clayton J. e Eric Gilbert: *VADER: A parsimonious rule-based model for sentiment analysis of social media text.* Em Adar, Eytan, Paul Resnick, Munmun De Choudhury, Bernie Hogan e Alice H. Oh (editores): *Proceedings of the Eighth International Conference on Weblogs and Social Media, ICWSM 2014, Ann Arbor, Michigan, USA, June 1-4, 2014*. The AAAI Press, 2014, ISBN 978-1-57735-659-2. http://www.aaai.org/ocs/index.php/ICWSM/ICWSM14/paper/view/8109. 17

[24] Oracle: *Openjdk.* Online. http://openjdk.java.net/, Visitado em 7 de julho de 2018. 20

[25] Oracle: *Project lambda.* Online. http://openjdk.java.net/projects/lambda/, Visitado em 7 de julho de 2018. 20

[26] Oracle: *State of the lambda v4.* Online. http://cr.openjdk.java.net/~briangoetz/lambda/lambda-state-4.html, Visitado em 7 de julho de 2018. 20

Anexo I

As 100 perguntas lidas

https://stackoverflow.com/questions/20363719
https://stackoverflow.com/questions/14830313
https://stackoverflow.com/questions/18198176
https://stackoverflow.com/questions/28607191
https://stackoverflow.com/questions/20938095
https://stackoverflow.com/questions/13604703
https://stackoverflow.com/questions/23308193
https://stackoverflow.com/questions/19254884
https://stackoverflow.com/questions/23057549
https://stackoverflow.com/questions/24328679
https://stackoverflow.com/questions/25147094
https://stackoverflow.com/questions/19757300
https://stackoverflow.com/questions/27644361
https://stackoverflow.com/questions/22694884
https://stackoverflow.com/questions/22703412
https://stackoverflow.com/questions/22725537
https://stackoverflow.com/questions/32264773
https://stackoverflow.com/questions/30425836
https://stackoverflow.com/questions/29945627
https://stackoverflow.com/questions/32884195
https://stackoverflow.com/questions/25186216
https://stackoverflow.com/questions/19235606
https://stackoverflow.com/questions/24054773
https://stackoverflow.com/questions/22742974
https://stackoverflow.com/questions/23407014
https://stackoverflow.com/questions/23620360
https://stackoverflow.com/questions/21233183
https://stackoverflow.com/questions/27872387

https://stackoverflow.com/questions/30012295
https://stackoverflow.com/questions/28818506
https://stackoverflow.com/questions/25270467
https://stackoverflow.com/questions/19278443
https://stackoverflow.com/questions/25055392
https://stackoverflow.com/questions/23004921
https://stackoverflow.com/questions/22588518
https://stackoverflow.com/questions/27677256
https://stackoverflow.com/questions/17640754
https://stackoverflow.com/questions/14039995
https://stackoverflow.com/questions/27870136
https://stackoverflow.com/questions/27228961
https://stackoverflow.com/questions/24228279
https://stackoverflow.com/questions/31130457
https://stackoverflow.com/questions/24112715
https://stackoverflow.com/questions/28032827
https://stackoverflow.com/questions/22637900
https://stackoverflow.com/questions/23701943
https://stackoverflow.com/questions/32335335
https://stackoverflow.com/questions/30330688
https://stackoverflow.com/questions/22917270
https://stackoverflow.com/questions/22933296
https://stackoverflow.com/questions/26422166
https://stackoverflow.com/questions/21833537
https://stackoverflow.com/questions/31683375
https://stackoverflow.com/questions/31251629
https://stackoverflow.com/questions/23261803
https://stackoverflow.com/questions/24541786
https://stackoverflow.com/questions/23860533
https://stackoverflow.com/questions/24553761
https://stackoverflow.com/questions/21912314
https://stackoverflow.com/questions/31629324
https://stackoverflow.com/questions/19676750
https://stackoverflow.com/questions/37726874
https://stackoverflow.com/questions/30608360
https://stackoverflow.com/questions/32262059
https://stackoverflow.com/questions/26304858
https://stackoverflow.com/questions/27969584
https://stackoverflow.com/questions/18400210
https://stackoverflow.com/questions/29215375

https://stackoverflow.com/questions/31763930
https://stackoverflow.com/questions/24378646
https://stackoverflow.com/questions/28790784
https://stackoverflow.com/questions/23213891
https://stackoverflow.com/questions/24396544
https://stackoverflow.com/questions/21970719
https://stackoverflow.com/questions/22544064
https://stackoverflow.com/questions/28989841
https://stackoverflow.com/questions/26179001
https://stackoverflow.com/questions/24649971
https://stackoverflow.com/questions/21920039
https://stackoverflow.com/questions/24779806
https://stackoverflow.com/questions/26340688
https://stackoverflow.com/questions/35928747
https://stackoverflow.com/questions/16859992
https://stackoverflow.com/questions/31456898
https://stackoverflow.com/questions/28508253
https://stackoverflow.com/questions/26771953
https://stackoverflow.com/questions/31693781
https://stackoverflow.com/questions/23324782
https://stackoverflow.com/questions/22807912
https://stackoverflow.com/questions/32521972
https://stackoverflow.com/questions/24436871
https://stackoverflow.com/questions/26549659
https://stackoverflow.com/questions/24704791
https://stackoverflow.com/questions/31044041
https://stackoverflow.com/questions/22598665
https://stackoverflow.com/questions/25712591
https://stackoverflow.com/questions/30095651
https://stackoverflow.com/questions/23682243
https://stackoverflow.com/questions/25439277
https://stackoverflow.com/questions/26568555