UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE INFORMÁTICA
GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Análise da Comunidade de Desenvolvedores Open Source

Karla Polyana Silva Falcão

RECIFE, DEZEMBRO DE 2018

Resumo

Este trabalho baseia-se na análise das redes complexas para fazer um exame quantitativo dos perfis sociais e técnicos dos desenvolvedores de software Open Source. A Análise de Redes Complexas visa analisar a estrutura das relações existentes no universo através da Teoria dos Grafos. Neste trabalho, aplicamos técnicas de Análise de Redes Complexas para construir a rede de social-coding do Github, onde além de gerenciar versão de código os desenvolvedores podem Criar, "Forkar" ou "Favoritar" repositórios de código, além de seguir atividades de outros desenvolvedores. Construímos um grafo de relacionamento entre desenvolvedores onde calculamos várias das suas características. Em seguida, identificamos desenvolvedores e tecnologias influentes nesta sub-rede do GitHub usando os algoritmos bem consolidados da área para detectar estrutura de comunidade. Compreender como os desenvolvedores e tecnologias são relacionados uns aos outros em um site de social-coding como o Github é essencial para construir ferramentas que ajudem o ecossistema de desenvolvimento Open Source.

Sumário

1 – Introdução 1
2 – Visão Geral 7
3 – Coleta e Análise Exploratória dos Dados 9
4 – Construção e Propriedades da Rede de Desenvolvedores 13
4.1 – Densidade e Reciprocidade 16
4.2 – Conectividade 19
4.3 – Efeito Mundo Pequeno 20
5 – A Comunidade de Desenvolvedores 25
5.1 – Detecção de Comunidades 26
5.2 – Análise das Comunidades 29
6 – Conclusão e Trabalhos Futuros 34
Referências Bibliográficas 36

1 – Introdução

Em redes sociais tais como LinkedIn ou Facebook a amizade é representada como links em um grafo não direcionado, já serviços como o Twitter e o Github são baseados numa rede direcionada. Uma abordagem de rede direcionada permite que os usuários sigam outros de acordo com seus interesses sem pedir reciprocidade de relacionamento. Em redes sociais tradicionais alguns usuários são seguidos por muitos, mesmo que estes, por sua vez, não façam o mesmo. Estes são conhecidos como "estrelas" ou "celebridades". Será que esse também é o caso de redes de colaboração de código como o Github? As pessoas no GitHub geralmente são seguidas porque elas trabalham em projetos interessantes.

Aqui é proposta uma pesquisa quantitativa para entender como funciona o ecossistema de social coding da cidade do Recife. Coletamos os usuários do GitHub, então, para cada um dos usuários, recuperamos seus projetos e as tecnologias associadas. Foram coletados aproximadamente 2000 desenvolvedores recifenses do GitHub. Primeiro, analisamos as informações dos usuários do Github calculando várias estatísticas.

Depois, construímos um grafo de relacionamentos do tipo "seguir" entre desenvolvedor-desenvolvedor e extraímos várias propriedades da rede complexa formada. A rede de seguidores do GitHub forma um grafo direcionado caracterizado por redes sociais como o Twitter, em que poucos têm muitos seguidores, considerados usuários "estrelas". Em seguida, identificamos desenvolvedores influentes nesta rede do GitHub usando o PageRank e Edge Betweeness.

Por fim, aplicamos alguns algoritmos renomados de clustering em redes para de detecção de comunidades para encontrar os grupos, e os caracterizamos de acordo com o perfil dos desenvolvedores. Uma visão arquitetônica da nossa abordagem é mostrada na Figura 1.

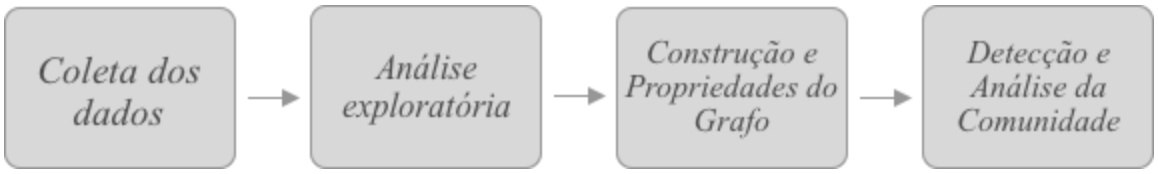

Figura 1. Visão arquitetônica da abordagem

2 – Visão Geral

A Análise de Redes Complexas envolve diversos campos não computacionais, e ainda abrange o seu estudo computacional/matemático. Suportada pela teoria dos grafos, propriedades relevantes podem ser extraídas, como por exemplo a propriedade chamada Mundo Pequeno, alusiva ao fenômeno "seis-degraus-de-separação", que é comum em alguns tipos de redes (S Milgram -The Small world problem).

2.1 – Redes Complexas

As Redes Complexas podem ser biológicas, sociais, tecnológicas, etc. e abordam as relações entre os nós, para entender as interações, prever conexões e analisar a intensidade de impactos, por exemplo: numa rede de proteínas, quais são as necessárias para um evento bioquímico; no facebook, quem se tornará amigo de quem; na disseminação de vírus e epidemias, o contágio de um hub seria catastrófico. Portanto, a análise dessas redes visa encontrar fenômenos e comportamentos sociais interessantes nas interações existentes no universo. Queremos investigar como a conectividade influencia a funcionalidade e os processos relacionados ao ambiente em questão.

2.2 – A Plataforma GitHub

O Github é a maior e mais popular plataforma de hospedagem, armazenamento e gerenciamento de código com o controle de versão distribuído Git (Git, 2008). Em 2018 o GitHub completa 10 anos e atualmente ela suporta aproximadamente 25 milhões de usuários, 3 bilhões de repositórios, dentre os quais apenas 25 milhões estão ativos, e mais de bilhões de commits. Ela é a principal rede social para programadores. O termo "social coding" (Git, 2008) já é conhecido no

ambiente de desenvolvimento de software, estamos numa nova era do mundo da computação, em que programar de forma transparente é uma ação de caráter social e humanista.

O Github [1] é também a plataforma que possui os recursos mais sociais. Ele ultrapassa muitas outras plataformas alternativas, não apenas fornecendo controle de versão, mas também fornecendo uma maneira de os desenvolvedores colaborarem em projetos de software de código aberto. Com a adição de funcionalidades comuns em redes sociais e quadros de produtividade como: follow, watch e fork, os usuários podem acompanhar as atividades de outros e estar ciente das mudanças nos projetos. Além disso, usando essas ferramentas na comunidade, os usuários podem criar e gerenciar seus projetos públicos, para os quais outros usuários podem contribuir [4].

A introdução de recursos sociais em um site de hospedagem de código atraiu muita atenção dos pesquisadores. Existem vários estudos interessantes envolvendo Github. Os estudos são geralmente caracterizadas como qualitativos e quantitativos. Qualitativos envolvem pesquisas de mais baixo nível, próximas dos usuários, enquanto quantitativos tentam extrair estatísticas de um conjunto relativamente grande de dados. Análises quantitativas por exemplo, podem mostrar se a estrutura de interações entre os usuários do Github se assemelha a estrutura de redes sociais mais comuns, como o Facebook e o Twitter, onde pouquíssimos tem milhares de amigos ou seguidores.

Vários estudos qualitativos entrevistaram usuários do GitHub para entender melhor como esses recursos sociais estão sendo usados [4]. As pesquisas mostram que as atividades dos desenvolvedores servem como um portfólio profissional que gera uma impressão para os outros, que então internalizam essas impressões para decidir quem e o que acompanhar ou onde contribuir em seguida. McDonald e Goggins [3] entrevistaram usuários do GitHub para identificar como medir o sucesso em seus projetos. O estudo mostra que os desenvolvedores veem os recursos sociais como um fator responsável pelo aumento da contribuição. Tsay et al. [4] realizaram um estudo quantitativo sobre 5.000 projetos para entender como os recursos sociais do GitHub impactam o sucesso do projeto. Pesquisas adicionais se estenderam além dos recursos sociais do GitHub. Thung et al. [5] construíram redes sociais de desenvolvedores envolvidos com 100.000 projetos do GitHub para demonstrar a estrutura social do ecossistema da plataforma. Outros [6, 7]

examinaram as localizações geográficas dos desenvolvedores do GitHub através de informações de localização auto-referidas, disponíveis em seus perfis.

Promover a disseminação efetiva de conhecimento é um aspecto importante em áreas de conhecimento intensivo como desenvolvimento de software. Estudos encontraram que rotinas formais devem ser suplementadas por processos sociais e colaborativos para promover aprendizagem e percepção. Seguir a pessoa certa é essencial para ter informação sobre "gurus" e líderes da comunidade [10].

Neste trabalho usaremos técnicas de ciências de dados para analisar atividades dos usuários no GitHub e também nos apoiaremos na análise das redes complexas para investigar esse ecossistema de computação social. Através da análise da estrutura do grafo social da comunidade de desenvolvedores do Github, desejamos encontrar grupos com interesses comuns, para assim ter um panorama do estado do desenvolvimento de software livre na cidade do Recife e fomentar idéias que apoiem o ecossistema de computação social na cidade.

3 – Coleta e Análise Exploratória dos Dados

A coleta dos dados foi feita através de requisições de busca na API Rest do Github. Lá podem ser feitas buscas em mais de 24 milhões de usuários que mantém cerca de 67 milhões de repositórios públicos e privados, dos quais apenas 25M estão ativos. Todos os meses mais de 85.000 novos usuários entram no GitHub [1]. Neste artigo, nos concentramos no comportamento social de desenvolvedores Recifenses que participam do GitHub, como já mencionado o objetivo é analisar o ecossistema de desenvolvimento de software open source na cidade de Recife. Assim, a nossa busca pode ser feita nas informações de localização que podem ser encontradas nos perfis dos usuários. A lista completa de endpoints públicos que existem no Github está disponível no site da API v3 do Github.

A API do GitHub suporta vários tipos de eventos, incluindo commits recentes, eventos de fork, solicitação de pull, informações dos desenvolvedores e acompanhamento de issues de todos os projetos do GitHub de código aberto. Cada evento, independentemente de seu tipo, geralmente inclui alguns metadados sobre as entidades envolvidas (por exemplo, as informações de perfil de um usuário, seu número e sua lista de seguidores, repositórios criados, as tecnologias de um repositório, etc.). Foram encontrados 2000 usuários que têm o campo de localização “Recife”. Como o campo é opcional, há pouca motivação para preenchê-lo com informações falsas. Portanto, pode-se razoavelmente assumir que a maioria das entradas não vazias é verdadeira.

Identificando os desenvolvedores. Foram encontrados cerca de 2000 desenvolvedores. Inicialmente coletamos os usuários candidatos (1000 usuários) que tem pelo menos 1 seguidor. Em seguida, para cada usuário, reunimos os metadados (ou seja, username, seguidores, repositórios e tecnologias). No total, analisamos cerca de **2000** desenvolvedores com **12000** seguidores e trabalhando em **10826** repositórios e **100** tecnologias diferentes. As Figuras abaixo mostram as estatísticas dos dados coletados.

Desenvolvedores	**Seguidores**	**Tecnologias**	**Repositórios**
1992	12100	100	10826

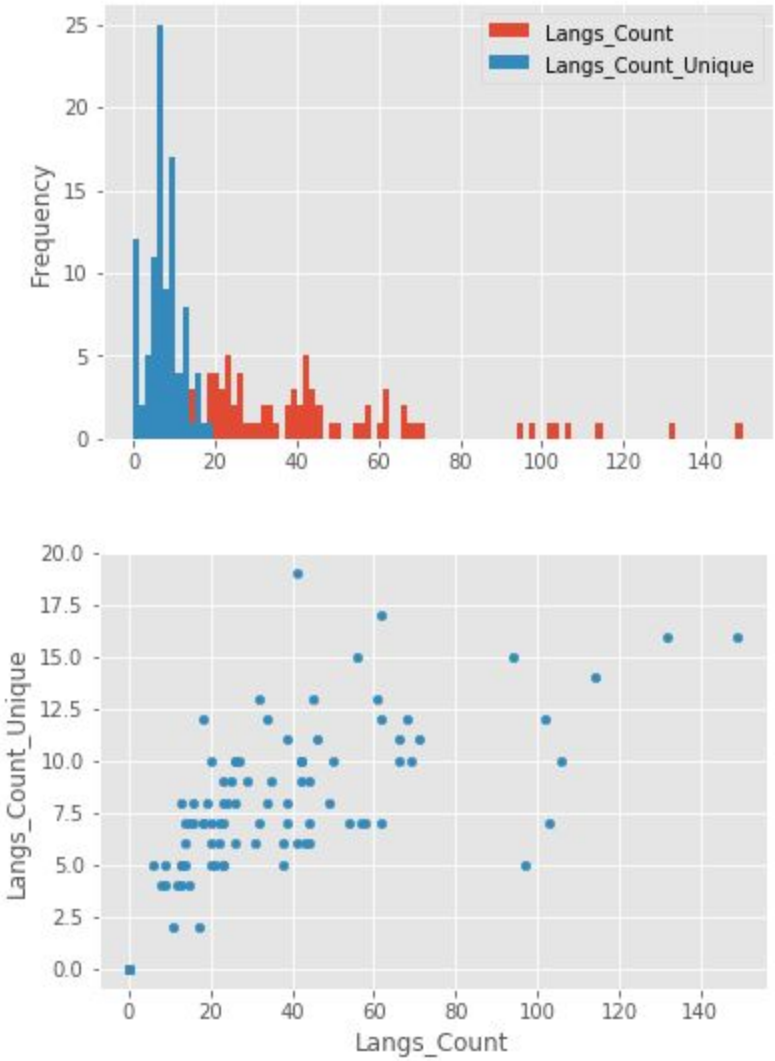

Figura 2: Correlação entre o número de tecnologias únicas (excluindo a repetição da mesma tecnologia em repositórios diferentes) e com repetição, ou seja, os desenvolvedores trabalham em repositórios que abrangem uma mesma tecnologia.

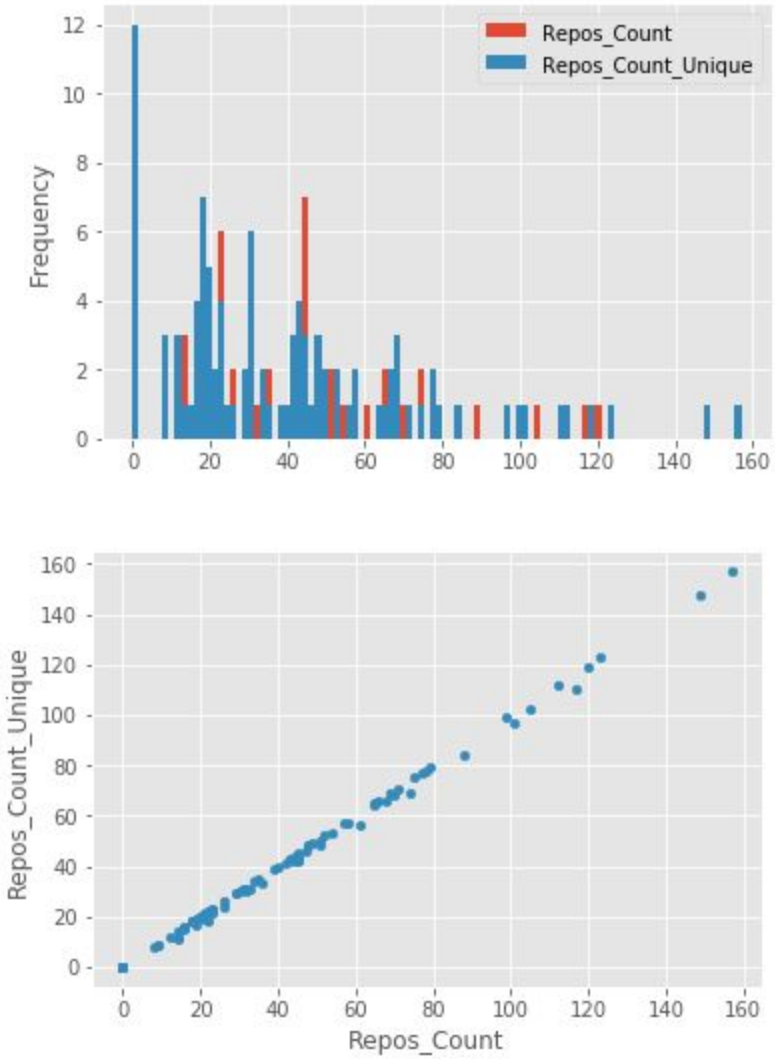

Figura 3: Correlação entre o nr de repositórios totais e únicos. Ou seja pouca frequência de forks

Identificando seguidores. O conjunto de seguidores é relativamente grande, pois coletamos um total de 12100 "relações de seguir" no conjunto de 2000 desenvolvedores recifenses. Porém, para cada desenvolvedor, identificamos os seguidores válidos como os que também são recifenses, portanto removemos os usuários que estão no conjunto de seguidores coletado inicialmente, resultando em apenas 2320 "relações de seguir". Em outras palavras, filtramos cerca de 9780 relações externas (provavelmente de usuários que não são Recifenses) que incidem na comunidade de desenvolvedores recifenses, o que mostra que a sub-rede é bem conectada à rede mundial do

GitHub. O gráfico na figura abaixo mostra a correlação entre o número de seguidores do nosso conjunto antes e depois da filtragem.

Seguidores(arestas) Totais	**Seguidores(arestas) Recifenses**
12100	2320

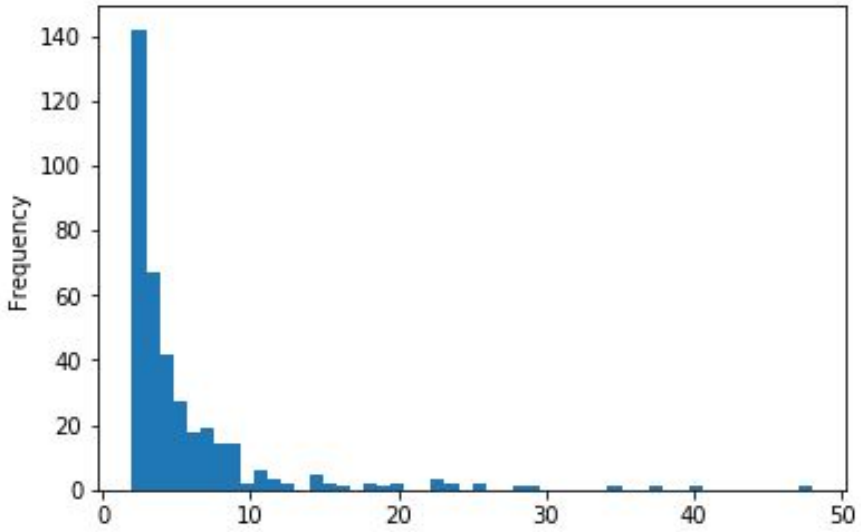

Figura 4. Distribuição de seguidores

Também com relação a distribuição de seguidores, após a filtragem, o número de desenvolvedores com 0 seguidores aumenta consideravelmente de 889 para 1300, o que nos dá a distância de 1103 para 692 desenvolvedores com pelo menos 1 seguidor Recifense. 411 desenvolvedores não interagem com outros desenvolvedores Recifenses, isto é, têm apenas seguidores de fora.

Devs. sem seguidores(0)	**Devs. sem seguidores(0) Recifenses**	**Devs com seguidores**	**Devs com seguidores apenas Recifenses**
889	1300	1103	692

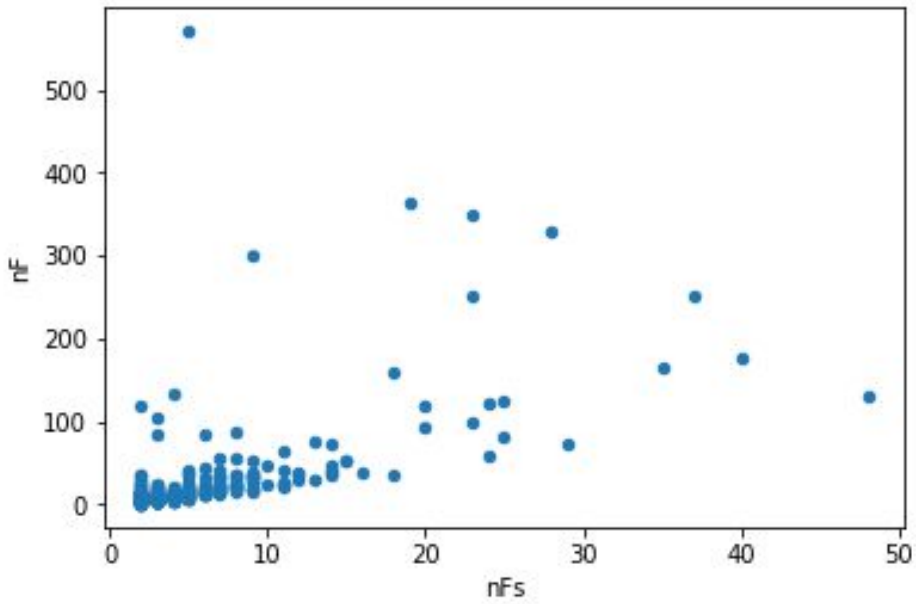

Figura 5. Correlação entre seguidores Globais e Recifenses

4 – Construção e Propriedades da Rede de Desenvolvedores

O grafo aleatório, rede gerada a partir de processos computacionais, foi bem estudado por matemáticos. Muitos resultados, tanto aproximados quanto exatos, foram provados com rigor. No entanto, a maioria das características interessantes das redes do mundo real que atraíram a atenção dos pesquisadores nos últimos anos diz respeito a como elas não se parecem com redes aleatórias. Redes reais não são aleatórias, de maneiras tão surpreendentes, que sugerem mecanismos que tanto podem estar guiando a formação, quanto explorando a estrutura da rede para atingir certos objetivos [8].

As técnicas de análise de redes sociais oferecem um rico conjunto de resultados interessantes e ferramentas para analisar as características estruturais (topológicas) e o comportamento humano em comunidades online, como por exemplo: análise de redes sociais, fenômeno do mundo pequeno, leis de potência, auto-organização e Teoria dos grafos. Nesta seção, será mostrada a técnica para a construção do grafo que representa nossa rede social, e a partir dela serão mineradas e analisadas algumas propriedades que parecem ser comuns as redes sociais.

Nossa hipótese é: as relações sociais formadas pelos desenvolvedores Recifenses são bem estruturadas. É de se se esperar que a cidade tenha um rico ecossistema de desenvolvimento de software já que existem grandes iniciativas reunidas no Porto Digital e pelo Grande Recife. Portanto, escolhemos esse subconjunto de usuários recifenses do Github para entender alguns padrões de comportamento social-técnico dos desenvolvedores e mostrar um paralelo com as propriedades encontradas em redes maiores já estudadas por vários pesquisadores ao redor do mundo.

Como já mencionado, no Github um desenvolvedor pode seguir outro por diversos motivos já explicitados, portanto, construímos a rede a partir dessa relação “seguir” entre os desenvolvedores, usaremos como características dos desenvolvedores a quantidade de tecnologias associadas. Acredita-se que estas informações podem refletir diretamente as relações dos usuários em atividades sociais e ainda inferir o perfil técnico.

Definição: A rede de seguidores pode ser definida como um Grafo direcionado $G = \langle V, E \rangle$, onde V é um conjunto de vértices que representam os usuários do conjunto de dados e E é um conjunto de pares de nós $E(V) = \{(u, v) \mid u, v \in V\}$ que representam o conjunto de arestas no grafo G. Se o nó *vi* segue o *vj*, então existe uma aresta do *vi* para o *vj*. Sejam *u1* e *u2* usuários, se *u1* segue *u2*, significa que as atividades de *u1* são influenciadas pelo *u2*.

Para um dado nó *vi*, o número de arestas que incidem nele é chamado de Grau de Entrada (deg-) e o número de arestas que saem a partir dele representa seu Grau de Saída (deg+). E o Grau Total é a soma do grau de entrada + saida (<deg−> + <deg+>). Montamos assim o grafo G. Na figura 6 mostramos a configuração inicial do grafo G quando importado no software Gephi.

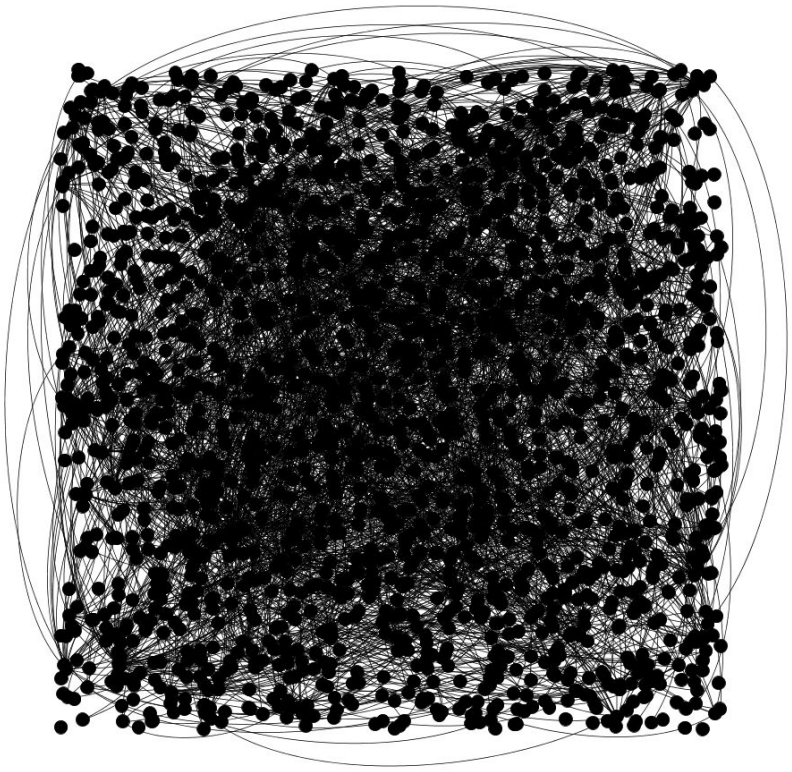

Figura 6. Construção do Grafo G: Grafo inicial quando os dados são importados no Gephi.

Várias técnicas e métodos são empregados para localizar um grupo de especialistas dentro de um organização ou comunidade. Em muitos sistemas, incluindo empresas de grande porte, diretórios pesquisáveis ou bancos de dados que incluem descrições do conhecimento e experiência dos funcionários são geralmente usados para localizar esses especialistas. O problema com essa abordagem é que as redes sociais, assim como as grandes empresas, estão em constante estado de mudança [10]. Em comunidades on-line e organizações dinâmicas de grande escala, torna-se inviável revisar e atualizar constantemente as informações de perfil de experiências e habilidades, visto que os próprios especialistas mudam rapidamente.

Estudos descobriram que uma pessoa com mais modificações no código pode ser um especialista dentro de uma comunidade, e essa experiência depende da porção do código que está sendo modificada. Porém, o "grau de conhecimento modelo" mostrou como essa perícia decai com posteriores alterações por outros autores [10]. Por isso, nos nossos experimentos, usaremos apenas as tecnologias associadas aos repositórios como uma métrica de habilidade técnica entre os desenvolvedores (e não a taxa de commits). A frequência em que as tecnologias aparecem nos repositórios também serão levadas em consideração. Usaremos o modelo do tf-idf (term frequency e inverse document frequency) no corpus de tecnologias para medir o grau de habilidade técnica de cada desenvolvedor e também para medida de similaridade.

Nas próximas seções, descrevemos algumas características presentes na nossa rede que são comuns em redes sociais.

4.1 – Densidade e Reciprocidade

Numa primeira análise do grafo de seguidores, existem cerca de 2000 vértices (desenvolvedores) e 2320 arestas (“Relações de seguir”). A figura 7 mostra a densidade das conexões entre os desenvolvedores recifenses que, em comparação com outras redes, de fato formam relações sociais ricas.

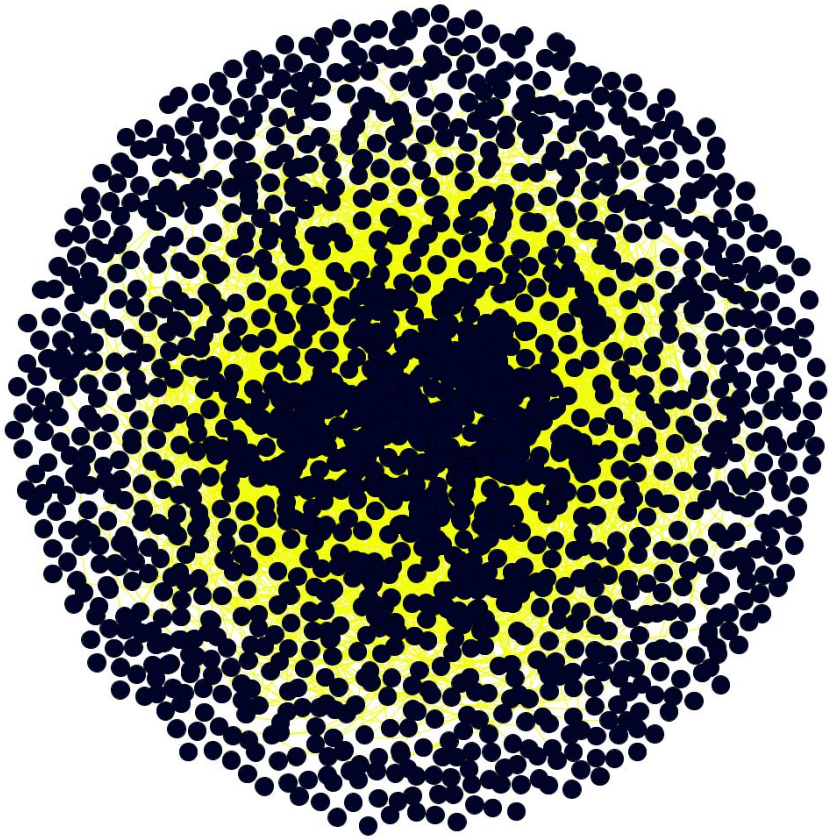

Figura 7. A densidade da rede de seguidores é de 000,1: os vértices estão pintados de preto e as arestas estão pintadas de amarelo. Fica claro que a rede é pouco densa pela pouca presença da cor amarela nas bordas, concentrada apenas no centro. A rede tem grau médio 1,133.

Isso significa que a metade dos desenvolvedores usam o GitHub sem a ajuda do recurso social baseado em "seguir". Com isso, é possível mostrar que um grande número de desenvolvedores é caracterizado por um comportamento coletivo sem uma centralização (comportamento Herd). Ainda, pela quantidade de desenvolvedores que ficaram sem seguidores após filtrarmos usuários recifenses Na Figura 8, mostramos a rede de seguidores pré-processada visualizada no Gephi [1].

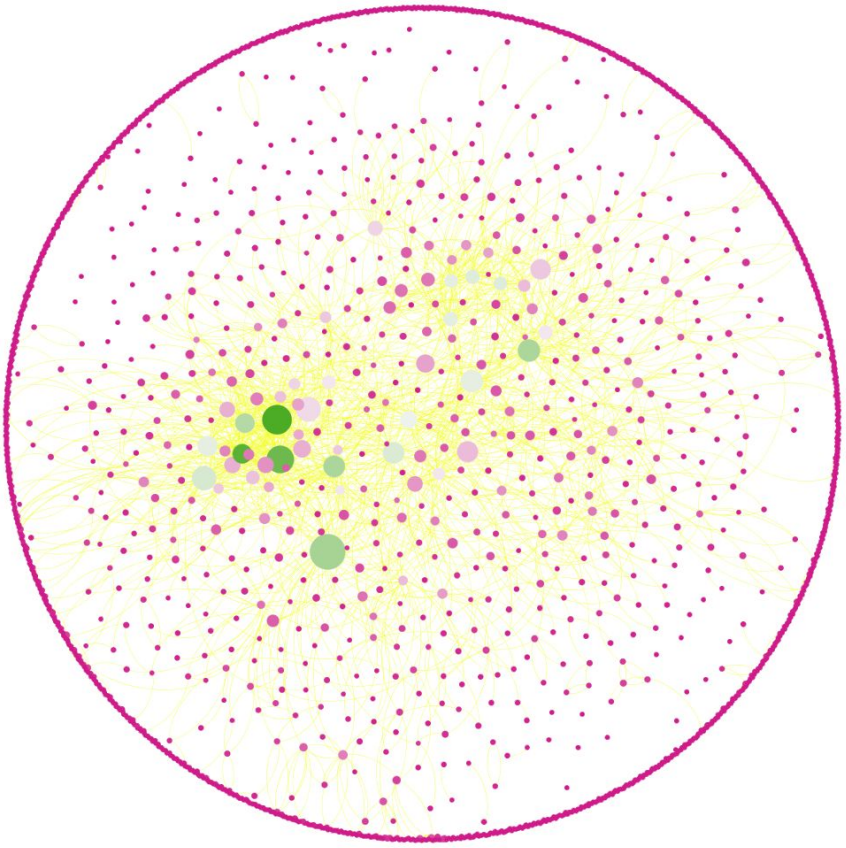

Figura 8. Rede de seguidores pré-processada visualizada no Gephi. O tamanho do nó indica maior grau de entrada, ou seja maior quantidade de seguidores. A borda densa é formada nós com grau 0, ou seja, nenhum seguidor.

Reciprocidade.

Reciprocidade é o conceito em redes sociais direcionadas de que dois atores estão mutuamente conectados. Em uma rede como a rede seguidora do GitHub ou do Twitter, isso significa que a relação recíproca entre u e v é dada se u segue v e v segue u. Nas discussões seguintes, chamaremos os vizinhos recíprocos de um determinado usuário de "amigos recíprocos".

Analisamos a reciprocidade no contexto do nosso trabalho porque a informação de reciprocidade ajuda a entender o uso do recurso social “seguir”. O recurso social pode ser usado para manter relações pessoais ou para acompanhar membros populares da comunidade ("estrelas" ou "gurus", normalmente caracterizados como hubs na rede social) [10].

Em nossa análise encontramos que a rede de seguidores recifenses é caracterizada por baixa reciprocidade: apenas 13% dos pares de usuários têm um relacionamento recíproco, ou seja, aresta indo e vindo, enquanto o restante dos pares de usuários têm um relacionamento unidirecional. Podemos comparar a reciprocidade da nossa rede com outras estudos sobre redes sociais: a reciprocidade do Flickr é de 68% e a do Yahoo! 84%. Esses sites têm reciprocidade relativamente alta. Enquanto o Twitter também tem uma baixa reciprocidade de apenas 22%, conforme relatado em [10].

Essas observações com relação à reciprocidade resultam na conclusão de que o grafo de seguidores dos desenvolvedores é comparável a um serviço de “news” como o Twitter, em vez de uma rede social (de amigos) típica. A reciprocidade consistentemente menor no GitHub é parcialmente motivada pela presença de alguns desenvolvedores populares, os chamados “desenvolvedores estrelas”, que exibem altos graus entrada e baixos graus de saída. Uma possível explicação para esse comportamento é que no GitHub muitos seguem outros desenvolvedores populares para obter atualizações sobre sua atividade de codificação em vez de manter relações pessoais.

Vamos usar um modelo de clustering adequado para essa rede social baseado no conceito de centralidade e intermediação (autoridade e reputação). Para isso, destacamos em seguida outras características pertinentes a redes sociais, como conectividade e caminho médio.

4.2 – Conectividade

Em geral, as redes podem ser divididas em duas partes, parte isolada e parte conectada. Na parte isolada, podemos encontrar dois padrões típicos, contendo o padrão de independência e o

padrão de grupo. A Figura 9 mostra alguns exemplos típicos destes. O padrão de independência indica que um desenvolvedor usa o Github de um modo tradicional e ele sempre se conecta apenas com conhecidos, ele apenas hospeda seu código ou acompanha um projeto interessante, mas raramente faz uma contribuição para ele.

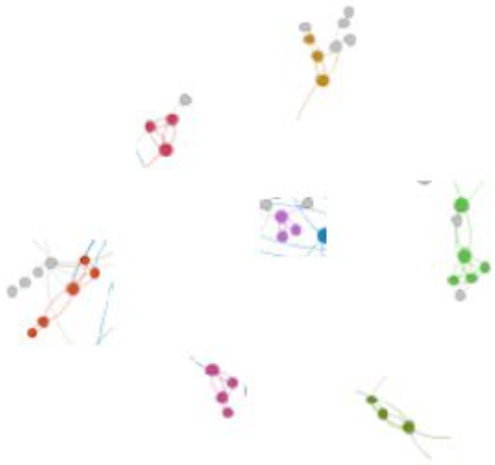

Figura 9. (exemplos de padrões conectados encontrados)

Para mostrar as características do padrão de grupo mais claramente, usamos dois algoritmos diferentes de layout de grafos direcionados no Gephi (à força) para redesenhar a rede de seguidores, como mostrado na figura 10. Segundo nossas estatísticas, no subconjunto de 2018-08, 49,33% de nós são isolados e 13,80% de nós se conectam apenas com um nó. O padrão de grupo é muitas vezes formado por desenvolvedores de um grupo que colaboram entre si para desenvolver o mesmo projeto.

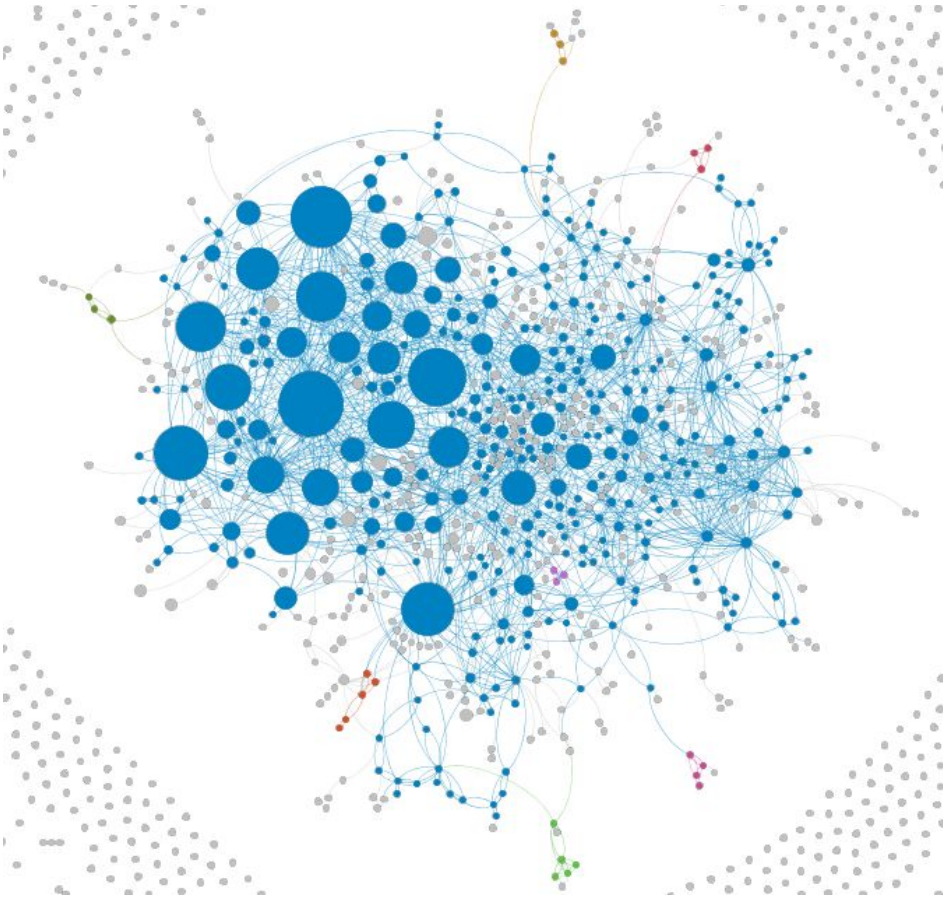

Figura 10. The Rich Club: Nós influentes.

4.3 – Efeito Mundo Pequeno

Talvez você já tenha ouvido falar dos famosos experimentos realizados por Stanley Milgram na década de 1960, nos quais as cartas passadas de pessoa a pessoa eram capazes de atingir um indivíduo-alvo designado em apenas um pequeno número de passos - em média seis, nos casos publicados. Esse resultado é uma das primeiras demonstrações diretas do chamado Efeito Mundo Pequeno, isto é, o fato de que a maioria dos pares de vértices na maioria das redes parecem estar conectados por um curto caminho [8].

A existência do efeito mundo pequeno havia sido especulada antes do trabalho de Milgram, em um notável conto de 1929 do escritor húngaro Frigyes Karinthy, e mais rigorosamente no

trabalho matemático de Pool e Kochen, que, embora publicado após os estudos de Milgram, estava em circulação na forma de pré-impressão por uma década antes que Milgram assumisse o problema. Hoje em dia, o efeito do mundo pequeno foi estudado e verificado diretamente em um grande número de redes diferentes e é mais conhecido como a lei dos "seis graus de separação" [8].

Caminho médio.

Considere uma rede não direcionada e vamos definir L como a distância geodésica média (isto é, do caminho mais curto) entre os pares de vértices:

$$\ell = \frac{1}{\frac{1}{2}n(n+1)} \sum_{i \geq j} d_{ij},$$

onde *dij* é a distância geodésica do vértice i até o j. Incluímos também a distância de cada vértice até ele mesmo (que é zero) nessa média. Isso é matematicamente conveniente por vários motivos, mas nem todos os autores o fazem. Em qualquer caso, sua inclusão simplesmente multiplica ℓ por *(n - 1) / (n + 1)* e, portanto, fornece uma correção de ordem $n - 1$, que é freqüentemente insignificante para propósitos práticos [8].

Para uma rede de n vértices e m arestas o cálculo do ℓ pode ser feito no tempo *O(mn)* usando busca em largura simples. Os valores de ℓ para uma variedade de redes sociais são em todos os casos bastante pequenos - muito menores que o número *n* de vértices. Por exemplo, autores mostraram que o Twitter tem um comprimento médio de caminho de 4,12. Em uma análise da rede de mensagens instantâneas foi encontrado o comprimento médio de caminho igual a 6. A maior rede social até o momento, o Facebook, tem um comprimento médio de caminho de 4,7 [7].

A definição (1) de ℓ é problemática em redes que possuem mais de um componente. Em tais casos, existem pares de vértices que não possuem caminho de conexão. Por outro lado, atribui-se uma distância geodésica infinita a esses pares, mas o valor de ℓ também se torna infinito. Para evitar

esse problema, geralmente define-se ℓ em tais redes como a distância geodésica média entre todos os pares que possuem um caminho de conexão. Pares que caem em dois componentes diferentes são excluídos da média. Os números da análise foram todos calculados desta maneira.

Algumas redes têm distâncias média que aumentam mais lentamente que *log n*. Pesquisas mostraram que as redes com distribuição de grau que seguem a lei de potência têm valores de ℓ que não aumentam mais rápido que *log n / n log n*, já outras pesquisas apresentam argumentos que sugerem que a variação real pode ser até mais lenta que isso [8].

Na análise da rede, para 90% dos pares de nós, o comprimento médio do caminho é 6,4 ou menor. Para 50% dos pares de nós, o comprimento médio do caminho é 5 ou menor. O comprimento médio do caminho na rede de seguidores do GitHub está alinhado com os seis graus de separação esperados em redes sociais.

Influência e Autoridade.

Uma rede social on-line como o Github e seus usuários formam um sistema sociotécnico no qual a persistência da comunidade depende tanto da interação social entre os usuários quanto da implementação e design da plataforma da rede social em questão [5]. Especificamente, a plataforma tecnológica subjacente da rede social pode alterar positiva ou negativamente o ambiente dos usuários. Como exemplo, as mudanças nos recursos da plataforma, como a interface do usuário, podem representar um risco geral para o envolvimento do usuário na rede social [5]. Porque basta que apenas alguns poucos desenvolvedores influentes fiquem insatisfeitos para que a comunidade se acabe. Nesta seção, fornecemos evidências de que os usuários "centrais" desempenham um papel importante na rede em relação à formação de estrutura de comunidade.

Uma propriedade importante de muitas redes sociais é a presença de influenciadores (ou seja, líderes comunitários). As decisões relativas à participação ou à saída da rede são determinadas em grande parte pelo número de amigos e pelo próprio envolvimento. Assim, os usuários que deixam uma comunidade têm efeitos indiretos negativos sobre seus amigos [6]. Como resultado,

isso pode desencadear uma cascata de usuários que partem, o que pode colocar em risco toda a comunidade. Portanto reiteramos que os usuários influentes são os que têm mais autoridade na rede.

Aqui, usamos uma aproximação simplificada da autoridade do desenvolvedor com base no grau de seguidores. Observe, no entanto, que o grau de seguidores é um parâmetro essencial do nosso modelo de autoridade personalizada, por motivos já explanados anteriormente. Vamos definir a autoridade do desenvolvedor baseada no seguidor usando técnicas e algoritmos de análise de links como PageRank e Edge Betweenness (Intermediação de arestas) que podem ser aplicados em nossa análise da comunidade da redes social do Github e que têm resultados práticos satisfatórios.

Técnicas de personalização, como o PageRank sensível a contexto [8,9], permitem a classificação de importância dos desenvolvedores sensível ao contexto. Algoritmos de análise de links foram aplicados com sucesso para estimar a importância do ator em redes sociais e colaborativas [7], bem como para sistemas de produção em massa on-line, como os emergentes ambientes de crowdsourcing.

Autores [7] propuseram técnicas de análise de links, como o PageRank, para mineração de conhecimento em comunidades online. No entanto, nenhuma personalização com relação às áreas de especialização foi realizada por trabalho anterior. Em outros trabalhos [5,6] também foram propostos outros algoritmos de análise de link sensíveis ao contexto para a mineração de conhecimento, na figura abaixo detectamos os desenvolvedores influentes utilizando o algoritmo de Page Rank.

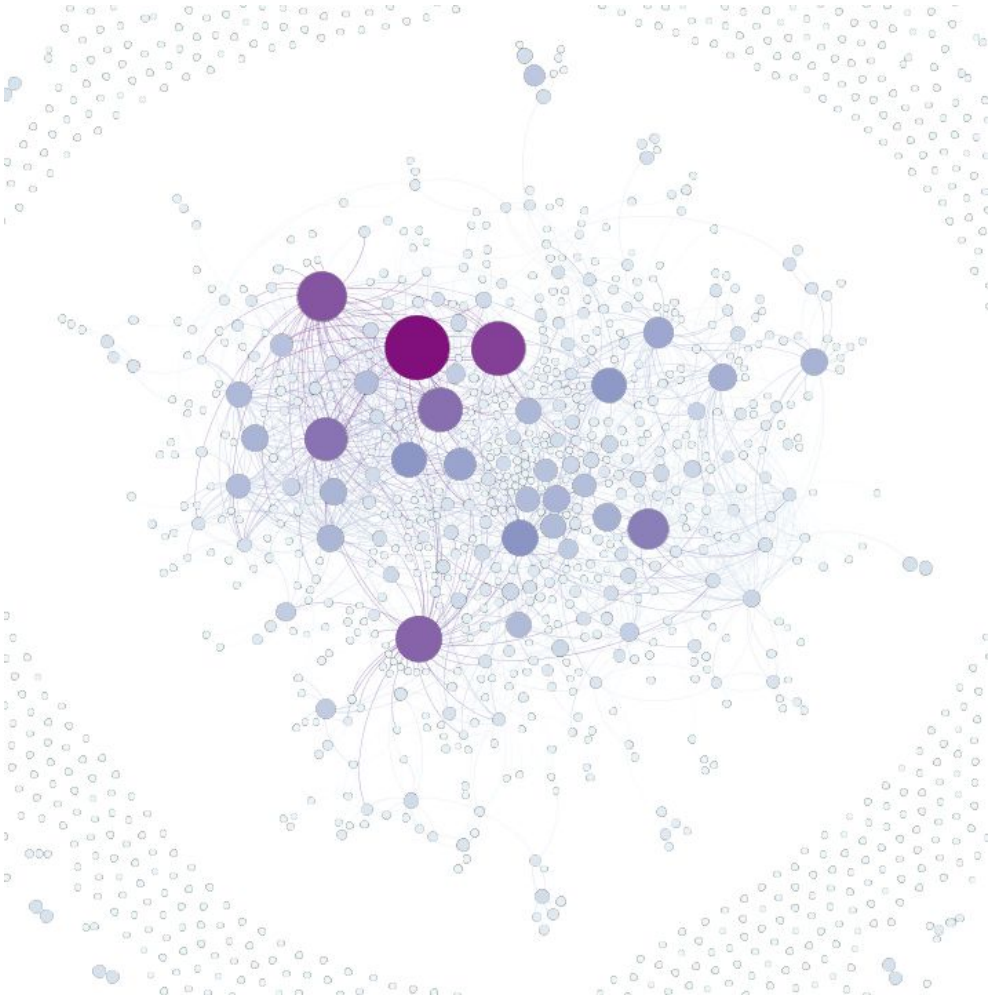

Figura 11. Rede ilustrando desenvolvedores influentes utilizando o algoritmo de Page Rank. Os 5 desenvolvedores mais influentes são respectivamente lmarinho, filipeximenes, lailsonbm, brunofarache, fernandocastor.

Propomos uma métrica baseada na topologia da rede para capturar autoridade e detecção da estrutura de comunidades. A métrica de autoridade proposta mede a posição relativa (ou seja, reputação) de um membro da comunidade utilizando o algoritmo Edge Betweenness. A métrica baseia-se na estrutura das relações sociais entre os desenvolvedores, a rede com os desenvolvedores influentes é ilustrada na figura 12.

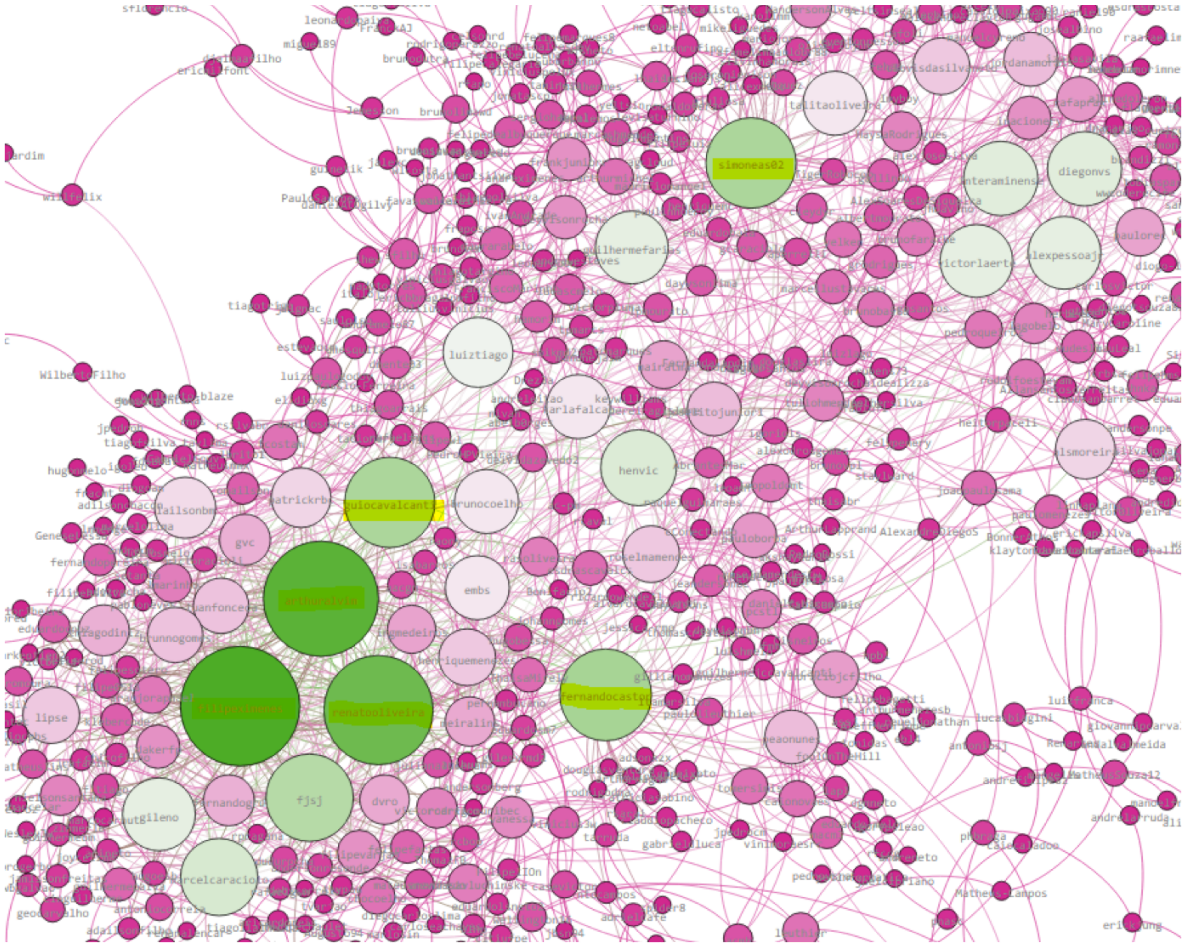

Figura 12. Rede com os desenvolvedores influentes. Podemos ver claramente os 5 desenvolvedores mais influentes, respectivamente filipeximenes, arthuralvim, renatooliveira, guicavalcanti, fernandocastor.

5 – A Comunidade de Desenvolvedores

Estrutura de comunidade é uma propriedade bem conhecida na maioria das redes sociais, ou seja, grupos de vértices coesos que têm uma alta densidade de arestas ou conexões entre eles, e com uma menor densidade de arestas entre os grupos. Por conta do compartilhamento de características comuns fica possível dividir as pessoas em grupos de acordo com áreas de interesse, ocupação, idade e assim por diante.

Essa característica está presente nas redes acontece muitas vezes por conta do fenômeno chamado assortatividade, ou seja indivíduos relacionados tendem a compartilhar características tais como, características físicas, hábitos e opiniões. Vários estudos sugerem que certamente esse é o caso, principalmente nas redes sociais. É possível que uma rede tenha uma combinação de variedades, mas não uma estrutura de comunidade. Isso pode ocorrer, por exemplo, quando há uma mistura alternativa por idade ou outras grandezas escalares. Às vezes, redes com esse tipo de estrutura são ditas ”estratificadas".

5.1 – Detecção de Comunidades

Outra questão ligeiramente diferente sobre a estrutura da comunidade, mas relacionada ao propósito do estudo é: se alguém recebe um exemplo de vértice extraído de uma rede conhecida, pode-se identificar a comunidade à qual ele pertence? Os métodos algorítmicos para responder a essa pergunta seriam claramente de algum valor prático para pesquisar redes como a World Wide Web e redes de citações. Pesquisadores encontraram o que parece ser um algoritmo de muito sucesso, pelo menos no contexto da Web, baseado em um método de fluxo máximo [9].

Existem diversos algoritmos que se propõe a responder a questão acima, alguns com resultados satisfatórios com um propósito mais geral como o tradicional algoritmo de clustering hierárquico, Block Model e Equivalência Estrutural, outros para alguns tipos específicos de redes como o Algoritmo encaixe de mola, Infomap e Algoritmo de Girvan-Newman para redes sociais. O leitor pode buscar referências deste documento [9] se tiver interesse em detalhes sobre os algoritmos mencionados. Estes detalhes serão omitidos daqui por não pertencerem ao objetivo deste

estudo. Descrevemos aqui brevemente o método tradicional para fins de entendimento sobre o método de detecção de comunidade.

Algoritmo Clustering Hierárquico.

O método tradicional para extrair a estrutura da comunidade de uma rede é a análise de cluster, às vezes também chamada de clustering hierárquico [8]. Nesse método, atribuímos uma "força ou peso de conexão" aos pares de vértices na rede de interesse. Em geral, cada um dos *1/2 n (n - 1)* pares possíveis em uma rede de *n* vértices é atribuída tal força, não apenas aqueles que estão conectados por uma aresta, embora existam versões do método onde nem todos os pares são atribuídos uma força; Nesse caso, pode-se supor que os pares restantes tenham uma força de conexão igual a zero.

O algoritmo inicia com *n* vértices sem arestas entre eles, arestas são adicionadas em ordem decrescente de força/peso de conexão vértice-vértice. Pode-se fazer uma pausa em qualquer ponto desse processo e examinar a estrutura da componente formada pelas arestas adicionadas até o momento; essas componentes são consideradas comunidades (ou "clusters") nesse estágio do processo. Quando todas as arestas foram adicionadas, todos os vértices estão conectados a todos os outros e há apenas uma comunidade. Todo o processo pode ser representado por uma árvore ou dendograma de operações de união entre conjuntos de vértices nos quais as comunidades, em qualquer nível, correspondem a um corte horizontal através da árvore - como na Figura 13.

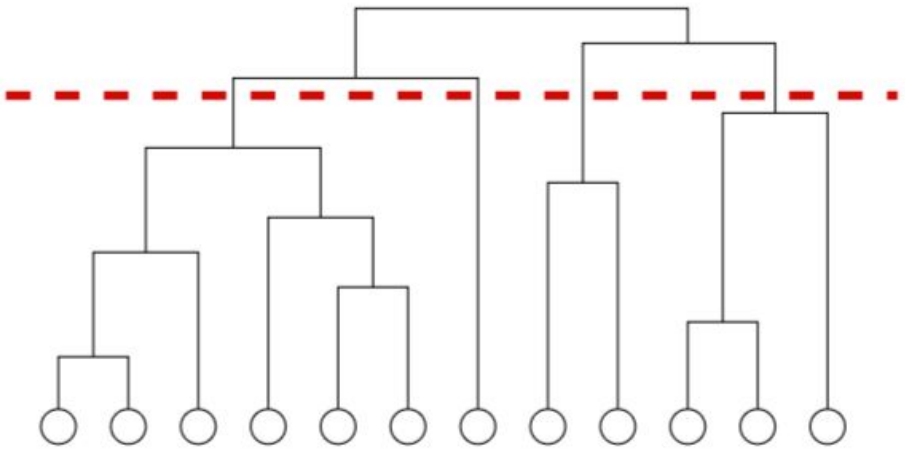

Figura 13. Ilustração do dendograma gerado pelo Algoritmo Clustering hierárquico. Cortes horizontais correspondem a partições das comunidades no grafo.

O agrupamento é realizado de acordo com as várias definições possíveis para força da conexão. Escolhas razoáveis incluem várias medidas de distâncias para arestas ponderadas, os tamanhos mínimos para corte dos conjuntos (ou seja, fluxo máximo) [9] e contagens de caminhos ponderados entre vértices. Recentemente, vários autores tiveram sucesso com métodos baseados em Edge Betweenness, que é a contagem de caminhos geodésicos entre os vértices ao longo de cada aresta da rede [8].

Os resultados parecem mostrar que, pelo menos para as redes sociais e biológicas, a estrutura de comunidade é uma propriedade comum, embora algumas redes alimentares não possuam essas estruturas. As redes alimentares podem ser diferentes das outras redes por parecerem densas: o grau médio dos vértices aumenta aproximadamente linearmente com o tamanho da rede, em vez de permanecer constante como na maioria das redes. O mesmo deve acontecer também em redes metabólicas [8].

O agrupamento de redes não deve ser confundido com a técnica de agrupamento de dados, que é uma forma de detectar agrupamentos de pontos de dados em espaços de alta dimensão [9]. Os dois problemas têm algumas características comuns, e algoritmos para um podem ser adaptados para o outro, e vice-versa. Por exemplo, dados de alta dimensão podem ser convertidos em uma rede colocando-se arestas entre pontos de dados espaçados e, em seguida, algoritmos de cluster de rede podem ser aplicados ao resultado. Em suma, normalmente descobre-se que os algoritmos especialmente concebidos para o agrupamento de dados funcionam melhor do que esses métodos emprestados, e o mesmo também com clustering em redes.

Block Model e Equivalência Estrutural.

Na literatura de redes sociais, o agrupamento de redes tem sido discutido em grande parte no contexto dos chamados modelos de blocos [9], que são essencialmente divisões de redes em comunidades ou blocos de acordo com um determinado critério. Os sociólogos se concentraram

particularmente na equivalência estrutural. Dois vértices em uma rede são considerados estruturalmente equivalentes se tiverem todos os mesmos vizinhos. A equivalência estrutural exata é rara, mas a equivalência aproximada pode ser usada como base para um método de agrupamento hierárquico como o descrito acima.

5.2 – Análise da Comunidade

Seria de algum interesse e importância prática se descobríssemos quais os tipos de redes mostram a estrutura de grupo forte e bem definida (como algumas mostradas na figura 14). Por exemplo, pode-se imaginar que as redes de citação se dividiram em grupos representando áreas particulares de interesse de pesquisa, e uma grande quantidade de energia foi investida em estudos sobre esse fenômeno. Da mesma forma, as comunidades na World Wide Web podem refletir o assunto das páginas; comunidades em redes metabólicas, neurais ou de software podem refletir unidades funcionais; comunidades em redes alimentares podem refletir subsistemas dentro de ecossistemas, e assim por diante [8].

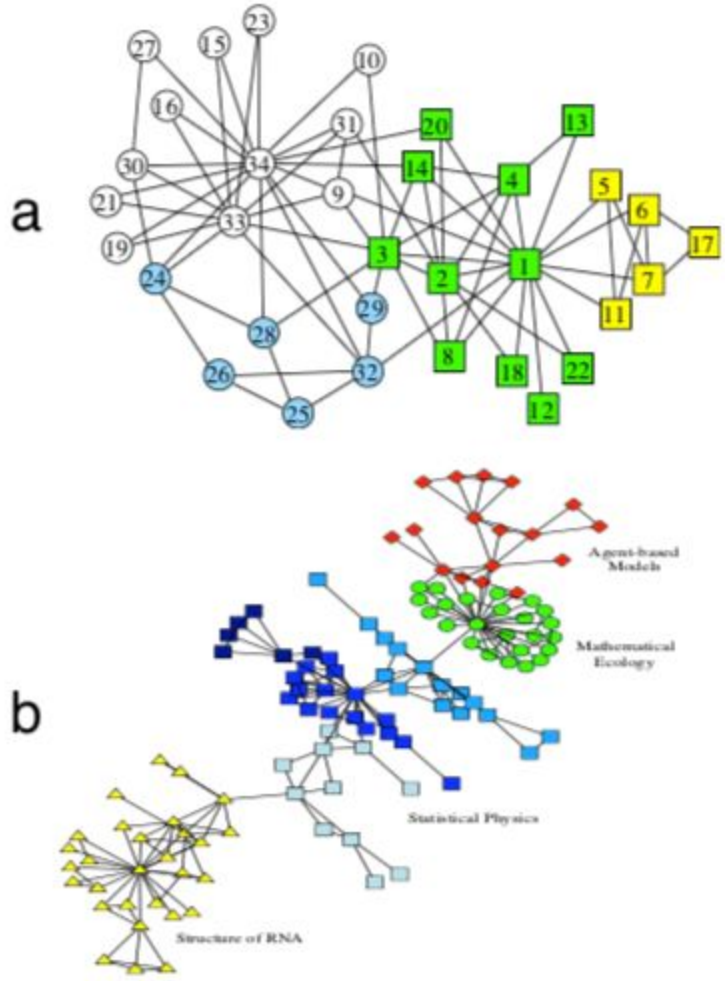

Figura 14. Estrutura de comunidades em algumas redes sociais. A figura a) é uma rede social do clube de karatê de Zachary, bastante usada como benchmark dos algoritmos de detecção de comunidades. A figura b) é a rede de colaboração de cientistas do Instituto de Santa Fé. Os vértices foram coloridos de acordo com a comunidade detectada utilizando o Algoritmo de Modularidade de Girvan-Newman como critério de parada.

O GitHub permite que os usuários criem repositórios e, em seguida, os criadores e/ou colaboradores os alterem. Por exemplo, se um desenvolvedor cria um projeto, significa que o projeto está diretamente relacionado a ele. Nesse caso, ele está diretamente em contato com as tecnologias associadas ao projeto. Portanto, os seguidores, tecnologias e repositórios de um desenvolvedor são métricas de um contexto social e técnico que vamos analisar. Por um lado a quantidade de seguidores indica o grau de conexão social entre os desenvolvedores, enquanto a

quantidade de repositórios e tecnologias envolvidas indicam um grau de atividade open source e habilidade técnica.

Pesquisas recentes [11] testaram a relação do nível de aceitação da contribuição com a conexão social entre o proprietário(s) e contribuidor, examinando a associação da distância social (caminho) e interações prévias com a aceitação da contribuição e encontraram que ambas as medidas de conexão social foram associadas positivamente à aceitação do pedido pull request, ou seja, são fortes critérios para a aceitação da contribuição. A medida de distância social teve a mais forte influência na probabilidade de aceitação em comparação com outros fatores no nível da solicitação de pull, aumentando a aceitação em 187% quando quem submete segue o gerente de projeto.

Isso demonstra que a habilidade técnica não é o principal fator (critério) para que uma contribuição open source seja aceita pela comunidade. Entretanto, nenhum trabalho foi feito na direção de descobrir se há alguma a correlação entre conexão social com habilidades técnicas comuns. Aqui vamos medir essa correlação através da análise da estrutura comunidade formada e descobrir o quão forte é a relação da conexão social com habilidade técnica através da caracterização das comunidades por tecnologias. Interações prévias entre desenvolvedores também são positivamente associada à aceitação, aumentando a probabilidade de aceitação em 35,6%. Vamos deixar essa análise de lado já que a correlação não é tão grande quando comparada a medida de distância social e também por conta da limitação de escopo do nosso estudo.

Em nossa implementação, extraímos a estrutura da comunidade da rede de seguidores usando o Gephi que integra e otimiza um algoritmo bem conhecido de detecção de comunidades [12]. O tamanho da comunidade é representado pelo número de desenvolvedores que ela contém e a atividade de um desenvolvedor é refletido pelas tecnologias que ele teve contato através dos repositórios que ele criou, ou em que foi adicionado como colaborador no presente momento da coleta dos dados. Assim esperamos que essas informações reflitam o grau de interesse, atividade e perfil de um desenvolvedor por diferentes projetos e tecnologias.

Na Fig. 15 mostramos uma visualização da rede de seguidores. A figura foi criada usando o algoritmo Modularidade de Louvain, no qual calcula-se o grau de modularidade de cada subgrupo da rede e move-se os nós de acordo com essa medida.

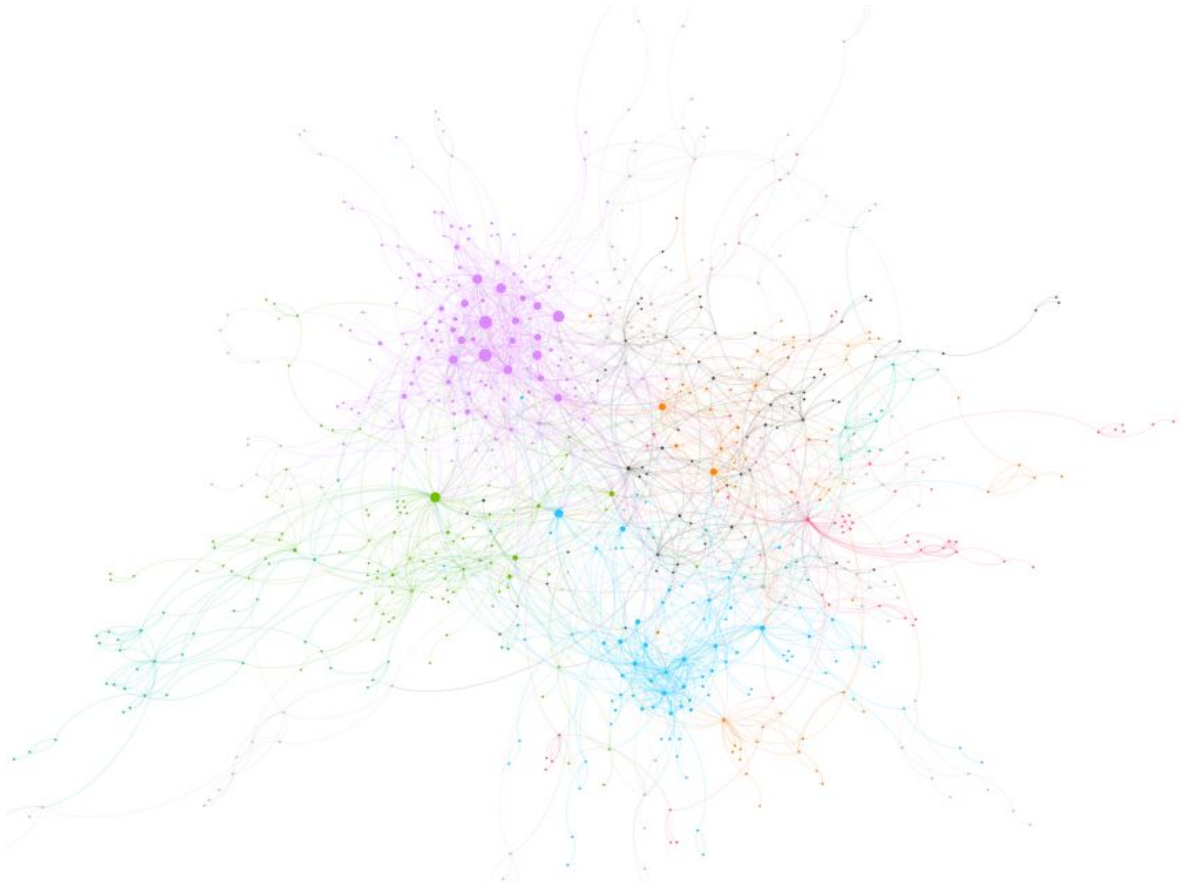

Fig. 15. Grupos formados na comunidade de desenvolvedores que tem a topologia formada pelos seguidores e caracterizada por tecnologias. Os maiore grupos se formam em volta dos desenvolvedores mais influentes de acordo com a centralidade Eigenvector.

Podemos ver Na Fig. 11 a existência de 8 principais comunidades no maior componente conectado. Cada conjunto de nós (ou seja, desenvolvedores) na mesma comunidade é densamente conectado internamente. Descobrimos que as comunidades se formaram espontaneamente entre desenvolvedores compartilhando interesses comuns na rede; isso espelha um fenômeno bem conhecido nas redes sociais de e-mail e na rede "seguidores" do twitter por exemplo.

A razão de usarmos tal algoritmo se dá pela popularidade e conhecimento de resultados práticos quando aplicados a redes sociais do mundo real, apesar de não termos certeza que o algoritmo revelaria algo particularmente útil sobre a rede, demonstrando uma estrutura de comunidade forte o suficiente para que, de fato, as comunidades apareçam claramente na figura 15.

Colorimos os vértices de acordo com as comunidades associadas aos desenvolvedores (que eles representam), como mostrado na figura 16, torna-se claro que uma das principais divisões na rede é pelas tecnologias associadas, e isto é presumivelmente o que está impulsionando a formação de comunidades neste caso. Outra possível divisão seria entre pessoas que trabalham na mesma empresa ou estudam na mesma faculdade, sendo geralmente o principal motivo das pessoas se engajarem no github.

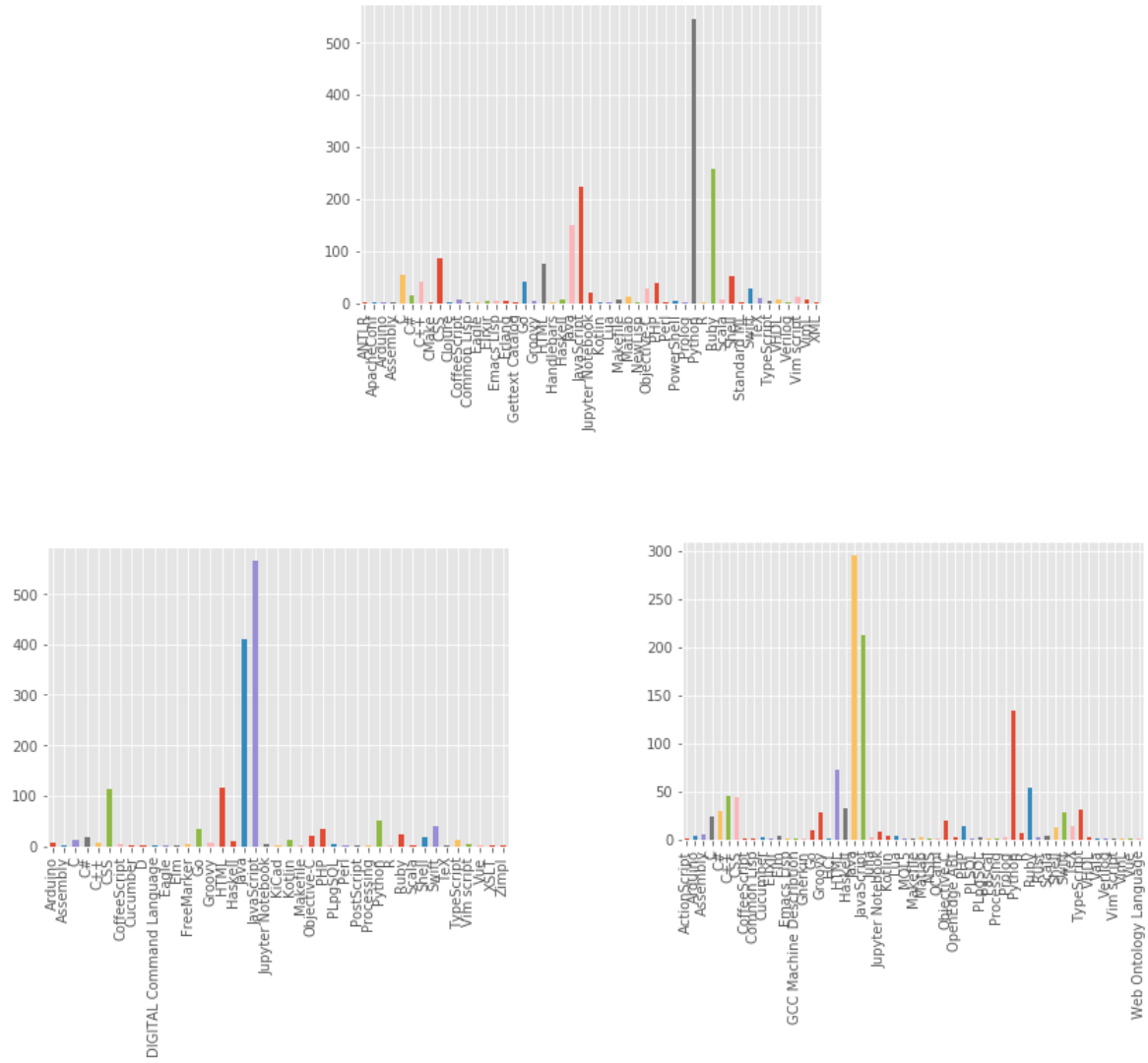

Figura 16. Caracterização dos 3 maiores grupos detectados o Rosa, Azul e Verde respectivamente. No grupo Rosa encontramos mais desenvolvedores Python, no grupo Azul temos principalmente desenvolvedores Javascript enquanto no grupo verde encontramos mais desenvolvedores Java.

6 – Conclusão e Trabalhos Futuros

Para concluir, argumentamos sobre certas propriedades que o maior componente conectado na a rede social deve preservar para garantir que o nível de colaboração seja efetivo na rede.

- De acordo com a análise, a rede de seguidores é aparentemente livre de escala, ou seja, o grau de distribuição segue a lei de potência.

- Observamos que a rede depende mais dos usuários 'centrais' em relação à informação ou propagação de notícias através da mesma. O comprimento médio do caminho deve aumentar rapidamente após a remoção do nó, e isto é garantido pela distribuição de grau que segue a lei de potência.

- O maior componente conectado do GitHub no grafo de seguidores tem um grau médio muito menor quando comparado a outras redes sociais. O grau médio da rede de seguidores é 2.8, indica que há muito mais usuários/colaboradores do que seguidores (rede pouco densa). Como os usuários seguem em média apenas dois desenvolvedores, o grau médio é baixo. A rede de seguidores do GitHub é usada principalmente para receber atualizações de membros da comunidade popular, em vez de manter relações pessoais.

- O grafo de seguidores do GitHub é altamente sensível à remoção de nós baseada em autoridade do seguidor. Já em um número de 5% de nós removidos, o grau médio, assim como o comprimento médio do caminho, cai para 1⁄4 no maior componente conectado.

- Nossos resultados mostram que 5% dos usuários desempenham um papel central no GitHub, para ter um grafo de seguidores conectado. O desempenho desses desenvolvedores centrais deve ser considerado antes da introdução de um recurso em toda a comunidade, para evitar a insatisfação dos líderes e influenciadores da comunidade.

Toda análise está codificada em notebooks jupyter, em um repositório do github: github.com/karlafalcao/network-analysis.

Referências Bibliográficas

1. Eirini Kalliamvakou, Georgios Gousios, Kelly Blincoe, Leif Singer, Daniel M. German, and Daniela Damian. The promises and perils of mining GitHub. In Proceedings of the 11th Working Conference on Mining Software Repositories (MSR 2014). ACM, New York, NY, USA, 92-101. 2014.

2. Kelly Blincoe, Jyoti Sheoran, Sean Goggins, Eva Petakovic, Daniela Damian. Understanding the popular users: Following, affiliation influence and leadership on GitHub. Information and Software Technology, ISSN: 0950-5849, Vol: 70, Page: 30-39. 2016

3. N. McDonald and S. Goggins. Performance and participation in open source software on github. In CHI'13 Extended Abstracts on Human Factors in Computing Systems, pages 139–144. ACM, 2013.

4. Jason T. Tsay, Laura Dabbish, and James Herbsleb. 2012. Social media and success in open source projects. In Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work Companion (CSCW '12). ACM, New York, NY, USA, 223-226.

5. F. Thung, T. Bissyande, D. Lo, and L. Jiang. Network structure of social coding in github. In 17th European Conference on Software Maintenance and Reengineering (CSMR), pages 323–326, 2013.

6. Antonio Lima, Luca Rossi, Mirco Musolesi. Coding Together at Scale: GitHub as a Collaborative Social Network. Proceedings of 8th AAAI International Conference on Weblogs and Social Media, 2014

7. Yue Yu, Gang Yin, Huaimin Wang, and Tao Wang. 2014. Exploring the patterns of social behavior in GitHub. In Proceedings of the 1st International Workshop on Crowd-based Software Development Methods and Technologies (CrowdSoft 2014). ACM, New York, NY, USA, 31-36. DOI=http://dx.doi.org/10.1145/2666539.2666571

8. Newman, M., The Structure and Function of Complex Networks. SIAM Review, volume 45 number 2 pages 167-256, 2003, https://doi.org/10.1137/S003614450342480

9. Santo Fortunato, Community detection in graphs. Physics Reports, Volume 486, Issues 3–5, 2010, Pages 75-174, ISSN 0370-1573, https://doi.org/10.1016/j.physrep.2009.11.002.

10. Daniel Schall, Who to follow recommendation in large-scale online development communities. Information and Software Technology, Volume 56, Issue 12, 2014, Pages 1543-1555, ISSN 0950-5849, https://doi.org/10.1016/j.infsof.2013.12.003.

11. Jason T. Tsay, Laura Dabbish. Influence of Social and Technical Factors for Evaluating Contribution in GitHub.

12. Fast unfolding of communities in large networks, Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, Etienne Lefebvre, Journal of Statistical Mechanics: Theory and Experiment 2008 (10), P1000