Ecossistemas de colaboração em redes de desenvolvimento de software: definição e caracterização

Gabriel Lage Calegari[1], **Ana Paula Couto da Silva**[1]

[1] Departamento de Ciência da Computação
Universidade Federal de Minas Gerais (UFMG) – Belo Horizonte, MG – Brasil

{gabrielcalegari,ana.coutosilva}@dcc.ufmg.br

Abstract. *Collaboration is an activity present in different organizations and biological levels, and has been the object of study in several areas of science, especially biology. In recent years, a human collaboration system has gained notoriety: collaborative software development networks. Among many web portals dedicated to the topic, GitHub is the most popular of them. In this work, we focus on GitHub to analyze the collaborative development of software, adapting the biological concept of ecosystem in order to obtain a complementary view in relation to those obtained with the methodologies practiced in the literature. This analysis seeks to understand the main characteristics of collaborative ecosystems on GitHub and if there are similarities between them.*

Resumo. *A colaboração é uma atividade presente em diferentes organizações e níveis biológicos, e tem sido objeto de estudo de diversas áreas da ciência, sobretudo a biologia. Nos últimos anos, um sistema de colaboração humano ganhou notoriedade: as redes de desenvolvimento colaborativo de software. Entre muitos portais Web dedicados ao tema, o GitHub é o mais popular deles. Neste trabalho, focamos no GitHub para analisar o desenvolvimento colaborativo de software, adaptando o conceito biológico de ecossistema com o objetivo de obter uma visão complementar com relação às obtidas com as metodologias praticadas na literatura. Esta análise busca entender as principais características dos ecossistemas de colaboração no GitHub e se existem semelhanças entre eles.*

1. Introdução

A colaboração é uma atividade presente em diferentes organismos, organizações e níveis biológicos: de genes cooperando para genomas, passando por células colaborando para criar organismos, até organismos já constituídos colaborando entre si. Nas sociedades humanas, a colaboração faz parte do circuito social, econômico e científico [Lopes et al. 2009, Fehr and Schurtenberger 2018].

Tradicionalmente, sistemas de colaboração têm sido estudados por diferentes áreas da ciência, sobretudo na biologia. Há trabalhos buscando entender a colaboração a partir da teoria da evolução biológica [Hamilton 1963, Nowak and Sigmund 1998], alguns com foco na colaboração humana [Barclay and Raihani 2016]. Outros trabalhos [Rand et al. 2011, Gracia-Lázaro et al. 2012] têm se dedicado ao entendimento dos sistemas de colaboração humanos sob a perspectiva de redes, uma vez que esses sistemas apresentam características de sistemas complexos.

Considerando a área de tecnologia da informação, o surgimento do movimento de software livre contribuiu para o aumento de notoriedade de um sistema de colaboração humano específico: as redes de desenvolvimento colaborativo de software. O software livre é essencialmente colaborativo, uma vez que qualquer indivíduo pode ler, modificar e melhorar o código existente. Essas redes tornaram-se uma fonte quase inesgotável de troca de conhecimento em torno do desenvolvimento de software, impulsionando o aumento de trabalhos científicos que buscam entender a dinâmica da criação e fortalecimento da colaboração estabelecida entre os indivíduos nesses sistemas.

Entre muitos portais na Web dedicados ao desenvolvimento colaborativo de software, destaca-se o GitHub: um sistema web para repositórios de software, baseado no sistema de controle de versões **git**. Lançado em 2008, o GitHub possui mais de 40 milhões de usuários (março de 2020)[1] e é um dos portais do gênero mais relevantes, ocupando a 65^a posição (março de 2020) entre os sites mais populares do mundo.[2] Como parte de seu sistema de desenvolvimento de software colaborativo, o GitHub também oferece funcionalidades de gestão de projetos, *wiki* e rede social. Assim, devido à sua influência crescente, o GitHub tem sido foco de muitos trabalhos [Dabbish et al. 2012, Lima et al. 2014, Batista et al. 2017, El Asri et al. 2017], sendo que a maior parte deles está dedicada a entender a estrutura e dinâmica da colaboração, a formação de lideranças, e a influência de certos usuários na rede de colaboração.

Neste trabalho, focamos no GitHub para analisar o desenvolvimento colaborativo de software, aplicando conceitos biológicos que podem agregar valor às análises de interesse. Desse modo, propomos um modelo para a construção de ecossistemas de colaboração que utilizamos para caracterizar o GitHub. Esses ecossistemas diferenciam-se das abordagens frequentemente adotadas na literatura, uma vez que são focados no colaborador e conferem diversidade, incorporando uma série de linguagens e repositórios. Além disso, esses ecossistemas permitem reconstruir o processo colaborativo indivíduo a indivíduo em uma estrutura de rede temporal.

É importante ressaltar que a metodologia apresentada neste artigo não tem como objetivo substituir ou competir com as demais metodologias propostas na literatura, que visam entender o surgimento e a evolução do processo de colaboração no GitHub. O nosso objetivo é introduzir uma nova visão, que complemente a complexa tarefa de entender e explicar o fenômeno de colaboração nesta rede de desenvolvimento de software.

Nossas análises buscam respostas para as seguintes questões de pesquisa (QP):

- **QP1** - Quais são as principais características dos ecossistemas de colaboração no GitHub?
- **QP2** - Existem semelhanças entre os ecossistemas modelados a partir de diferentes repositórios?

Através da modelagem do sistema GitHub utilizando conceitos de ecossistemas, apresentamos uma nova visão de como entender a diversidade do processo colaborativo de desenvolvimento de software. Nossas análises sugerem que a modelagem proposta é útil para que se possa aprofundar os estudos em sistemas de colaboração a partir de teorias

[1]Uma busca por usuários em 05/03/2020 retornou 41.140.758 usuários ativos. A busca pode ser realizada em https://github.com/search?q=type:user&type=Users

[2]O *ranking* pode ser consultado em http://www.alexa.com/siteinfo/github.com

biológicas para, por exemplo, compreender melhor o uso de estratégias de colaboração, e consequentemente poder atuar sobre elas para que os processos colaborativos sejam ainda mais eficientes.

Este artigo está organizado da seguinte forma: a Seção 2 descreve os principais trabalhos relacionados; a Seção 3 apresenta a definição de ecossistemas de colaboração, bem como a metodologia utilizada para modelar esses ecossistemas no GitHub; a Seção 4 apresenta os resultados da caracterização dos ecossistemas de colaboração no GitHub; as implicações deste trabalho e trabalhos futuros são apresentados na Seção 5.

2. Trabalhos Relacionados

O desenvolvimento colaborativo de software tem sido objeto de estudo de muitos pesquisadores. A maior parte desses trabalhos focam no GitHub, buscando entender como as estruturas sociais afetam a colaboração. Por exemplo, o trabalho de [Lima et al. 2014] realiza uma caracterização do GitHub, enquanto rede social e rede de colaboração. Esse trabalho comparou a s distribuições de grau entre as redes construídas. Em todas as redes, a distribuição de grau seguia o fenômeno *scale-free*. A partir das análises, verificou-se também que a colaboração entre usuários acontece em pequenos projetos. Um dos principais resultados desse trabalho é que existe uma tendência de maior colaboração intra-países.

Para entender a força da colaboração no GitHub, os autores em [Batista et al. 2017] modelaram a colaboração em uma rede, em que os vértices são os colaboradores e há uma aresta sempre que dois colaboradores realizaram commit em um mesmo repositório. O trabalho selecionou todos os repositórios *non-forked* da linguagem JavaScript. Os autores coletaram métricas clássicas de redes complexas e compararam com outras três novas métricas propostas para medir a força da colaboração. Em um outro trabalho [Batista et al. 2018], os mesmos autores analisaram as redes sob aspectos temporais de repositórios da linguagem Java, JavaScript e Ruby. Nesse trabalho, os autores demonstram a relevância da análise temporal, para evitar a formação de cliques e compreensões equivocadas do processo colaborativo.

Em [Baudry and Monperrus 2012], os autores discorrem acerca de conceitos biológicos como ecossistema, biodiversidade e redes ecológicas, e traçam possíveis aplicações desses conceitos na Engenharia de Software com o objetivo de fornecer soluções para os desafios de construir softwares abertos em larga escala. Assim, os autores argumentam, por exemplo, que a noção de biodiversidade pode ser aplicada ao software (em nível de código) para construir sistemas mais adaptáveis e estáveis, uma vez que biodiversidade traz estabilidade para sistemas biológicos. Um outro trabalho muito parecido foi publicado por [Mens and Grosjean 2015]. Os autores também buscam conceitos biológicos para aplicação na Engenharia de Software, como o de ecossistemas de software, definido pelos autores como "uma coleção de projetos de software que são desenvolvidos e evoluem junto em um mesmo ambiente".

Nem todos os trabalhos que estudam as redes de desenvolvimento colaborativo de software utilizam o conceito de commit como unidade básica da colaboração. Muitos focam nos aspectos sociais, que estão relacionados a funcionalidades como *following*, *star* e comentários. Neste trabalho, nós modelamos ecossistemas de colaboração, a partir do conceito de commit. Nós acreditamos que como o commit é o resultado da colaboração,

essa abordagem permite trazer informações diferentes daquelas encontradas analisando estrelas e comentários.

Os poucos trabalhos que utilizaram redes modeladas a partir do conceito de commits não levaram em consideração a forma temporal e linear da colaboração, ou quando o fizeram, focaram em repositórios de apenas uma única linguagem. O modelo que propomos neste trabalho busca reproduzir o conceito biológico de ecossistema, e portanto, permite reproduzir a colaboração de forma temporal incorporando uma diversidade de linguagens de programação. Dessa forma, acreditamos que é possível entender a evolução dos ecossistemas de forma mais fiel à realidade, pois permite estudar todas as colaborações que um desenvolvedor executou ao longo do tempo, em diferentes repositórios simultaneamente, e como as relações de colaboração podem afetar o ecossistema como um todo.

3. Ecossistemas de colaboração

Nesta seção definimos ecossistemas de colaboração (Seção 3.1) e apresentamos a metodologia utilizada para modelar ecossistemas de colaboração no GitHub (Seção 3.2).

3.1. Definição

O termo ecossistema foi mencionado pela primeira vez por [Tansley 1935]. Com esse termo, [Tansley 1935] defendeu que não apenas os seres vivos, mas também todos os fatores que possam influenciar o bioma fossem levados em consideração no estudo das "unidades básicas da natureza". Com a inclusão dos fatores abióticos poderia-se observar os efeitos que eles causam sobre o bioma e vice-versa. Nessa definição, [Tansley 1935] também adicionou que ecossistemas são "uma reconhecida entidade auto-contida".

[Lindeman 1942] mostrou a importância de se estudar o ecossistema dinamicamente, devido a existência de fluxos de energia e de *loops* de retroalimentação que fluem tanto entre os fatores bióticos, quanto entre os fatores bióticos e abióticos. Nesse sentido, ecossistemas foram caracterizados pela capacidade de sofrer perturbações e se recuperar delas. Isso deixa claro que a ideia de ecossistema armazena a história dos fatores bióticos e abióticos.

Como resultado da reunião do bioma com o abioma, ecossistemas são caracterizados por uma ampla diversidade. Tanto os fatores bióticos quanto abióticos apresentam características variantes entre si. Na biologia, ecossistemas variam entre desertos, florestas, oceanos, entre outros. Os fatores bióticos são adaptados para viver em cada um desses tipos de ecossistema.

Neste artigo, nos referimos a ecossistema de colaboração como um ecossistema modelado com foco na colaboração, isto é, um ecossistema que ao incluir os fatores bióticos e abióticos deixe evidente os relacionamentos colaborativos entre eles. Assim, um ecossistema de colaboração se caracteriza por uma visão relativamente fechada no tempo de indivíduos colaborando entre si e o conjunto de meios e fatores envolvidos na colaboração. No GitHub, os fatores bióticos são representados pelos desenvolvedores, enquanto que os fatores abióticos são características do processo, como o repositório e a linguagem de programação utilizada.

Essa abordagem traz muitos benefícios: (1) como é capaz de reproduzir os relacionamentos colaborativos do indivíduo, pode-se enxergar de forma mais sistemática o pro-

cesso colaborativo, isto é, como as partes que compõem esses sistemas influenciam umas às outras; (2) a ideia de ecossistema introduz a diversidade de "habitats" de colaboração que existe; (3) é possível examinar a influência que fatores abióticos possuem no processo colaborativo; (4) propostas de novos modelos de estratégias de colaboração, baseadas, por exemplo, em reputação dos colaboradores participantes dos ecossistemas.

3.2. Ecossistemas no GitHub

Para aplicar o conceito de ecossistema de colaboração sobre o GitHub, utilizamos metadados obtidos a partir do projeto de código aberto GHTorrent [Gousios 2013], que tem por objetivo fornecer um espelho dos eventos ocorridos no GitHub. Este trabalho utilizou um *dump* de GHTorrent disponibilizado em 1 de abril de 2018[3].

Para construir o modelo de ecossistemas de colaboração no GitHub, selecionamos três repositórios a partir dos quais iniciamos a modelagem de três ecossistemas. Esses repositórios são denominados repositórios-raiz dos ecossistemas. Para análise neste artigo, selecionamos repositórios com diferentes tendências de crescimento no número total de colaboração, bem como de diferentes linguagens de programação. Além disso, os repositórios escolhidos têm números de commits diferentes, o que nos permite classificá-los em repositórios com alta, média e baixa intensidade de colaboração.

A criação de cada ecossistema segue o Algoritmo 1. Esse algoritmo foi elaborado com o intuito de reproduzir o conceito biológico de ecossistema, definido pelo conjunto de comunidades (colaboradores de um repositório) e outros fatores não vivos (como por exemplo, repositórios e linguagens de programação). Os dados aplicados como entrada no algoritmo foram pré-filtrados para considerar a colaboração durante 10 anos, compreendendo o início de 2008 até o fim de 2017.

Algoritmo 1: Cria ecossistema de colaboração do GitHub

Entrada: $RepositorioRaiz$
Saída: $Commits$ $(conjunto\ de\ commits)$
$Commits \longleftarrow \emptyset$
$Repositorios \longleftarrow RepositorioRaiz$
$DesenvolvedoresComputados \longleftarrow \emptyset$
repita
  $Desenvolvedores \longleftarrow RecuperaDesenvolvedores(Repositorios)$
  $Desenvolvedores \longleftarrow Desenvolvedores - DesenvolvedoresComputados$
  se $Desenvolvedores = \emptyset$ **então**
    retorna
  fim
  $c \longleftarrow RecuperaCommits(Desenvolvedores)$
  $Repositorios \longleftarrow RecuperaRepositoriosDosCommits(c)$
  $Commits \longleftarrow Commits \cup c$
  $DesenvolvedoresComputados \longleftarrow DesenvolvedoresComputados \cup Desenvolvedores$
até $Repositorios \neq \emptyset$

Para entender a execução do algoritmo, suponha que há um desenvolvedor no repositório raiz. Na primeira iteração do algoritmo, são recuperados todos os commits realizados pelo desenvolvedor do repositório raiz durante toda sua vida. Isto é, se o desenvolvedor colaborou em três repositórios com 1 commit em cada, vamos armazenar os commits em um conjunto e os repositórios em outro. Na segunda iteração, recuperamos os desenvolvedores que colaboraram em cada um dos três repositórios da iteração anterior. Sabendo quem são os desenvolvedores, recuperamos novamente todos os commits

[3]Site do projeto: https://ghtorrent.org/

que eles realizaram ao longo da vida. Esses commits são unidos ao conjunto de commits, e os repositórios são adicionados ao conjunto de repositórios para serem analisados na próxima iteração. As iterações prosseguem até que não haja mais repositórios a serem recuperados, o que significa que o ecossistema está fechado. É possível alterar o algoritmo para executar um número desejado de iterações. Neste trabalho, utilizamos 3 iterações devido ao grande volume de dados. Utilizando 3 iterações, garantimos a recuperação da história completa dos desenvolvedores recuperados nas 2 iterações anteriores. Os ecossistemas analisados neste artigo foram criados a partir dos repositórios-raiz TensorFlow[4], Spring[5] e SignalR[6], de linguagens de programação C++, Java e C#, respectivamente.

4. Caracterização dos ecossistemas de colaboração

Os ecossistemas TensorFlow, Spring e SignalR foram caracterizados quanto a distribuição de commits, desenvolvedores, linguagens e *forks*. A Tabela 1 apresenta as principais características dos ecossistemas analisados.

	TensorFlow	**Spring**	**SignalR**
# Commits	16.500.170	4.171.663	1.550.549
# Repositórios	52.585	14.559	5.895
# Desenvolvedores	1.203.348	397.163	178.770
# Linguagens	153	65	65

Tabela 1. Sumário dos ecossistemas de colaboração analisados

4.1. Commits

Entre os ecossistemas analisados, o maior número de commits foi encontrado no ecossistema TensorFlow, com cerca de 16,5 milhões de commits, seguido de Spring com pouco menos de 4,2 milhões de commits e SignalR com aproximadamente 1,5 milhão de commits. O número de commits é um bom indicador do nível de colaboração de cada ecossistema, dado que o commit é o resultado da colaboração no GitHub.

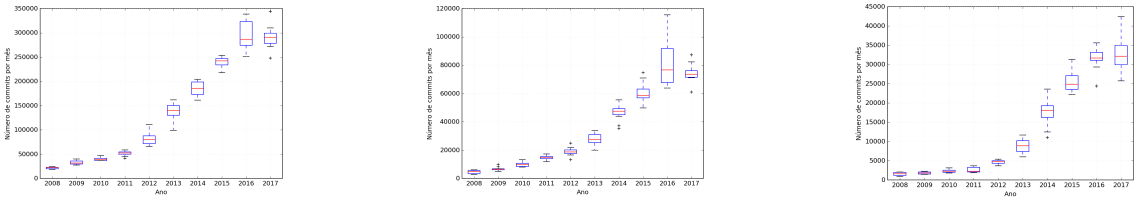
A Figura 1 mostra a distribuição de commits por mês em cada um dos ecossistemas em um período de 10 anos completos. A colaboração em cada ano foi sumarizada por meio de boxplots. Em todos os ecossistemas, é possível observar o crescimento gradual da colaboração, acentuando-se a partir do ano de 2012 no ecossistema TensorFlow e em 2013 nos demais. Nos primeiros anos, o número de commits variou pouco durante os meses do ano, passando a variar mais nos anos em que o número de commits foi maior. No último ano analisado, a produção caiu nos ecossistemas TensorFlow e Spring – única vez em toda a série.

A maior parte dos repositórios que compõe os ecossistemas têm poucos commits, como pode ser visto na Figura 2a. A distribuição cumulativa é muito concentrada, com metade dos repositórios tendo até 6 commits nos ecossistemas TensorFlow e Spring, e 3 commits no ecossistema SignalR. Somente cerca de 10% dos repositórios tem mais de 100 commits. O número máximo de commits que um repositório teve nos ecossistemas TensorFlow, Spring e SignalR foi, respectivamente, 234.590, 184.133, e 175.242.

[4]https://github.com/tensorflow/tensorflow

[5]https://github.com/spring-projects/spring-framework

[6]https://github.com/SignalR/SignalR



(a) Ecossistema TensorFlow **(b) Ecossistema Spring** **(c) Ecossistema SignalR**

Figura 1. Evolução dos commits ao longo do tempo

O baixo número de commits na maior parte dos repositórios conduz à investigação acerca do tempo em que os repositórios receberam seu último commit. A Figura 2b mostra a proporção de repositórios que receberam commits nos últimos x meses. Cerca de 63% dos repositórios estiveram ativos (recebendo commits) nos últimos 24 meses analisados (jan/2016 a dez/2017) para o ecossistema TensorFlow. No ecossistema Spring esse número cai para aproximadamente 57%. A maior taxa de repositórios ativos no período, total de 69%, foi observada no ecossistema SignalR.

O tempo de vida dos repositórios, isto é, a diferença entre a data de criação e o último commit, também foi medido para todos os ecossistemas. A Figura 2c apresenta esses resultados. A mediana do tempo de vida dos repositórios é de 9 dias nos ecossistemas TensorFlow e Spring e de 2 dias no ecossistema SignalR. Apenas 13% dos repositórios tem duração maior que 1 ano no ecossistema TensorFlow. Essa taxa é de 11% no ecossistema Spring e de 16% em SignalR.

4.2. *Forks*

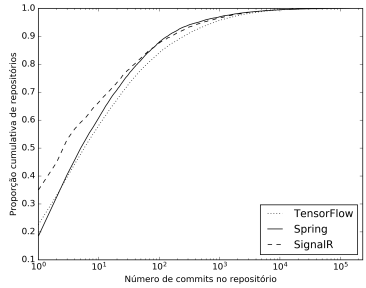
Desenvolvedores sem direito de commit no repositório devem criar um *fork*, realizar as alterações necessárias e solicitar um *pull request* para que a colaboração seja efetivada no repositório de origem. No ecossistema TensorFlow, 39% dos repositórios têm pelo menos um *fork*. Essa porcentagem é similar no ecossistema Spring, com *forks* ocorrendo em 32% dos repositórios. Diferentemente, o ecossistema SignalR apresenta mais da metade (55%) dos seus repositórios com pelo menos um *fork*.

A Figura 2d apresenta a distribuição cumulativa do número de *forks* por repositório, para cada um dos ecossistemas. Em todos os ecossistemas, a maior parte dos repositórios possui apenas um *fork*: apenas 13,7% dos repositórios do ecossistema TensorFlow possuem dois ou mais *forks*; essa porcentagem é de cerca de 20,6% no ecossistema Spring e de 4,3% em SignalR.

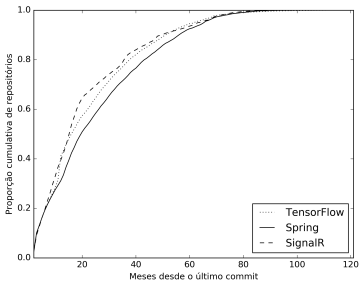
4.3. Desenvolvedores

Durante o período de 10 anos analisados, 1.203.348 desenvolvedores únicos colaboraram no ecossistema TensorFlow, 397.163 no ecossistema Spring, e 178.770 em SignalR. A Figura 3 exibe a variação do número de desenvolvedores realizando commits por mês nos ecossistemas. Houve um crescimento gradual do número de desenvolvedores colaborando, para todos os anos analisados, exceto no ano de 2017. A variação de desenvolvedores durante os meses do ano foi baixa até o ano de 2014, tornando-se maior a partir de então.

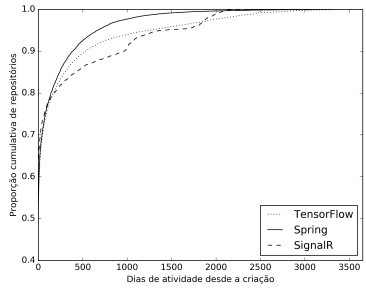
Essa análise não diferencia novos desenvolvedores daqueles que já estavam colaborando nos ecossistemas. A Figura 4 compara ano a ano o número de novos desen-



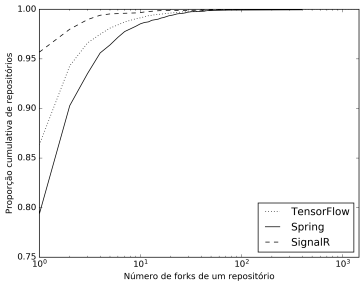
(a) Proporção cumulativa de repositórios com um dado número de commits. A maioria dos repositórios tem poucos commits.



(b) Proporção cumulativa de repositórios ativos durante os últimos *x* meses desde 1° de janeiro de 2008.



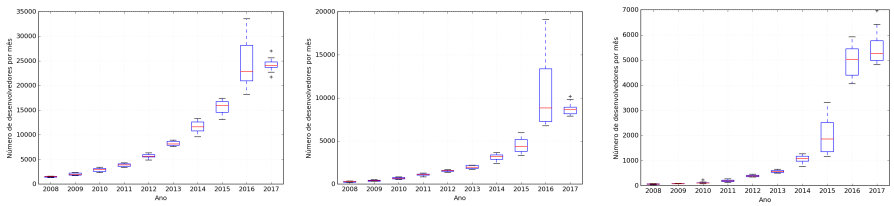
(c) Proporção cumulativa de repositórios que tiveram atividade nos últimos *x* dias desde a sua criação.



(d) Proporção cumulativa de repositórios com um determinado número de *forks*. Inclui repositórios com pelo menos um *fork*.

Figura 2. Visão geral da colaboração nos repositórios

volvedores com o número de desenvolvedores que já colaboravam nos ecossistemas. No ecossistema TensorFlow, o número de novos desenvolvedores nunca superou o número de desenvolvedores antigos em nenhum ano. O maior valor registrado para a relação entre novos e antigos desenvolvedores foi 0,55 no ano de 2016. Diferentemente, os ecossistemas Spring e SignalR tiveram mais participação de novos desenvolvedores do que de desenvolvedores antigos na maior parte dos anos observados. Novamente, foi no ano de 2016 que ocorreu o maior valor para essa relação, sendo de 1,19 no ecossistema Spring e 1,46 em SignalR.

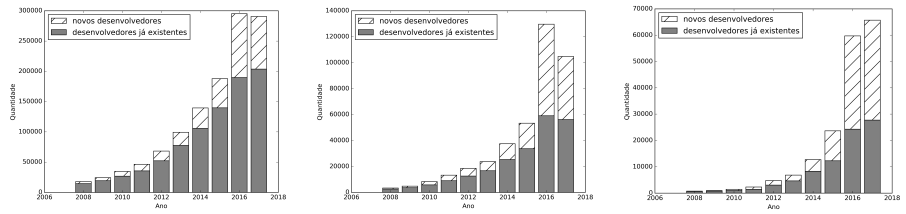


(a) Ecossistema TensorFlow **(b) Ecossistema Spring** **(c) Ecossistema SignalR**

Figura 3. Evolução do número de desenvolvedores nos ecossistemas

4.4. Linguagens de programação

Os ecossistemas possuem repositórios com linguagens principais muito diversas. O ecossistema TensorFlow, por exemplo, possui repositórios de 153 linguagens diferentes. Esse



(a) Ecossistema TensorFlow **(b) Ecossistema Spring** **(c) Ecossistema SignalR**

Figura 4. Entrada de novos desenvolvedores nos ecossistemas em comparação com os desenvolvedores já existentes

número é quase 3 vezes menor nos ecossistemas Spring e SignalR (65), mas ainda preserva a diversidade de linguagens.

Realizamos duas análises com relação às linguagens de programação: quantos repositórios utilizavam cada linguagem de programação (Tabela 2) e quantos commits são feitos em cada linguagem (Tabela 3). O termo "Indefinido" sumariza o total de repositórios ou commits em que não há informação de linguagem nos dados disponibilizados por GHTorrent.

Em todos os ecossistemas, muitas linguagens que aparecem entre as dez primeiras em um dos *rankings* também aparece em outro. No entanto, o posicionamento não se mantém. Calculamos a correlação de Spearman (r_s) entre o *ranking* de linguagens por repositório e o *ranking* de linguagens por commit. Para o ecossistema TensorFlow, r_s = 0,742, o que demonstra uma correlação alta; no entanto, para o ecossistema Spring (r_s = 0,558) e SignalR (r_s = 0,153) as correlações são insignificantes (para α = 0,05; tamanho da amostra = 10; r_c = 0,648).

Outra característica observada é que a linguagem do repositório-raiz dos ecossistemas nem sempre é a primeira linguagem nos posicionamentos, mas está pelo menos entre as três primeiras.

	TensorFlow		**Spring**		**SignalR**	
Posição	**Linguagem**	**# repositórios**	**Linguagem**	**# repositórios**	**Linguagem**	**# repositórios**
1	Python	8.643	Java	6.251	JavaScript	1.970
2	JavaScript	4.810	JavaScript	1.289	C#	1.097
3	C++	3.915	Shell	819	CoffeScript	208
4	Java	3.389	Ruby	427	Python	171
5	C	2.741	Groovy	303	HTML	156
6	Go	1.635	CSS	292	Java	127
7	Ruby	1.630	HTML	222	CSS	123
8	HTML	1.210	Python	218	Ruby	118
9	Julia	1.067	C	179	C++	100
10	Shell	1.034	Scala	167	Shell	72
#	Outras	8.953	Outras	862	Outras	535
#	Indefinido	13.558	Indefinido	3.530	Indefinido	1.218

Tabela 2. 10 linguagens de programação com o maior número de repositórios

4.5. Discussão dos Resultados

Nossas análises buscaram caracterizar diferentes facetas dos ecossistemas de colaboração, de modo a entender se o modelo proposto pode, de fato, ser utilizado para uma melhor

	TensorFlow		Spring		SignalR	
Posição	**Linguagem**	**# commits**	**Linguagem**	**# commits**	**Linguagem**	**# commits**
1	C	2.371.050	Java	1.343.290	C#	469.123
2	C++	2.326.058	Ruby	372.591	Shell	187.890
3	Python	2.284.208	JavaScript	280.977	Java	161.039
4	Java	1.544.964	Python	267.873	Python	125.606
5	Ruby	1.026.953	Shell	211.124	JavaScript	91.241
6	JavaScript	799.907	C	195.007	PHP	73.534
7	Go	602.837	Go	184.524	Go	59.461
8	Shell	417.760	C++	149.138	Ruby	56.820
9	HTML	274.198	Scala	108.017	C++	35.621
10	Scala	250.390	Nix	94.337	Scala	33.596
#	Outras	1.844.603	Outras	246.128	Outras	180.824
#	Indefinido	3.033.735	Indefinido	787.208	Indefinido	107.787

Tabela 3. 10 linguagens de programação com o maior número de commits

compreensão do fenômeno colaborativo no GitHub. A partir da caracterização dos três diferentes ecossistemas, concluímos que:

A intensidade da colaboração aumenta em todos os ecossistemas, porém com taxas diferentes. Houve crescimento gradual do número de commits ao longo dos anos, porém a variação ocorreu de forma diferente em cada ecossistema. O ecossistema TensorFlow sofreu uma evolução mais rápida da colaboração, se comparado com os demais. Algumas explicações possíveis estão relacionadas ao número de desenvolvedores, que é o maior entre os ecossistemas; ou até mesmo o empenho de alguns desenvolvedores ser maior que o empenho de outros.

A maior parte dos repositórios possuem poucos commits. A ocorrência de poucos repositórios com mais de 100 commits foi uma característica comum a todos os ecossistemas, o que pode ter relação com a estrutura oferecida pelo GitHub, uma vez que colaboradores que não são membros-efetivos de repositórios necessitam criar um *fork* do mesmo antes de colaborar.

Os ecossistemas estão "vivos". Nossas análises mostraram que grande parte dos repositórios que compõem os ecossistemas estão ativos e recebendo commits. A porcentagem de repositórios ativos nos últimos 24 meses analisados é similar entre os ecossistemas, fortalecendo a ideia de que o comportamento evolutivo deles é semelhante. A porcentagem de repositórios com tempo de vida maior que 1 ano também é similar entre os ecossistemas. Em TensorFlow, porém, o tempo de vida dos repositórios é maior, o que poderia novamente sugerir mais empenho dos desenvolvedores desse ecossistema. Porém, novas análises são necessárias para entender o que provocou essa diferença.

A intensidade da colaboração, considerando a métrica número de commits, aumenta à medida que novos desenvolvedores participam do ecossistema. Apesar de o número de commits aumentar com a entrada de novos colaboradores, dando indícios de que novos colaboradores aumentam a "força de trabalho", no caso de TensorFlow, o total de desenvolvedores novos é sempre menor do que os desenvolvedores que já fazem parte dos ecossistemas. Estabelecemos duas hipóteses para explicar o fenômeno: (1) novos colaboradores são mais ativos, ou (2) existem colaboradores fieis, que sustentam a

colaboração.

Os ecossistemas são diversos em linguagens de programação. Os três ecossistemas possuem um número grande de repositórios com linguagens de programação diferentes. TensorFlow é 3 vezes mais diverso que os demais, o que pode ser uma característica específica dos repositórios que o compõe. Dependendo do propósito do software, é comum combinar várias linguagens, o que torna seus desenvolvedores mais abertos quanto ao uso de linguagens de programação. É relevante destacar que existe diversidade inclusive entre os ecossistemas: existem linguagens nos rankings que construímos que não são comuns a todos os ecossistemas.

***Forks* são muito utilizados.** *Forks* correspondem a mais da metade (55%) dos repositórios que compõe o ecossistema SignalR, o que indica que o processo de colaboração nesse ecossistema é realizado mais frequentemente por membros não-efetivos do que por membros efetivos. Como a evolução desse ecossistema é mais lenta do que os demais, pode-se ponderar se o mecanismo de colaboração por *forks* gera resultados positivos ou negativos para o processo de colaboração. Repositórios que utilizam *forks* estão interessados em contribuições de qualidade, pois cada uma das contribuições será avaliada antes de ser efetivada. No entanto, precisamos de mais análises para verificar se esse controle de qualidade pode inibir a colaboração entre desenvolvedores.

Esses resultados mostram que o modelo proposto foi capaz de revelar a diversidade e a complexidade da colaboração no GitHub, e constituem o primeiro passo para utilizar ecossistemas de colaboração para estudar sob a ótica de modelos biológicos como ocorre o desenvolvimento colaborativo de software no GitHub. A vantagem dessa abordagem é a possível aplicação das teorias biológicas sobre sistemas de colaboração, como por exemplo, modelos de reciprocidade direcionada e modelos de benefícios por produtos [Barclay and Raihani 2016], que requerem diversidade e aspectos temporais e fornecem um melhor entendimento de estratégias de colaboração. O entendimento dessas estratégias é fundamental para que se possa atuar sobre esses ecossistemas, com o intuito de promover a persistência da colaboração, disseminar conhecimento, e consequentemente promover softwares de melhor qualidade.

5. Conclusão

Ao introduzirmos o conceito de ecossistema buscamos compreender, sob uma nova ótica, a dinâmica do processo de colaboração no GitHub. Como principal diferença entre os demais trabalhos na literatura, definimos um "sistema" fechado, que armazena a história dos fatores bióticos (desenvolvedores) e abióticos (repositórios e linguagens de programação). Uma das características principais é a presença da diversidade em diferentes aspectos, como por exemplo, linguagens de programação e repositórios de diferentes propósitos.

Como trabalhos futuros, pretendemos aplicar modelos e estratégias de colaboração oriundas de teorias biológicas para o melhor entendimento dos processos de colaboração que ocorrem nos ecossistemas caracterizados neste artigo.

Referências

Barclay, P. and Raihani, N. (2016). Partner choice versus punishment in human prisoner's dilemmas. *Evolution and Human Behavior*, 37(4):263 – 271.

Batista, N. A., Brandão, M. A., Alves, G. B., da Silva, A. P. C., and Moro, M. M. (2017). Collaboration strength metrics and analyses on github. In *Proceedings of the International Conference on Web Intelligence*, pages 170–178. ACM.

Batista, N. A., Sousa, G. A., Brandão, M. A., da Silva, A. P. C., and Moro, M. M. (2018). Tie strength metrics to rank pairs of developers from github. *Journal of Information and Data Management*, 9(1):69–69.

Baudry, B. and Monperrus, M. (2012). Towards ecology inspired software engineering. *arXiv preprint arXiv:1205.1102*.

Dabbish, L., Stuart, C., Tsay, J., and Herbsleb, J. (2012). Social coding in github: Transparency and collaboration in an open software repository. In *Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work*, CSCW '12, pages 1277–1286, New York, NY, USA. ACM.

El Asri, I., Kerzazi, N., Benhiba, L., and Janati, M. (2017). From periphery to core: a temporal analysis of github contributors' collaboration network. In *Working Conference on Virtual Enterprises*, pages 217–229. Springer.

Fehr, E. and Schurtenberger, I. (2018). Normative foundations of human cooperation. *Nature Human Behaviour*, 2(7):458–468.

Gousios, G. (2013). The GHTorrent dataset and tool suite. In *Proceedings of the 10th Working Conference on Mining Software Repositories*, MSR '13, pages 233–236.

Gracia-Lázaro, C., Ferrer, A., Ruiz, G., Tarancón, A., Cuesta, J. A., Sánchez, A., and Moreno, Y. (2012). Heterogeneous networks do not promote cooperation when humans play a prisoner's dilemma. *Proceedings of the National Academy of Sciences*, 109(32):12922–12926.

Hamilton, W. D. (1963). The genetical evolution of social behavior. *Journal of Theoretical Biology*, 7:1–16.

Lima, A., Rossi, L., and Musolesi, M. (2014). Coding together at scale: Github as a collaborative social network. In *Proceedings of 8th AAAI International Conference on Weblogs and Social Media (ICWSM 2014)*.

Lindeman, R. L. (1942). The trophic-dynamic aspect of ecology. *Ecology*, 23(4):399–417.

Lopes, H., Santos, A. C., and Teles, N. (2009). The motives for cooperation in work organizations. *Journal of Institutional Economics*, 5(3):315–338.

Mens, T. and Grosjean, P. (2015). The ecology of software ecosystems. *Computer*, 48:85–87.

Nowak, M. A. and Sigmund, K. (1998). Evolution of indirect reciprocity by image scoring. *Nature*, 393(6685):573.

Rand, D. G., Arbesman, S., and Christakis, N. A. (2011). Dynamic social networks promote cooperation in experiments with humans. *Proceedings of the National Academy of Sciences*, 108(48):19193–19198.

Tansley, A. G. (1935). The use and abuse of vegetational concepts and terms. *Ecology*, 16(3):284–307.