



# "We are dying!" On Death Signals of Software Ecosystems

Felipe Soupinski  
FACOM/UFMS  
Campo Grande, MS, Brasil  
felipe\_s@ufms.br

Igor Wiese  
UTFPR  
Campo Mourão, PR, Brasil  
igor@utfpr.edu.br

Pedro Arantes  
FACOM/UFMS  
Campo Grande, MS, Brasil  
pedro.arantes@ufms.br

Hudson Borges  
FACOM/UFMS  
Campo Grande, MS, Brasil  
hudson.borges@ufms.br

Igor Steinmacher  
UTFPR  
Campo Mourão, PR, Brasil  
igorfs@utfpr.edu.br

Bruno B.P. Cafeo  
FACOM/UFMS  
Campo Grande, MS, Brasil  
bruno.cafeo@ufms.br

Awdren Fontão  
FACOM/UFMS  
Campo Grande, MS, Brasil  
awdren.fontao@ufms.br

## ABSTRACT

Software Ecosystems (SECO) depend on platforms that serve as environments for developers interaction. When the organization that owns the platform does not support the synergy between organizational goals and developers' expectations, the SECO may die. The death results in the definitive suspension of vital activities, directly impacting developers, who lose part of the learning and experience gained. Projects that depend on the SECO platform may be discontinued. Ultimately, the responsible corporation loses the resources invested in the development and maintenance of the software and communities. Thus, understanding the signs that may indicate a SECO death is important. In this paper, we report a study conducted on StackOverflow (SO) and GitHub (GH) focusing on Web SECOs: AngularJS, PhantomJS, and MomentJS. We analyzed metrics based on answer rates, closed issues, and modified files to understand what happens in these SECOs before, during, and after the platform's death. We identified that an SO answer rate below 0.8 is a sign of risk. In GH, the files that involve source code and text are the ones that concentrate on the activities before and during death.

## ACM Reference Format:

Felipe Soupinski, Pedro Arantes, Igor Steinmacher, Igor Wiese, Hudson Borges, Bruno B.P. Cafeo, and Awdren Fontão. 2022. "We are dying!" On Death Signals of Software Ecosystems. In *XXXVI Brazilian Symposium on Software Engineering (SBES 2022)*, October 5–7, 2022, Virtual Event, Brazil. ACM, New York, NY, USA, 7 pages. <https://doi.org/10.1145/3555228.3555264>

## 1 INTRODUÇÃO

A plataforma de um Ecossistema de Software (ECOS) depende de sua evolução contínua, pois é isso que determina a atratividade do ambiente para os usuários e comunidades de desenvolvedores [10, 11]. Além disso, os ambientes de ECOS são meios importantes

para a troca de recursos e informações, uma vez que um ECOS depende das interações entre seus membros [7].

Ambientes colaborativos, como plataformas de Q&A (StackOverflow) e de repositórios de projetos de software (GitHub), são frequentemente utilizados como locais de interação. Os desenvolvedores que fazem parte da comunidade criam ou evoluem (mantêm) os artefatos em torno da plataforma de software. Contudo, as organizações que mantêm essas plataformas lidam com desafios ao tentar atender as necessidades das comunidades.

Caso a organização detentora da plataforma não estabeleça estratégias sustentáveis, que não suportam a sinergia entre os objetivos organizacionais e as expectativas e experiências dos desenvolvedores, corre-se o risco do ECOS ter suas atividades vitais suspensas e morrer. Um sinal da morte de um ECOS é quando o software começa a se tornar estático [4, 11] ou passa a não ser mantido [3]. Uma plataforma de software também pode ser considerada morta quando ela é descontinuada, ou seja, quando a organização responsável pelo ECOS anuncia a interrupção das evoluções.

Em síntese, as demandas dos usuários esperam por uma constante evolução do software e quando isso não ocorre, as comunidades interrompem a sua interação com o ECOS. Essa queda de estímulos da plataforma pode ocasionar a evasão de membros, o que resulta na morte do ECOS. Com isso, os desenvolvedores podem perder parte do aprendizado e experiência conquistada [11]. Os projetos que dependiam da plataforma principal precisam se adaptar, isto quando não se tornam estáticos ou descontinuados por terem uma alta dependência da plataforma [3]. Por fim, a corporação responsável perde recursos aplicados no desenvolvimento e manutenção do software e da comunidade [4].

Observa-se na literatura que os estudos existentes focam em projetos não-mantidos [3], atitudes mortais [5] e morte do Windows Phone numa análise do StackOverflow [11]. Para complementar o estado-da-arte, o objetivo deste estudo é investigar indicadores em torno da morte de ECOS para apoiar o entendimento deste fenômeno no cenário de Engenharia de Software. Nesse estudo, focou-se em ECOS direcionados para ferramentas de desenvolvimento Web. Para guiar o estudo, foram definidas três questões de pesquisa, a saber:

ACM acknowledges that this contribution was authored or co-authored by an employee, contractor or affiliate of a national government. As such, the Government retains a nonexclusive, royalty-free right to publish or reproduce this article, or to allow others to do so, for Government purposes only.

SBES 2022, October 5–7, 2022, Virtual Event, Brazil

© 2022 Association for Computing Machinery.

ACM ISBN 978-1-4503-9735-3/22/10...\$15.00

<https://doi.org/10.1145/3555228.3555264>

- Qual o **ritmo da atividade** em relação à taxa de resposta no SO em torno da morte de um ECOS? Taxa de resposta abaixo de 0,8 é sinal de perigo. Um ritmo normal de resposta demonstra efetividade de estratégia de migração de recursos e comunidades. E não há como escapar de arritmia e morte;
- Qual é a **estratégia de sobrevivência** de desenvolvedores do ponto de vista do solicitação de mudanças? Há organização coletiva para sobrevivência. Comunicar sobre a morte do ECOS afeta as solicitações de mudanças. Próximo do ano da morte há um comportamento de repouso (uso de funções vitais). O ambiente do ECOS morto deve se tornar inabitável;
- Que tipos de artefatos fazem parte da **reciclagem de recursos** durante e após a morte do ECOS? Arquivos de código-fonte e texto são os mais modificados antes da morte do ECOS. Após a morte, não há quadro de evolução de código como forma de reciclar recursos.

## 2 MORTE DE ECOS

Um Ecossistema de Software (ECOS) consiste em um ambiente em torno de uma plataforma de software, onde atores interagem com artefatos e outros atores [10]. Nesse ambiente, os atores podem trocar informações e recursos entre si, além de consumir e produzir artefatos em torno das plataformas do ecossistema.

O StackOverflow (SO) e GitHub (GH) são exemplares desses ambientes que compõem os ECOS. O SO consiste de um ambiente de perguntas e respostas em torno de dúvidas técnicas das comunidades de desenvolvedores. Já o GH é um ambiente colaborativo que consiste de artefatos em torno do gerenciamento de mudanças, controle de versão e integração/implantação contínua de projetos de software. O SO e GH possuem métricas que permitem o monitoramento da atividade de ECOS.

A morte pode acontecer em ECOS que não haja mais atividade, o que leva o ecossistema a se tornar estático. Segundo a Ecologia [2], um ecossistema estático é um ecossistema morto. Para o caso de ECOS, sua morte ocorre quando (i) há um anúncio oficial dos seus mantenedores, corporação ou comunidade [11]; ou (ii) quando o software fica sem atividade pelo período de um ano [3].

Um ECOS possui alto custo de produção, uma vez que envolve um conjunto de plataformas tecnológicas, comunidades de desenvolvedores, comunidades de especialistas e recursos técnicos, por exemplo. Quando um ECOS morre, alguns desses recursos podem ser reaproveitados. A esse reaproveitamento, com base na Ecologia, para este artigo será usado o termo reciclagem de recursos.

## 3 TRABALHOS RELACIONADOS

Evertse *et al.* [5] analisam condutas, princípios e comportamentos adotados por ECOS que resultam, ou facilitam, na morte de seus projetos. Aponta-se as principais causas que resultam no desengajamento da comunidade e possíveis contramedidas para as mesmas. Portanto, os autores analisam atitudes "mortais" para ECOS enquanto o estudo a seguir identifica o comportamento da comunidade em torno do projeto.

Coelho *et al.* [3] analisam os riscos e vulnerabilidades de projetos *open source* que enfrentam dificuldades em manter sua atividade de desenvolvimento e manutenção constante por parte da comunidade. Sendo assim, é proposto um modelo de aprendizado de máquina

focado na classificação de repositórios sem atividade de manutenção. Portanto, os autores mineram repositórios de código para detecção de vulnerabilidades enquanto o estudo a seguir coleta os dados para análise da atividade do ECOS.

Massanori *et al.* [11] analisam a morte de ECOS na perspectiva da área de Developer Relations (DevRel). Utiliza-se para isso o repositório de perguntas e respostas StackOverflow e o ECOS Windows Phone. Portanto, os autores estudam o domínio de sistemas operacionais para aparelhos móveis com foco em um ambiente de Q&A enquanto o estudo a seguir investiga ECOS em torno de ferramentas de desenvolvimento Web observando sinais em diferentes ambientes.

## 4 METODOLOGIA DE PESQUISA

A metodologia utilizada segue as diretrizes para mineração de repositórios de software (MRS) propostas por Hemmati *et al.* [6].

### 4.1 Objetivo e Questões de pesquisa

O objetivo é analisar sinais da morte de um ECOS, com base nos seguintes indicadores: (1) ritmo de atividade, (2) estratégia de sobrevivência e (3) reciclagem de recursos, para apoiar o entendimento deste fenômeno no cenário da Engenharia de Software. Neste estudo, o foco está direcionado para ECOS em torno de plataformas de desenvolvimento para Web.

**QP1.** Qual o **ritmo da atividade** em relação a taxa de resposta no SO em torno da morte de um ECOS? **Motivo:** O SO favorece aos desenvolvedores a solução de dúvidas e evolução de contribuições. Investigar a taxa de respostas (razão entre o número de questões respondidas e o número total de questões realizadas em um mês) demonstra a interação entre questionador/respondente e um indício de uma questão que representa qualidade para a comunidade [1]. **Métrica:** Taxa de Resposta no SO.

**QP2.** Qual é a **estratégia de sobrevivência** de desenvolvedores do ponto de vista de solicitação de mudanças? **Motivo:** Mudanças em recursos em torno de um software podem estar relacionadas à mudanças no ambiente. Desta forma, as *issues* fechadas serão analisadas do ponto de vista temporal como a frequência de ações tomadas antes, durante e após a morte do ECOS. **Métrica:** *Issues* fechadas no GH.

**QP3.** Que tipos de artefatos fazem parte da **reciclagem de recursos** durante e após a morte do ECOS? **Motivo:** A reciclagem é importante pra minimizar o impacto da morte do ECOS e preservar/reusar recursos. Os recursos podem ser decompostos para serem reaproveitados. Assim, os investimentos no ECOS não são completamente perdidos **Métrica:** Tipos de artefatos modificados no GH.

### 4.2 Critérios de seleção de ECOS

Realizou-se uma busca exploratória no Google com palavras-chave "morte plataforma de software" e "descontinuação de plataforma de software", com suas traduções em inglês. E se verificou quais ECOS aceitavam a contribuição de desenvolvedores. O **primeiro critério** trata da morte do ECOS, que pode ser verificada por anúncios em portais oficiais da plataforma. Ou de acordo com Coelho *et al.* [3], um software é considerado descontinuado quando possui

período maior que um ano sem atividade. O **segundo critério** para seleção é a presença no SO (verificação por meio de *tag*) ou no GH (repositório público). O **terceiro critério** considera o tipo ECOS que, para este estudo, deve considerar ecossistemas em torno de ferramentas de solução Web.

Os ECOS selecionados são listados na Tabela 1, incluindo a datas de descontinuação, início e fim do período de suporte. Foram encontrados, ao todo, três ECOS (AngularJS, Moment.js e PhantomJS). AngularJS é um *framework* JavaScript para execução de *single-page applications* (SPA). O Moment.js é uma biblioteca JavaScript *open source* para trabalhar com datas. Já o PhantomJS é um navegador sem cabeça usado para automatizar a interação em páginas web. A Tabela 2 apresenta o motivo da descontinuação, se houve reciclagem de recursos e motivo da reciclagem.

**Tabela 1: Datas de lançamento, descontinuação e suporte**

ECOS	Lançamento	Descontinuação	Início Suporte	Fim Suporte
AngularJS	2009	01/2018	07/2018	12/2021
PhantomJS	01/2011	01/2016	01/2016	03/2018
Moment.js	2011	2020	2020	-

**Tabela 2: Reciclagem de recursos.**

Software	Motivo Descontinuação	Migrou?	Reciclagem	Motivo Reciclagem
AngularJS	Substituído pelo Angular	Sim	Angular	Refatorado no Angular
Moment.js	Concorrentes	Não	Não houve	Não houve
PhantomJS	Falta de contribuições	Sim	CasperJS, YSlow	Uso para automação de testes

### 4.3 Extração de Dados

O SO é uma relevante plataforma sobre desenvolvimento de software, recebendo mais de seis mil perguntas diariamente [8]. O GH foi selecionado por possuir as maiores comunidades de desenvolvedores *open source*. Além disso, ambos fornecem indicadores sobre a interação entre seus usuários.

Para a recuperação dos dados de cada repositório foram realizadas duas abordagens distintas. No SO, com o auxílio da ferramenta *Stack Exchange*,<sup>1</sup> buscou-se pelos ECOS, filtrando-os pelas suas respectivas tags, para no final exportá-los à etapa de modelagem. No GH, com apoio da biblioteca PyDriller,<sup>2</sup> foram extraídos dados sobre os ECOS.

<sup>1</sup><https://data.stackexchange.com/stackoverflow>

<sup>2</sup><https://github.com/ishepard/pydriller>

### 4.4 Modelagem de Dados e Síntese

**Para o SO**, a ferramenta Stack Exchange trabalha com a entrada de código SQL para busca de dados na plataforma. Após isso, o trabalho de modelagem se resumiu na substituição de valores nulos por zero e a normalização dos dados. Recuperou-se a taxa de resposta agrupada em períodos anuais, gerando assim uma série temporal. A taxa de resposta evidencia a qualidade de questões e envolvimento de questionadores/respondentes. Essa taxa é calculada como a razão do número de questões respondidas pelo número de questões não respondidas. **Para o GH**, foi recuperada a métrica de quantidade de issues fechadas que foram separadas por mês e por repositório. Além disso, os arquivos modificados dentro de cada projeto foram classificados de acordo com Ma *et al.* [9]. As categorias e extensões de arquivos são descritas na Tabela 3.

**Tabela 3: Atualizações das classes de arquivos**

Classe	Formatos
Aplicação	.bat .cmd .exe .ser .swf
Arquivo	.a .gz .jar .pack .zip
Áudio	.kt .mp3 .ogg .wav
Imagem de	.scl
Disco	
Fonte	.eot .otf .ttf .woff
Imagem	.blp .bmp .dds .gif .ico .jpeg .jpg .png .psd .rs .svg .tga .tif .xpm
Projeto	.csproj .pbxproj .vcproj .vcxproj
Código-fonte	.as .asm .c .cc .class .coffee .cpp .cs .cshtml .css .ctp .cxx .d .dll .ebuild .ejs .el .erb .erl .f .f90 .go .gradle .groovy .h .haml .hpp .hs .i .java .js .jsp .less .lua .m .mo .o .php .phpt .phtml .pl .pm .pp .py .pyc .r .rb .s .scala .scss .scss .sh .smali .so .sql .swift .t .tcl .ts .vb .vim .rkt .html .json .yaml .launch .template .ngdoc .gdoc .xml .md
Texto	

Foram aplicados então, gráficos temporais nas métricas *taxa de resposta* e *issues fechadas* utilizando intervalos anuais. Essas séries temporais foram essenciais para entender o comportamento das métricas antes, durante e depois da morte de um ECOS. Para consolidar a análise, foram realizadas reuniões com os pesquisadores envolvidos no estudo, na qual individualmente apontaram para possíveis pontos de melhoria.

### 5 QUAL O RITMO DA ATIVIDADE EM RELAÇÃO A TAXA DE RESPOSTA NO SO EM TORNO DA MORTE DE UM ECOS?

**Taxa de Resposta abaixo de 0,8 pode ser sinal de perigo.** A taxa de respostas pode demonstrar a relação entre a quantidade de perguntas respondidas e não respondidas mensalmente no ambiente pela comunidade.

Nota-se que um ECOS estável, no período anterior a sua morte, tende a alcançar uma taxa de resposta superior a 80%. Isto pode ser um indicio de sinal vital do ECOS. A Figura 1 apresenta séries

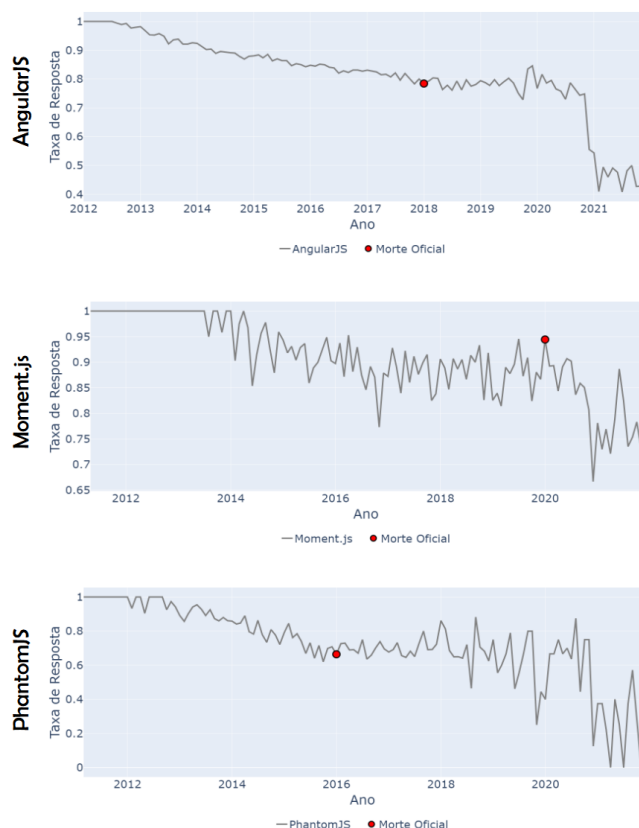


Figura 1: Taxa de resposta: AngularJS, PhantomJS e Moment.js

temporais a partir da taxa de resposta de cada ECOS. Pode-se notar uma tendência de se manter acima de 0,8 em períodos anteriores à morte, entretanto em descendente. Esse comportamento é comum entre os ECOS analisados. Contudo, a comparação com ECOS saudáveis é necessária para confirmar que taxa de resposta abaixo de 80% é um indicativo que um ECOS pode morrer.

**ECOS sem estratégia de migração demonstram arritmia na taxa de resposta.** Analisando o Moment.js que não possui migração de recursos para outro ECOS, é possível perceber que a série temporal gerada a partir da taxa de resposta se comporta como uma arritmia, ou seja, possui alterações no ritmo normal. A métrica a partir de três anos antes da morte, vai se aproximando da taxa de 0,8. Isto pode acontecer devido a algum agente patológico no ECOS, que necessita de investigação, como: qualidade da questão, ausência de especialistas, queda de engajamento e fuga da comunidade, perguntas desconectadas de necessidades do ECOS.

**Ritmo normal da taxa de resposta demonstra efetividade de estratégia de migração.** O AngularJS contou com comunicações, engajamento da comunidade e dos mantenedores do ECOS para uma migração para o Angular. No AngularJS, a organização elaborou um plano de migração e suporte para seus usuários, mantendo assim um alto número de comentários e respostas no SO. É possível

perceber pela taxa de resposta, que a estratégia foi efetiva, inclusive, como um tratamento preventivo para futura arritmia. Antes da morte (2018), a taxa de resposta já apresentava tendência para se aproximar do 0,8, mas esse decaimento é controlado, não gerando níveis de tensão entre questionador/respondente. Deve-se ressaltar que, um apoio maior das organizações dá credibilidade perante a comunidade.

**Não há como escapar da arritmia e morte.** O que é comum entre os ECOS é que não há como escapar de, na taxa de resposta, haver arritmia após a morte do ECOS. Considerando as devidas proporções, a arritmia fica mais acentuada. Percebe-se que no AngularJS, há quase um comportamento como "último suspiro" do ECOS, principalmente perto do final do período de suporte (2021), onde a taxa cai drasticamente. Neste caso, pode ser que a comunidade já tenha migrado de forma mais generalizada para o Angular. Ainda considerando os três ECOS, pode-se perceber a tensão que pode ser gerada por um consumo específico e excessivo de um determinado tema nas questões/respostas, naquele tempo específico. Como se ainda houvesse tentativa em recuperar algum recurso.

## 6 QUAL É A ESTRATÉGIA DE SOBREVIVÊNCIA DE DESENVOLVEDORES DO PONTO DE VISTA DA SOLICITAÇÃO DE MUDANÇAS?

**Organização coletiva para sobrevivência.** Nos ECOS analisados (Figura 2) é possível perceber que há um movimento dos membros do ecossistema para atuar no ciclo de solução de issues próximas a datas de releases. Ou seja, a atitude que acontece naturalmente em projetos de software não muda, na relação entre as métricas de issues fechadas e releases.

**A estratégia de sobrevivência baseada em comunicação sobre a morte afeta a atividade em torno das solicitações de mudança** - O período após a morte do ECOS apresenta comportamento diferente, isso se dá pelas diferentes políticas de gestão aplicadas para cada ECOS. Para o AngularJS, a quantidade de issues fechadas decaiu de forma drástica pelo menos um ano depois da morte, fato esse que pode ser ligado ao planejamento de migração, descontinuação e suporte aplicado ao projeto. Vale ressaltar que, o anúncio de descontinuação foi oficializado em 2016, dois anos antes da morte do ECOS. O que não acontece com o Moment.js e o PhantomJS que após a morte alcançam picos de issues fechadas.

Para o Moment.js, que teve sua morte oficializada sem anúncio prévio, a quantidade de issues fechadas teve alta ou até atingiu a sua máxima nesse período. Tal comportamento pode ser visto como uma tentativa da comunidade em finalizar todos os trabalhos referentes ao projeto.

Para o PhantomJS, a métrica issues fechadas apresentou um valor baixo desde momentos anteriores à morte do projeto. Vale ressaltar que quando há uma disparidade entre a quantidade de issues abertas e fechadas dentro de um repositório, pode ser um indicio de que as demandas das comunidades não estão sendo atendidas. Isso vai de encontro com a causa da morte do ECOS, ocasionada pela falta de contribuição de desenvolvedores.

**Alterar características do ECOS pede reaproveitamento de recursos.** É possível perceber que há atividade para resolver issues mesmo próximo da morte, no período de morte e após a morte. Em

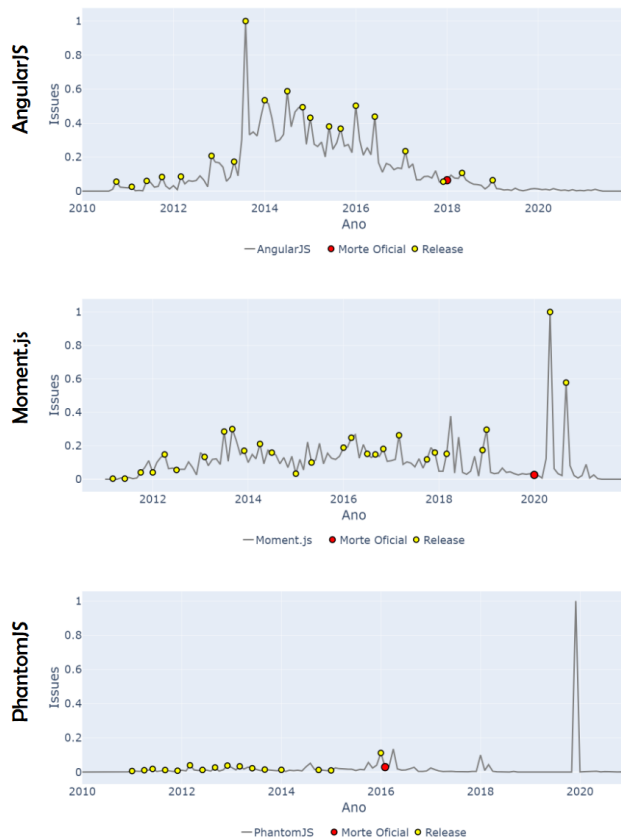


Figura 2: Issues fechadas no AngularJS, PhantomJS e Moment.js

uma analogia com a seleção natural, podem ser que os organismos (artefatos) que sejam mais aptos a serem úteis em um novo ambiente sejam selecionados. Essa seleção, dentro do ECOS, pode ser controlada por meio de issues que precisam ser resolvidas. Isso pode ter haver com o conceito de hereditariedade, ou seja, características vantajosas do ECOS são passadas para outros ECOS. Isto garante uma vantagem evolutiva como estratégia de sobrevivência a partir da morte de um ecossistema. O que é impulsionado por algum fluxo migratório.

**Situação de repouso - uso de funções vitais.** Pode-se perceber que a atividade próxima do ano da morte e logo após a morte apresenta um comportamento de baixa atividade para resolver issues. Esse comportamento apresenta uma estratégia de solução de issues que sejam vitais para o ECOS, ou prioritárias. Um estudo futuro pode investigar a severidade e conteúdo dessas issues para ver se essa hipótese de issue vital é confirmada.

**Tornar o ambiente do ECOS morto inabitável.** Especificamente no AngularJS que teve estratégia de migração para o Angular, pode-se analisar a queda de issues fechadas como a anulação do ambiente do ECOS Morto. Em analogia com ecologia, isso pode ser importante para que dois ambientes com características aproximadas podem dificultar a percepção dos desenvolvedores sobre os limites entre o

AngularJS (ECOS morto) e o ECOS (Angular) para onde ocorreu a migração. Organismos (desenvolvedores) do mesmo nicho podem não sobreviver no mesmo ambiente, pois competem pelos mesmos recursos e isso poderia por em risco o "novo" ECOS.

## 7 QUE TIPOS DE ARTEFATOS FAZEM PARTE DA RECICLAGEM DE RECURSOS DURANTE E APÓS A MORTE DO ECOS?

**Antes da morte arquivos de código sofrem mais modificações seguido por texto.** A dinâmica de trabalho e colaboração dos desenvolvedores são diferentes em cada ECOS, principalmente quando comparados os softwares PhantomJS, que possui desenvolvimento periódico da comunidade, e o AngularJS, que possui desenvolvimento contínuo da comunidade. Esse fato ocorre, principalmente, pela diferença entre os níveis de engajamento dos projetos.

**Após a morte do ECOS, o código-fonte não demonstra quadro de evolução na perspectiva de reciclagem.** Ao analisar o período, pode-se concluir que, a atividade de evolução e manutenção do código-fonte perde intensidade nesse intervalo. Ademais todos os projetos analisados apresentam quedas consideráveis na quantidade de arquivos modificados por ano. Além disso, a dinâmica de trabalho construída durante a vida do projeto parece influenciar o comportamento dos desenvolvedores nesse período. ECOS, como PhantomJS, que enfrentavam dificuldade na participação ativa da comunidade são os que possuem a menor atividade de desenvolvimento após a sua morte. Já ECOS, como o AngularJS, que possuíam um plano de migração e grande atividade de desenvolvimento, apresentam uma queda mais lenta na manutenção e evolução do seu código-fonte.

## 8 AMEAÇAS À VALIDADE

Em relação à validade de *construto*, a base teórica deste estudo considerou as fragilidades apontadas nos trabalhos relacionados. A escolha do SO e GH se deve à presença de desenvolvedores interagindo em torno dos ECOS. Para apoiar a validade interna, os conjuntos de dados não foram selecionados aleatoriamente, mas foram relacionados ao ecossistema estudado. Para reduzir o efeito da expectativa dos experimentadores, as análises do estudo seguiram os procedimentos indicados para estudos de MRS. Generalização - este estudo é focado em ferramentas de soluções Web, o estudo precisa ser replicado em outros ECOS. Efeito da expectativa do experimentador - foram realizadas reuniões de conciliação que envolveram a análise dos resultados.

## 9 CONCLUSÃO E FUTUROS TRABALHOS

Nesse artigo, avançou-se no estudo sobre o comportamento em torno da morte de um ECOS. Focou-se em ECOS relacionados a ferramentas de soluções para Web: AngularJS, PhantomJS e Moment.js. A partir das respostas às questões de pesquisa, chegou-se a algumas direções: **Ritmo de Atividade:** uma taxa de resposta abaixo de 0,8 pode ser sinal de perigo. ECOS que possuem estratégia para migrar suas comunidades e recursos para um novo demonstram um comportamento normal no ritmo. Mesmo que haja estratégia de migração, não há como escapar de arritmia e da morte. **Estratégia de**



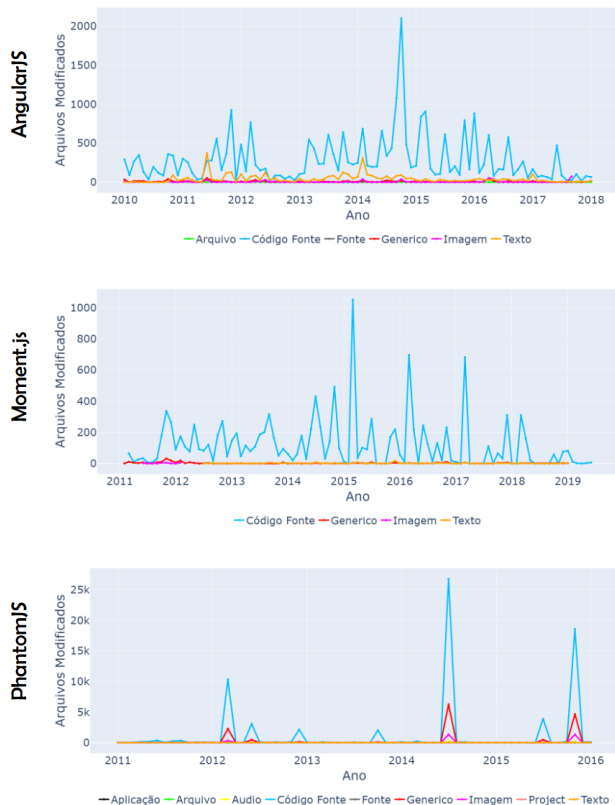


Figura 3: Arquivos editados no período de lançamento até a descontinuação.

**Sobrevivência:** há organização coletiva para sobreviver à morte. Dentre essa organização, aquelas que envolvem comunicação sobre a morte do ECOS afetam a atividade em torno de issues. Quanto mais se aproxima o ano da morte, há uma tendência a repouso, como se o ECOS tentasse manter suas funções vitais. Se houver migração (Morto para Novo ECOS) é necessário tornar o ambiente do ECOS morto inabitável para que não haja concorrência por recursos e gasto de energia para contribuições. **Reciclagem de Recursos:** arquivos de código-fonte e texto recebem mais manutenção. Porém, após a morte do ECOS, arquivos de código fonte não apresentam evolução em seu quadro. E, mesmo em ambientes com plano de migração, há uma queda (lenta) na manutenção e evolução.

As métricas de taxa de resposta e resolução de issues podem ser usadas para monitoramento da atividade "pré-morte" de um ECOS. As issues fechadas e tipos de artefatos modificados podem ser usadas para compreender a estratégia de sobrevivência e a reciclagem de recursos em torno da plataforma.

Implicações para indústria e academia: analisar séries temporais para identificar o ritmo de atividade com alertas de arritmia dentro das métricas analisadas; Investigar elementos patogênicos dentro do ECOS. Isto ajuda na criação de estratégias paliativas ou de restauração. Análise e exercício de estratégias para criar oportunidades



Figura 4: Arquivos editados mensalmente no período de início do suporte até o seu fim.

para regenerar recursos. Avanço num corpo de conhecimento sobre condições que levam um ECOS à morte.

## REFERÊNCIAS

- [1] Ashton Anderson, Daniel Huttenlocher, Jon Kleinberg, and Jure Leskovec. 2012. Discovering value from community activity on focused question answering sites: a case study of stack overflow. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. 850–858.
- [2] Michael Begon, Colin R Townsend, and John L Harper. 2009. *Ecologia: de indivíduos a ecossistemas*. Artmed editora.
- [3] Jailton Coelho, Marco Tulio Valente, Luciano Milen, and Luciana L Silva. 2020. Is this GitHub project maintained? Measuring the level of maintenance activity of open-source projects. *Information and Software Technology* 122 (2020), 106274.
- [4] Deepak Dhungana, Iris Groher, Elisabeth Schludermann, and Stefan Biffl. 2010. Software ecosystems vs. natural ecosystems: learning from the ingenious mind of nature. In *Proceedings of the Fourth European Conference on Software Architecture: Companion Volume*. 96–102.
- [5] Robert Evertse, Abel Lencz, Tea Šinik, Slinger Jansen, and Lamia Soussi. 2021. Is Your Software Ecosystem in Danger? Preventing Ecosystem Death Through Lessons in Ecosystem Health. In *International Conference on Agile Software Development*. Springer, 96–105.
- [6] Hadi Hemmati, Sarah Nadi, Olga Baysal, Oleksii Kononenko, Wei Wang, Reid Holmes, and Michael W Godfrey. 2013. The msr cookbook: Mining a decade of research. In *2013 10th Working Conference on Mining Software Repositories (MSR)*. IEEE, 343–352.
- [7] Sami Hyrynsalmi, Marko Seppänen, Tiina Nokkala, Arho Suominen, and Antero Järvi. 2015. Wealthy, Healthy and/or Happy—What does 'ecosystem health' stand for?. In *International conference of software business*. Springer, 272–287.
- [8] C Galina E Lezina and Artem M Kuznetsov. 2013. Predict closed questions on stackoverflow. (2013).

- [9] Yuzhan Ma, Sarah Fakhoury, Michael Christensen, Venera Arnaudova, Waheed Zogaan, and Mehdi Mirakhorli. 2018. Automatic Classification of Software Artifacts in Open-Source Applications. (2018).
- [10] Konstantinos Manikas. 2016. Revisiting software ecosystems research: A longitudinal literature study. *Journal of Systems and Software* 117 (2016), 84–103.
- [11] Daniel Massanori, Bruno BP Cafeo, Igor Wiese, and Awdren Fontão. 2020. Death of a Software Ecosystem: a Developer Relations (DevRel) perspective. In *Proceedings of the 34th Brazilian Symposium on Software Engineering*. 399–404.