
Envelhecimento e Rejuvenescimento de Software: 20 anos (1995 - 2014) – Panorama e Desafios

Nathália Assis Valentim



UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE COMPUTAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Uberlândia
2016

Nathália Assis Valentim

**Envelhecimento e Rejuvenescimento de
Software: 20 anos (1995 - 2014) – Panorama e
Desafios**

Dissertação de mestrado apresentada ao Programa de Pós-graduação da Faculdade de Computação da Universidade Federal de Uberlândia como parte dos requisitos para a obtenção do título de Mestre em Ciência da Computação.

Área de concentração: Ciência da Computação

Orientador: Autran Macêdo

Coorientador: Rivalino Matias Jr.

Uberlândia

2016

Dados Internacionais de Catalogação na Publicação (CIP)
Sistema de Bibliotecas da UFU, MG, Brasil.

V155e
2016

Valentim, Nathália Assis, 1990-
Envelhecimento e rejuvenescimento de software : 20 anos (1995-
2014) – panorama e desafios / Nathália Assis Valentim. - 2016.
73 f. : il.

Orientador: Autran Macêdo.
Coorientador: Rivalino Matias Jr
Dissertação (mestrado) - Universidade Federal de Uberlândia,
Programa de Pós-Graduação em Ciência da Computação.
Inclui bibliografia.

1. Computação - Teses. 2. Software - Teses. I. Macêdo, Autran. II.
Matias Júnior, Rivalino. III. Universidade Federal de Uberlândia,
Programa de Pós-Graduação em Ciência da Computação. III. Título.

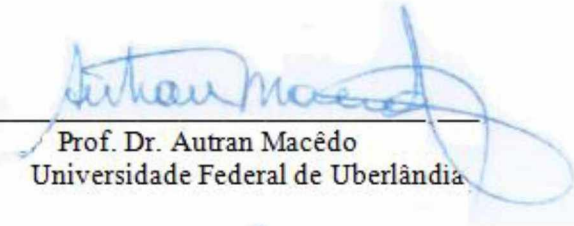
CDU: 681.3

UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE COMPUTAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

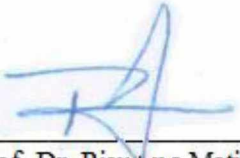
Os abaixo assinados, por meio deste, certificam que leram e recomendam para a Faculdade de Computação a aceitação da dissertação intitulada "**Envelhecimento e Rejuvenescimento de Software: 20 anos (1995-2014) – Panorama e Desafios**" por **Nathália Assis Valentim** como parte dos requisitos exigidos para a obtenção do título de **Mestre em Ciência da Computação**.

Uberlândia, 25 de Novembro de 2016

Orientador:

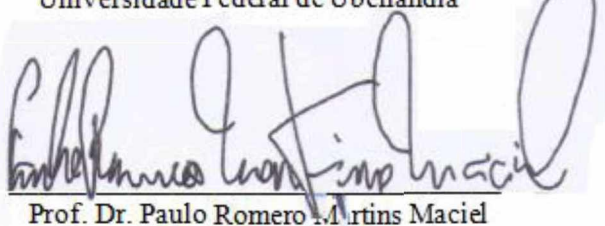

Prof. Dr. Autran Macêdo
Universidade Federal de Uberlândia

Coorientador:


Prof. Dr. Rivalino Matias Jr.
Universidade Federal de Uberlândia

Banca Examinadora:


Prof. Dr. Marcelo de Almeida Maia
Universidade Federal de Uberlândia


Prof. Dr. Paulo Romero Martins Maciel
Universidade Federal de Pernambuco

*Dedico este trabalho aos meus pais, Nelson e Antônia,
por todo amor, paciência, incentivo e apoio em minhas decisões.*

Agradecimentos

Não foi fácil, mas desistir nunca foi uma opção. E se hoje estou aqui, eu não consigo sozinha. Por isso, só tenho a agradecer. Agradeço aos meus orientadores, Autran e Rivalino, pela confiança em meu potencial e, claro, pela enorme paciência. Vocês foram verdadeiros orientadores! Ao Erisvaldo, sempre prestativo, com informações certas e acudindo alunos desesperados. Agradeço à CAPES pela bolsa de estudos durante o Mestrado. Aos meus amigos e familiares que entenderam e não julgaram a minha ausência por diversas vezes. Agradeço às minhas irmãs, Priscila e Anielise, que são também as minhas melhores amigas, sempre tão protetoras. Ao meu "namorado", Mateus, que sonhou esse sonho comigo, segurou minha mão e esteve ao meu lado o tempo todo. Aos meus pais, Nelson e Antônia, sempre tão compreensíveis e me incentivando a nunca parar de sonhar, pelos conselhos e apoio na hora das decisões difíceis que eu tive que tomar para seguir com os meus objetivos. Agradeço à Nossa Senhora, Mãe intercessora e medianeira, que nunca desampara os seus filhos aflitos. Por fim e mais importante, eu agradeço a Deus! Agradeço pela vida e por todos aqueles que Ele colocou em meu caminho, pois acredito que Ele se faz presente naqueles que estão em nossas vidas, com o propósito de nos ensinar e nos fortalecer.

A todos, o meu muito obrigada!

*“Não é sobre chegar no topo do mundo e saber que venceu. É sobre escalar e sentir que
o caminho te fortaleceu”
(Trem-Bala - Ana Vilela)*

Resumo

Embora o envelhecimento e rejuvenescimento de software seja um campo de pesquisa novo, em seus primeiros 20 anos muito conhecimento foi produzido. Hoje em dia, revistas e conferências científicas importantes incluem temas relacionados a SAR no seu âmbito de interesse. Este crescimento rápido e a grande variedade de locais de disseminação representam um desafio para os pesquisadores para manter o acompanhamento das novas descobertas e tendências nesta área. Neste trabalho, foram coletados e analisados dados de pesquisa em SAR para detectar tendências, padrões e lacunas temáticas, a fim de proporcionar uma visão abrangente deste campo de pesquisa em seus primeiros 20 anos. Adotou-se a abordagem de mapeamento sistemático para responder a perguntas de pesquisa, tais como: *Como os principais temas investigados em SAR têm evoluído ao longo do tempo? Quais são os efeitos do envelhecimento mais investigados? Quais técnicas e estratégias de rejuvenescimento são mais frequentemente usadas?*

Palavras-chave: Envelhecimento do Software. Rejuvenescimento de Software. Mapeamento Sistemático.

Abstract

Although software aging and rejuvenation is a young research field, in its first 20 years a lot of knowledge has been produced. Nowadays, important scientific journals and conferences include SAR-related topics in their scope of interest. This fast growing and wide range of dissemination venues pose a challenge to researchers to keep tracking of the new findings and trends in this area. In this work, we collected and analyzed SAR research data to detect trends, patterns, and thematic gaps, in order to provide a comprehensive view of this research field over its first 20 years. Adopted the systematic mapping approach to answer research questions such as: How the main topics investigated in SAR have evolved over time? Which are the most investigated aging effects? Which rejuvenation techniques and strategies are more frequently used?

Keywords: Software Aging. Software Rejuvenation. Systematic Mapping.

Lista de figuras

Figura 1 – Principais passos de uma RSL. Fonte: (KITCHENHAM; CHARTERS, 2007)	30
Figura 2 – Principais passos de um mapeamento sistemático.	32
Figura 3 – Relação entre falta, erro e falha.	41
Figura 4 – Relação entre falta, erro e falha em envelhecimento de software. Fonte: (GROTTKE; MATIAS; TRIVEDI, 2008)	41
Figura 5 – Classes de efeitos de envelhecimento. Fonte: (GROTTKE; MATIAS; TRIVEDI, 2008)	42
Figura 6 – Software com e sem rejuvenescimento de software.	44
Figura 7 – Abordagens de rejuvenescimento de software.	45
Figura 8 – Distribuição dos estudos ao longo dos anos.	56
Figura 9 – Distribuição de estudos de envelhecimento por tipo de estudo.	58
Figura 10 – Distribuição de estudos de rejuvenescimento por tipo de estudo.	58
Figura 11 – Distribuição de estudos relacionados com técnicas de modelagem.	60
Figura 12 – Distribuição de estudos relacionados com técnicas de modelagem e o tipo de estudo.	60
Figura 13 – Distribuição de estudos sobre envelhecimento efeitos por tipo de estudo.	61
Figura 14 – Porcentagem de classes de efeitos de envelhecimento por tipo de estudo.	62
Figura 15 – Recursos investigados por tipo de estudos relacionados com vazamento de recurso.	62
Figura 16 – Distribuição de estudos relacionados com o vazamento de memória.	63
Figura 17 – Recursos investigados por tipo de estudo relacionados com fragmentação.	63
Figura 18 – Distribuição de estudos ao longo dos anos sobre fragmentação	64
Figura 19 – Distribuição dos estudos utilizando indicadores de envelhecimento.	65
Figura 20 – Distribuição de estudos utilizando técnicas de rejuvenescimento.	67
Figura 21 – Distribuição de estudos envolvendo plataformas de sistemas operacionais.	69
Figura 22 – Distribuição de estudos envolvendo plataformas de sistemas operacionais por tipo de estudo.	70

Lista de tabelas

Tabela 1 – Comparação entre abordagens	32
Tabela 2 – Relação de especialistas consultados e seus respectivos locais de trabalho	50
Tabela 3 – Lista de conferências indicadas pelos especialistas	50
Tabela 4 – Lista de periódicos indicados pelos especialistas	51
Tabela 5 – Distribuição de autores por ano	57
Tabela 6 – Porcentagem de tipos de estudo por temas de estudo	57
Tabela 7 – Técnicas de modelagem mais usadas por tema de estudo.	59
Tabela 8 – Ocorrência de indicadores de envelhecimento	65
Tabela 9 – Técnicas de Rejuvenescimento por Tipo de Estudo	67
Tabela 10 – Número de Estudos por Estratégia de Rejuvenescimento.	68

Lista de siglas

ABNT	Associação Brasileira de Normas Técnicas
EBSE	Engenharia de Software Baseada em Evidências
NBR	Denominação de norma da Associação Brasileira de Normas Técnicas
RSL	Revisão Sistemática de Literatura
SAR	Envelhecimento e Rejuvenescimento de Software
VM	Máquina Virtual
VMM	Monitor de Máquina Virtual

Sumário

1	INTRODUÇÃO	23
1.1	Motivação	23
1.2	Objetivos e Desafios da Pesquisa	24
1.3	Hipótese	24
1.4	Contribuições	25
1.5	Organização da Dissertação	25
2	FUNDAMENTAÇÃO TEÓRICA	27
2.1	Principais Termos	27
2.2	Survey	28
2.3	Revisão Sistemática da Literatura	29
2.4	Mapeamento Sistemático	31
2.5	Survey x Revisão Sistemática x Mapeamento Sistemático . . .	32
2.6	Trabalhos Relacionados	34
3	ENVELHECIMENTO E REJUVENESCIMENTO DE SOFTWARE	39
3.1	Envelhecimento de Software	39
3.2	Rejuvenescimento de Software	43
4	MATERIAIS E MÉTODOS	47
4.1	Planejamento	47
4.1.1	Questões de Pesquisa	47
4.1.2	Estratégia de Busca de Estudos Primários	49
4.1.3	Critérios de Seleção	51
4.2	Condução	52
4.2.1	Procedimento de Seleção de Estudo	52
4.2.2	Estratégia de Extração de Dados	52

4.2.3	Síntese dos Dados Extraídos	52
4.3	Resultados	52
5	RESULTADOS E ANÁLISES	55
5.1	Temas dos Estudos	55
5.2	Tipos de Estudo	57
5.3	Técnicas de Modelagem	58
5.4	Efeitos de Envelhecimento	60
5.5	Indicadores de Envelhecimento	64
5.6	Técnicas de Rejuvenescimento	66
5.7	Estratégias de Rejuvenescimento	68
5.8	Plataformas de Sistemas Operacionais	69
6	CONCLUSÃO	71
6.1	Principais Contribuições	71
6.2	Ameaças à Validade	73
6.3	Trabalhos Futuros	73
6.4	Contribuições em Produção Bibliográfica	73
	REFERÊNCIAS	75
	 APÊNDICES	 79
	APÊNDICE A – LISTA DE ESTUDOS	81
A.1	Lista de Estudos da Amostra	81

Introdução

As pessoas, cada vez mais, têm se tornado dependentes dos computadores e de suas respectivas aplicações. Algumas dessas aplicações não podem ser interrompidas por executarem funções de extrema importância. Contudo, elas estão sujeitas ao envelhecimento de software.

O envelhecimento de software é um fenômeno que leva ao esgotamento progressivo de recursos, a degradação do desempenho, e, eventualmente, ao travamento ou falha do sistema de software. Esse fenômeno acontece devido a efeitos de envelhecimento. Há várias abordagens para mitigar os efeitos do envelhecimento de software. Algumas dessas abordagens constituem o rejuvenescimento de software (HUANG et al., 1995).

Esta pesquisa foi realizada para gerar informações relevantes para pesquisadores da área de envelhecimento e rejuvenescimento de software (SAR) e demais pesquisadores interessados nessa área. Essas informações serão geradas a partir de um rigoroso levantamento bibliográfico na área de SAR. Neste Capítulo serão apresentados a motivação, os objetivos e desafios da pesquisa, a hipótese e as contribuições desta pesquisa.

1.1 Motivação

Entre as aplicações de computadores, existem as denominadas aplicações críticas. Estas possuem vida crítica, cuja falha ou mau funcionamento pode resultar em mortes ou lesões graves a pessoas, perda ou danos graves de equipamentos e danos ambientais. Podem ser considerados como aplicações críticas, por exemplo, sistemas de monitoramento de pacientes em um hospital, ou então, sistemas embarcados em veículos (KUO; HUANG; LYU, 2001). Em outros casos, existem aplicações que devem ser executadas ininterruptamente, como os sistemas bancários e sistemas de monitoramento de câmeras de segurança.

Devido às suas características, por exemplo, ser executada continuamente, as aplicações críticas são mais suscetíveis a sofrer uma degradação de desempenho ao ponto de sua execução ser descontinuada. Essa degradação pode ocorrer, por exemplo, por ques-

tões de fragmentação de memória, erros numéricos e/ou corrupção de dados (GROTTKE; MATIAS; TRIVEDI, 2008).

O envelhecimento de software surgiu na década de 90 e consiste em um fenômeno onde ocorre a perda de desempenho do software ao longo de sua execução. Essa perda de desempenho pode afetar o ambiente de execução do sistema, podendo haver aumento na taxa de falhas, na medida em que o tempo de execução aumenta (COTRONEO et al., 2014). Para reduzir os efeitos do envelhecimento de software, surgiram ações como, por exemplo, descontinuar a aplicação que está sofrendo de envelhecimento de software, reiniciar sistemas operacionais, máquinas virtuais e aplicações de maneira independente das demais aplicações (ALONSO et al., 2013). Essas ações compõem o rejuvenescimento de software.

O rejuvenescimento de software consiste de um conjunto de ações ou técnicas que se aplica a um software que está sofrendo com os efeitos do envelhecimento. Mais especificamente, o rejuvenescimento de software consiste em procurar colocar o software em um estado onde ele não apresenta efeitos de envelhecimento. Entre as suas principais abordagens, encontra-se as técnicas de rejuvenescimento, técnicas de modelagem e estimativas de rejuvenescimento. Realizar o rejuvenescimento em um software requer a periodicidade adequada para detectar os efeitos de envelhecimento existentes. Além disso, o diagnóstico de um software sofrendo de envelhecimento deve ser preciso para que a tomada de decisão sobre realizar ou não o procedimento de rejuvenescimento seja feito com as técnicas adequadas.

1.2 Objetivos e Desafios da Pesquisa

O objetivo da pesquisa é identificar possíveis direções futuras para os principais campos de envelhecimento e rejuvenescimento de software. Essa identificação ocorrerá através de um levantamento bibliográfico rigoroso que visa fornecer uma visão geral sobre o estado da arte da área e detectar oportunidades de pesquisas. Estudos aprofundados sobre qualquer uma das direções apontadas neste trabalho serão objetos de trabalhos futuros. Esse levantamento rigoroso será baseado em um Mapeamento Sistemático, explicado no Capítulo 4.

1.3 Hipótese

É possível descobrir nichos de pesquisa inexplorados, pouco explorados e próximos à saturação na área de envelhecimento e rejuvenescimento de software através de um mapeamento sistemático.

1.4 Contribuições

A análise dos dados coletados visa identificar tópicos emergentes que ainda não foram percebidos ou foram poucos explorados pelos pesquisadores da área de SAR. Essa área está diretamente relacionada com qualidade de software, uma vez que, muitos efeitos de envelhecimento de software são provenientes de erros de programação que comprometem o bom funcionamento do software. Os tópicos identificados irão contribuir com interessados como, por exemplo, os pesquisadores ou que estão iniciando seus estudos na área de SAR. Para esses pesquisadores a contribuição é uma visão geral do estado da arte, ajudando-os a encontrar temas de pesquisa em potencial. Os tópicos também irão contribuir com os pesquisadores que já são experientes na área de SAR. Para esses pesquisadores a contribuição é identificar lacunas e oportunidades de pesquisa.

1.5 Organização da Dissertação

Neste trabalho, concentra-se no Capítulo 1 a introdução desta dissertação. A fundamentação teórica é apresentada no Capítulo 2. Os detalhes sobre envelhecimento e rejuvenescimento de software são apresentados no Capítulo 3. Os materiais e métodos utilizados são apresentados no Capítulo 4. Os resultados e análises são mostrados no Capítulo 5. Por fim, a conclusão do trabalho é apresentada no Capítulo 6.

Fundamentação Teórica

Os achados deste trabalho foram obtidos a partir de uma revisão da literatura. Há três abordagens para se realizar uma revisão de literatura: *Survey*, Revisão Sistemática de Literatura e Mapeamento Sistemático. Este capítulo descreve cada uma dessas abordagens, nas Seções 2.2, 2.3 e 2.4. Por fim, a Seção 2.5 apresenta as principais diferenças entre as mesmas.

2.1 Principais Termos

Em uma revisão de literatura existem termos nos quais o entendimento deles é fundamental para compreender e executar todo processo dessa revisão. Os principais termos de uma revisão de literatura são apresentados nesta seção, de acordo com as definições de (KITCHENHAM; CHARTERS, 2007).

❑ Critérios de Seleção

Os critérios de seleção de estudos são utilizados para determinar quais os estudos serão incluídos ou excluídos na amostra utilizada para a revisão de literatura.

❑ Estudos Primários

Os estudos primários são aqueles que investigam uma questão de pesquisa específica. Por exemplo, experimentos, estudos de caso e pesquisas de opinião.

❑ Estudos Secundários

Os estudos secundários são aqueles que revisam estudos primários relativos a uma pesquisa com o objetivo específico de integração/síntese de evidências relacionadas pesquisa.

❑ Protocolo

Um protocolo tem como objetivo especificar todos os métodos e procedimentos que serão utilizados para realizar um revisão de literatura específica.

❑ Questões de Pesquisa

As questões de pesquisa são essenciais para a realização de uma revisão de literatura. São essas questões que ajudam a conduzir todo o processo da revisão, uma vez que, a partir delas é que se tem uma orientação para coleta de estudos primários.

2.2 Survey

A abordagem conhecida como *Survey* consiste em uma técnica usada em pesquisas onde há a análise de literatura publicada em livros, artigos de revistas e da interpretação juntamente com uma análise crítica pessoal do autor (ROTHER, 2007). Nesse caso, os estudos são coletados primeiramente, constituindo a amostra. Os dados/informações são obtidos a partir da análise desses estudos. O *Survey* é um estudo em uma área particular para preencher lacunas no conhecimento do investigador, em uma determinada área de pesquisa, não gerando novos conhecimentos, apresentando somente os resultados do que foi encontrado, não apontando explicitamente, por exemplo, opiniões ou novas tendências.

Em um *Survey*, através de algum instrumento de pesquisa, se obtém dados ou informações sobre características, ações ou opiniões de determinada unidade de análise (por exemplo, um grupo de pessoas). Em (FREITAS et al., 2000) o autor diz que o *Survey* é quantitativo e sua escolha deve estar associada aos objetivos da pesquisa. Onde, tanto os métodos qualitativos quanto os quantitativos possuem vantagens e desvantagens. O pesquisador tem a responsabilidade de realizar uma análise e avaliar qual método é o mais adequado para sua pesquisa.

Utilizar o *Survey* como abordagem de pesquisa é apropriado quando é desejado responder questões onde o foco é sobre “o que está acontecendo” ou “como e porque isto está acontecendo”(FREITAS et al., 2000). Em (PINSONNEAULT; KRAEMER, 1993) os autores classificaram a pesquisa *Survey* quanto a sua finalidade em:

❑ Explanatória

Identifica os fatores que determinam ou contribuem para a ocorrência de determinados fenômenos. Essa finalidade testa uma teoria e relações causais, ou seja, relações entre causa e efeito, sendo o efeito a consequência da causa. Além disso, não apenas estabelece a existência de relações causais, mas também se questiona porque a relação existe.

❑ Exploratória

Familiariza-se com um tópico ou identifica conceitos iniciais sobre um tópico, visando o tornar mais explícito e dando ênfase na determinação de quais conceitos devem ser medidos e como eles devem ser medidos. No propósito exploratório, busca-se descobrir novas possibilidades e dimensões da comunidade de interesse.

□ Descritiva

Busca identificar quais situações, eventos, atitudes ou opiniões estão se manifestando em uma comunidade. A distribuição e características de fenômenos na comunidade são descritos. Neste tipo de *Survey* a hipótese não é causal, mas tem o propósito de verificar se a percepção dos fatos está ou não de acordo com a realidade.

Na área da computação, os *Surveys* são utilizados para a realizar revisões de literatura. Em (PEK; LÄMMEL, 2013) os autores realizaram um survey para obter evidências empíricas no uso de coleções de artefatos de software. A motivação foi melhorar a qualidade e reprodutibilidade da pesquisa, entendendo as características de artefatos de software utilizados. Para tanto, realizaram uma pesquisa bibliográfica utilizando a abordagem survey. Outros dois exemplos de *Survey* são apresentados na Seção 2.6.

Em (KITCHENHAM et al., 2002) os autores propuseram algumas diretrizes preliminares para Experimentos Formais e *Surveys*. Apesar dos *Surveys* terem vantagens como uma coleta ampla e eficiente de informações, eles também possuem desvantagens. Um *Survey* tenta lidar com vários tipos diferentes de estudos e, com isso, as orientações não são tão específicas e/ou detalhadas. Além disso, eles podem cometer erros com a unidade experimental. Por exemplo, pode haver uma situação em que questionários são enviados a vários entrevistados em uma determinada organização, contudo as questões a serem respondidas não dizem respeito aos entrevistados, mas sim à organização como um todo. Nesse caso houve um erro quanto a unidade experimental e se este problema não for detectado, há chances de resultados significativamente incorretos.

2.3 Revisão Sistemática da Literatura

A Revisão Sistemática da Literatura (RSL) começou a ser explorada na medicina. Em (BRERETON et al., 2007), a amostra continha pesquisas que indicavam que a recomendação médica baseada na opinião especializada não era tão confiável quanto a recomendações baseadas na acumulação de resultados de experimentos científicos. Desde então, muitas áreas têm adotado essa abordagem.

Na Engenharia de Software, a aplicação da RSL, surgiu em 2004 no ICSE (International Conference on Software Engineering), criando o conceito de Engenharia de Software Baseado em Evidências (EBSE) em (KITCHENHAM; DYBA; JORGENSEN, 2004). A EBSE busca fornecer os meios para que as melhores evidências da pesquisa possam ser integradas com a experiência prática na tomada de decisão no desenvolvimento e manutenção de software (BRERETON et al., 2007).

A RSL consiste em uma metodologia bem definida para identificar, analisar e interpretar todas as evidências disponíveis a respeito de uma questão de pesquisa específica

de maneira imparcial (KITCHENHAM; CHARTERS, 2007). Os estudos que contribuem com uma revisão sistemática podem ser estudos primários ou secundários.

A metodologia para uma RSL utiliza um guia para o desenvolvimento de ESBE que foi proposto por (KITCHENHAM, 2004) e é amplamente utilizado em pesquisas que envolvam RSL na Engenharia de Software em geral. Este guia divide o processo de uma RSL em três passos principais: planejamento da RSL, justificando a necessidade da RSL; condução da RSL, onde a amostra é coletada, os estudos primários/secundários são lidos e os dados são extraídos. Por fim, a análise dos resultados onde há a produção de relatórios para publicações das informações geradas. A Figura 1 mostra os passos de uma RSL. Cada uma das fases executada tem os seus resultados devem ser armazenados. Portanto, a fase de armazenamento de revisão sistemática é realizada através de todo o processo.

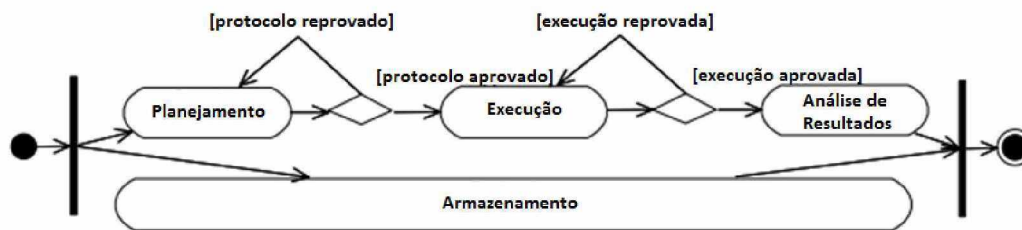


Figura 1 – Principais passos de uma RSL. Fonte: (KITCHENHAM; CHARTERS, 2007)

Há muitas razões para proceder a uma revisão sistemática da literatura. Os motivos mais comuns são para resumir evidências existentes sobre um tratamento ou tecnologia. Além disso, outra razão é identificar eventuais lacunas na pesquisa atual, com a finalidade de sugerir novas áreas de investigação, além de posicionar adequadamente novas atividades de investigação. No entanto, revisões sistemáticas de literatura também podem ser realizadas para examinar até em que ponto uma evidência empírica suporta ou contradiz hipóteses teóricas, ou mesmo para auxiliar a geração de novas hipóteses (KITCHENHAM; CHARTERS, 2007).

A RSL também é utilizada na área da computação. Em (LIMA et al., 2012) foi realizada uma RSL a fim de obter informações sobre a evolução dos processos de desenvolvimento de um software educativo. A motivação da RSL foi levantar as técnicas e os processos utilizados para desenvolver softwares educativos. Os resultados obtidos apresentaram tendências no desenvolvimento desses softwares. Além disso, foi fornecido um referencial de estudos primários para futuras pesquisas.

Em geral, as pesquisas começam com algum tipo de revisão de literatura. No entanto, se uma revisão de literatura não for completa e justa, ou seja, se a amostra não possuir o maior número possível de estudos relevantes e se a análise desses estudos não for imparcial e avaliada por outros pesquisadores, ela terá pouco valor científico. Por esse motivo, torna-se importante a realização de revisões sistemáticas. Uma revisão sistemática sintetiza o trabalho existente de uma maneira que seja justa e sem ser tendenciosa (KITCHENHAM;

CHARTERS, 2007). Os pesquisadores que realizam esse tipo de revisão devem se esforçar para identificar e relatar pesquisas que não suportam a hipótese de pesquisa definida, bem como identificar e relatar pesquisas que a suportam.

As vantagens de se realizar uma revisão sistemática consistem em possuir uma metodologia bem definida que reduz a possibilidade de existência de resultados incorretos. Além disso, uma RSL pode fornecer informações sobre algum fenômeno que esteja em observação. Por exemplo, se os estudos dão resultados consistentes, se tem evidências de que o fenômeno também é consistente e pode ser aplicado em outros contextos. Se os estudos dão resultados inconsistentes, outras fontes podem ser estudadas até que se obtenha resultados consistentes. Quanto às desvantagens, a principal desvantagem das revisões sistemáticas da literatura é que elas exigem muito mais esforço do que revisões de literatura tradicionais (KITCHENHAM; CHARTERS, 2007).

2.4 Mapeamento Sistemático

Os mapeamentos sistemáticos são projetados para fornecer uma ampla visão de uma área de pesquisa (KEELE, 2007). A partir de um mapeamento é possível saber se existem evidências de pesquisa sobre um determinado tempo e também quantificar essas evidências. O mapeamento sistemático também é um tipo de revisão de literatura, onde a revisão de estudos em um determinado tópico são realizadas de acordo com um protocolo (MIAN et al., 2005), identificando evidências sobre esse tópico. O mapeamento sistemático é considerado, portanto, um estudo onde se identifica lacunas onde novos estudos para preencher essas lacunas são necessários. Além disso, também identifica agrupamentos de evidências que podem levar a novas revisões de literatura (KEELE, 2007).

Assim como na RSL, o mapeamento sistemático também tem três etapas principais: planejamento, condução e resultados. Os passos de um mapeamento sistemático consistem, na fase de planejamento, em formular as questões de pesquisas que serão respondidas, selecionar os repositórios onde os estudos serão coletados (bibliotecas digitais) e aplicar critérios de inclusão e exclusão. Na fase de condução, é realizada a leitura e extração de dados. Na fase final, os resultados são produzidos e analisados. Os passos principais um mapeamento sistemático é mostrado na Figura 2

O estudo de mapeamento sistemático fornece uma visão geral de uma área de pesquisa, identificando a quantidade, os tipos de pesquisas realizadas, os resultados disponíveis, além das frequências de publicações ao longo do tempo para identificar tendências (RAMOS; BRASIL, 2011). Além disso, busca identificar lacunas no conjunto dos estudos primários ou secundários onde seja direcionado o foco de revisões sistemáticas futuras e identificar áreas onde mais estudos precisam ser conduzidos. Exemplos de estudos que utilizaram mapeamento sistemático como abordagem são mostrados em 2.6.

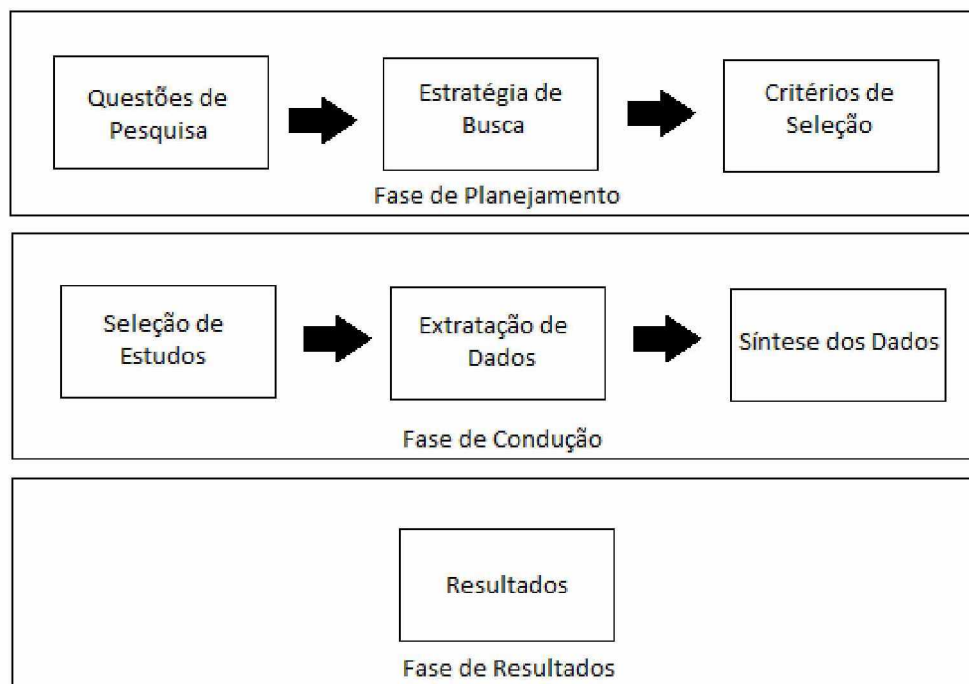


Figura 2 – Principais passos de um mapeamento sistemático.

2.5 Survey x Revisão Sistemática x Mapeamento Sistemático

A Tabela 1 mostra uma breve comparação entre as três abordagens para a realização de uma revisão de literatura: survey, revisão sistemática da literatura e mapeamento sistemático.

Tabela 1 – Comparação entre abordagens

Características	Survey	RSL	Map. Sistemático
Estudo Secundário	X	X	X
Seleção de participantes	X	X	X
Definição de Protocolo	X	X	X
Critérios de Seleção de Estudos		X	X
Produção de novos conhecimentos		X	X
Questões de pesquisas amplas	X	X	
Questões de pesquisas específicas			X

As três abordagens possuem semelhanças e diferenças quanto ao procedimento para realizar revisões de literatura. Entre as semelhanças entre as três abordagens está a seleção de participantes para a condução da revisão. Outra semelhança entre as abordagens é a construção de um protocolo onde se define todos os passos que serão realizados. Diferentemente de um mapeamento sistemático e uma revisão sistemática da literatura, no *Survey*, mesmo seguindo um protocolo para montagem da amostra, não há critérios de

seleção de estudos para inclusão ou exclusão de estudos. Em um *Survey* não há a produção de novos conhecimentos, conforme dito na Seção 2.2, trata-se apenas de um estudo realizado em um área particular para preencher lacunas no conhecimento do investigador, onde é apresentado os achados do *Survey* e retratando a área de estudo.

Uma grande diferença entre as três abordagens pode ser percebida no que diz respeito às questões de pesquisa. No *Survey* e no mapeamento sistemático as questões de pesquisa definidas tem a característica de especificidade menores quando comparadas às questões de pesquisa de uma RSL, consequentemente as respostas também são mais gerais em *Surveys* e mapeamentos sistemáticos. Assim, o número de estudos retornados em uma RSL tende a ser menor em relação aos números de estudos de um *Survey* e um mapeamento sistemático.

Em (KEELE, 2007) são apresentadas as principais diferenças entre uma revisão sistemática de literatura e um mapeamento sistemático. Geralmente, em um mapeamento as questões de pesquisa que conduzem a revisão são mais gerais do que em uma RSL. Os termos de pesquisa para estudos de um mapeamento sistemático são menos centrados do que para Revisões Sistemáticas de Literatura e tendem a retornar um número maior de estudos. Contudo, além da intenção de um mapeamento ser uma ampla cobertura de pesquisa, pode-se realizar filtros, a partir dos critérios de inclusão e exclusão para montar a amostra.

O processo de extração de dados dos estudos em um mapeamento também é mais amplo do que o processo de extração de dados em uma RSL. O objetivo desta etapa é classificar estudos com detalhes suficientes para responder às questões de pesquisa. A etapa de análise de um estudo em um mapeamento resume os dados para responder as questões definidas. É improvável que, em um mapeamento, técnicas de análise em profundidade sejam incluídas. Representações gráficas com distribuições de estudo por tipo de classificação pode ser um mecanismo de comunicação eficaz em um mapeamento sistemático.

Se ao realizar uma avaliação inicial sobre um determinado domínio para a realização de uma RSL é descoberto que há poucas evidências sobre um tópico ou então que se trata de tópico é muito amplo, é recomendável que seja feito um mapeamento sistemático em vez de uma RSL. A explicação é que um mapeamento sistemático permite que as evidências em um domínio sejam plotadas em um alto nível de granularidade. Esse nível de granularidade permite a identificação de grupos de evidências ou evidências isoladas para orientar o enfoque das futuras revisões sistemáticas e identificar áreas para a realização de estudos primários (KEELE, 2007). Em suma, um mapeamento sistemático procura explorar uma área de pesquisa ampla, detectando sub-áreas. A RSL, por sua vez, busca se aprofundar nessas sub-áreas.

Neste trabalho, o mapeamento sistemático com estudos primários mostrou ser a forma mais adequada aos objetivos da pesquisa. A justificativa é que esse tipo de revisão de

literatura utiliza uma metodologia de base semelhante a de uma RSL. Contudo, o estudo de mapeamento sistemático fornece uma visão mais ampla de uma área de pesquisa, com a intenção de ser uma revisão formal, rigorosa, confiável e passível de auditoria.

2.6 Trabalhos Relacionados

O mapeamento sistemático com estudos primários foi o procedimento escolhido para realizar a revisão de literatura apresentada neste trabalho. Dessa forma, são apresentados trabalhos que não estão relacionados, diretamente, com a área de envelhecimento e rejuvenescimento de software. Contudo, eles também realizaram mapeamento sistemático. Assim, foi considerado importante apresentar esses trabalhos com o objetivo de mostrar que o mapeamento sistemático é utilizado na realização de outras pesquisas que envolvem a área da Computação.

Em (MALCHER; OLIVEIRA,), o objetivo foi apresentar um mapeamento sistemático para descobrir, na literatura, as abordagens de apoio à rastreabilidade de requisitos no contexto de projetos de software e fazer a categorização dos dados encontrados, a fim de demonstrar, por meio de um método confiável, rigoroso e auditável, como esta área tem se desenvolvido e quais as principais abordagens são utilizadas para sua implementação.

Em (RAMOS; BRASIL, 2011), o objetivo foi investigar, catalogar e classificar os padrões para reengenharia de sistemas publicados em Conferências e Workshops PLoP (Pattern Languages of Programs), por meio de um estudo de mapeamento sistemático. Uma catalogação bibliográfica foi realizada e, além disso, o estudo apresentou a totalização dos padrões quanto ao: tipo de processo de reengenharia; o levantamento das linguagens de programação abordadas; as Conferências e Workshops onde os padrões foram publicados e, por fim, a evolução das publicações por ano. Para realizar a classificação, os padrões de reengenharia foram categorizados de acordo com as disciplinas do RUP (Rational Unified Process).

Em (XAVIER et al., 2014), o objetivo foi apresentar os resultados de um mapeamento sistemático sobre confiabilidade de software. Foram utilizadas técnicas estatísticas para agrupar artigos similares e identificar tópicos de pesquisa em confiabilidade de software. Os resultados mostraram que trabalhos que adotam uma abordagem puramente teórica predominam sobre aqueles que adotam abordagem experimental. Além disso, observou-se que entre os tópicos de pesquisa abordados no trabalho, os resultados indicaram uma demanda por pesquisas teórico-experimentais em testes de software para os próximos anos.

Considerou-se importante também a apresentação de trabalhos que estivessem diretamente relacionados com SAR. Há vários trabalhos sobre SAR desde quando o termo foi cunhado em 1995. Contudo, apenas dois trabalhos similares ao proposto nesta pesquisa foi apresentado em (BARADA; SWAIN, 2014) e em (COTRONEO et al., 2014). Os trabalhos mostrados a seguir estão diretamente relacionados com a área de envelheci-

mento e rejuvenescimento de software e são apresentados para mostrar quão promissoras as revisões de literatura são na área de SAR.

Em (BARADA; SWAIN, 2014), os autores realizaram um *Survey*. Nessa abordagem, a pesquisa foi limitada a estudos realizados em ambientes virtualizados, ou seja, estudos que envolvessem máquina virtual (VM) e/ou monitor de máquina virtual (VMM). O artigo faz uma breve introdução ao envelhecimento de software. No entanto, a principal intenção é falar sobre rejuvenescimento de software e suas respectivas técnicas para eliminação de efeitos de envelhecimento. Por conta dessa restrição de ambientes virtualizados, a pesquisa deu mais ênfase aos estudos realizados na área de Rejuvenescimento.

A motivação do trabalho foi propor um algoritmo que permitirá a escolha de uma técnica de rejuvenescimento para a máquina virtual (VM) e monitor de máquina virtual (VMM). A escolha da técnica de rejuvenescimento seria escolhida corretamente, uma vez que, seria feita de acordo com o efeito de envelhecimento sofrido no software.

Os ambientes virtualizados vêm se tornando uma escolha popular para implantar aplicações complexas e que exigem um longo tempo de execução. Além disso, ambientes virtualizados podem resultar em redução de custos e espaço já que é possível se ter várias máquinas lógicas funcionando em apenas uma máquina física.

A pesquisa realizada foi voltada para ambientes virtualizados e há uma explicação pertinente para a escolha dos autores. Os ambientes virtualizados sendo uma opção em ascensão, espera-se que eles possam estar sempre disponíveis. Contudo, essa disponibilidade os deixam mais propensos a sofrer de envelhecimento de software. O *Survey* dos autores analisa, na área de SAR, apenas 11 estudos que falam sobre ambientes virtualizados. Eles apresentaram a diferença entre efeitos de envelhecimento presentes em máquina física e em máquina virtual.

Os problemas de envelhecimento de software que se encontram em máquinas físicas também foram apresentados. Os autores fizeram uma comparação com diferentes problemas de envelhecimento de software que afetam os ambientes virtualizados. Após a apresentação dos problemas de envelhecimento em máquinas físicas e virtuais, salientando as diferenças entre eles, os autores introduziram diferentes técnicas de rejuvenescimento para mitigar os efeitos de envelhecimento presentes em sistemas virtualizados e realizando também comparações entre essas técnicas.

Os autores perceberam que problemas que surgem em máquinas físicas podem ser evitados em máquinas virtuais. Ainda que algumas técnicas de rejuvenescimento foram propostas para mitigar os efeitos do envelhecimento em ambiente virtualizado, nem sempre estas técnicas de rejuvenescimento poderão remover os efeitos do envelhecimento corretamente de VM e VMM. Então, a partir dos resultados do *Survey*, os autores tem como motivação, em um futuro, a construção de um algoritmo para escolher técnicas corretas de rejuvenescimento de software para a máquina virtual (VM) e monitor de máquina virtual (VMM) de acordo com os seus respectivos efeitos de envelhecimentos.

Em (COTRONEO et al., 2014) os autores também realizaram um *Survey* que abrangeu estudos de 1992 a 2011. Os estudos publicados antes de 1995 não eram específicos, mas de alguma forma estavam relacionados aos tópicos investigados no SAR. Foi apresentada uma análise abrangente da literatura de envelhecimento e rejuvenescimento de software, através da revisão de 495 artigos publicados nas áreas de Engenharia de Software e Confiabilidade de Software.

O artigo forneceu uma visão geral dos estudos sobre envelhecimento de software e rejuvenescimento de software (SAR) que têm sido publicado em principais periódicos e anais de conferências. Forneceu também uma visão geral no que diz respeito às abordagens estatísticas que têm sido utilizadas para previsão de fenômenos de envelhecimento de software e o planejamento do rejuvenescimento de software. Tratou-se também sobre os tipo de sistemas e os efeitos de envelhecimento que têm sido estudados e as técnicas que têm sido propostas para rejuvenescer sistemas de software.

O artigo fez uma breve introdução ao envelhecimento e rejuvenescimento de software, apresentando conceitos básicos em SAR como, por exemplo, indicadores de envelhecimento e o processo de rejuvenescimento de software. Em seguida, os autores fizeram uma análise da literatura, onde o processo de coleta dos estudos primários foram coletados. O processo foi dividido em duas etapas:

- ❑ Bootstrap Nesse passo, os autores realizaram a coleta dos estudos primários em diversos veículos de publicação. Eles não se limitaram a buscar os estudos primários em veículos de publicação específicos. Realizaram as coletas daqueles estudos primários condizentes com o tema SAR e, a partir dessa coleta, avaliaram quais foram os veículos de publicação.
- ❑ Recursive Closure Nesse passo, os autores realizaram uma nova coleta de outros estudos primários, a partir das referências bibliográficas dos estudos primários coletados inicialmente.

Os autores realizaram uma análise detalhada dos estudos coletados, apresentando quatro dimensões ortogonais, na intenção de fornecer uma visão abrangente da literatura e perspectivas sobre direções futuras pesquisas em SAR. Cada dimensão é composta por um conjunto de classes que são usados para classificar os estudos pesquisados. As dimensões são as seguintes:

- ❑ Tipo de Análises

Estudos baseados em modelo e em medições; estudos híbridos; estudos com técnicas de Rejuvenescimento; estudos sobre prevenção, verificação e depuração de Software Afetados por envelhecimento de software.

- ❑ Tipo de Sistemas

Sistemas *safety-critical*, sistemas *non-safety-critical* e sistemas não especificados.

❑ Indicadores de Envelhecimento e Efeitos de Envelhecimento

Consumo de Memória, Degradação de Desempenho e outros consumos de recursos.

❑ Técnicas de Rejuvenescimento

Restart de aplicação, *Reboot* de sistemas operacionais, *Restart* de monitor de máquina virtual (VMM) e *Restart* máquina virtual (VM).

Por fim, os autores fazem uma discussão dos resultados que encontraram em cada dimensão ortogonal. A análise abrangente da literatura de SAR foi útil para identificar os principais resultados da pesquisa SAR e destacar tendências e questões em aberto que merecem atenção no futuro próximo pela comunidade SAR.

Diferentemente do (COTRONEO et al., 2014) e (BARADA; SWAIN, 2014), neste trabalho se analisou separadamente os estudos sobre o envelhecimento de software e sobre o rejuvenescimento de software. Além disso, comparado aos trabalhos similares, foram incluídos estudos mais recentes na amostra deste trabalho, tornando-o importante em relação ao estado da arte das pesquisas em SAR. Concentrou-se, principalmente, em estudos publicados pela comunidade SAR, à procura de possíveis padrões que emergem a partir dele. Considerou-se também elementos da SAR que não foram cobertos em (COTRONEO et al., 2014), como as estratégias de rejuvenescimento e plataformas de sistemas operacionais.

Envelhecimento e Rejuvenescimento de Software

A noção de envelhecimento e rejuvenescimento de software (SAR) foi introduzida há duas décadas. O interesse em SAR vem aumentando na indústria e na área acadêmica. Os detalhes sobre envelhecimento e rejuvenescimento de software são mostrados nas seções a seguir.

3.1 Envelhecimento de Software

A sociedade nos dias atuais tem dependido cada dia mais dos sistemas computacionais que são constituídos de hardware e software. Contudo, esses sistemas podem se degradar, sendo o software o componente responsável pela maior parte dos erros dos sistemas (GROTTKE et al., 2006). Essa degradação pode ocorrer ao longo de horas de funcionamento do sistema ininterruptamente, tendo afetada a **qualidade do software**, especialmente relacionadas à desempenho e confiabilidade. Além disso, as taxas de falha do software podem crescer ao longo da execução. Essas características definem o envelhecimento de software.

O envelhecimento de software é um fenômeno que, ao contrário do que o termo pode sugerir, não tem relação com sistemas obsoletos, tão pouco está relacionado com deteriorações do hardware. O envelhecimento de software pode ser causado por defeitos no software, consequência de quando não há boas práticas no desenvolvimento, ou mesmo por degradações naturais na estrutura do software (sem interferência humana). É um fenômeno presente desde o início da era dos computadores, mas só foi formalmente postulado e investigado sobre uma base científica na metade da década de 90.

Em 1995, em (HUANG et al., 1995) os autores mencionaram casos reais que envolveram um sistema de coleta de dados de bilhetagem, um sistema de processamento de chamadas de telefonia e aplicações científicas. Esses sistemas sofreram perda de desempenho por serem executados por um longo período de tempo, sem interrupções. O termo

"envelhecimento de software" foi adotado em trabalhos posteriores e, desde então, muitas abordagens têm sido usadas para estudar esse fenômeno. Essas investigações estão concentradas em compreender os efeitos do envelhecimento de software à luz de experimentos e da teoria.

Em (GROTTKE; MATIAS; TRIVEDI, 2008) os autores discutiram sobre os fundamentos do envelhecimento de software. Eles concentraram em aspectos conceituais e práticos envolvidos no tema. Além disso, eles formularam um conjunto de definições que consideraram essenciais para compor uma taxonomia para a pesquisa em envelhecimento e rejuvenescimento de software. Para os autores, o envelhecimento de software geralmente é causado por falhas de software.

Considera-se como falha de software, quando há um desvio do resultado esperado pelo software de acordo com o que foi especificado para o seu funcionamento (GROTTKE; MATIAS; TRIVEDI, 2008). Esse desvio pode ser na forma de um serviço incorreto ou nenhum serviço. Além disso, o sistema pode estar em um estado de degradação em que o serviço é lento ou limitado, denominada falha parcial. Em envelhecimento de software a falha do software é causada especificamente por conta desse fenômeno. Geralmente, essas falhas acontecem em sistemas que ficam ininterruptamente funcionando durante longos períodos como, por exemplo, servidores web (MATIAS; FILHO, 2006). Quanto mais tempo um sistema fica em execução, sofrendo de envelhecimento de software, mais a sua taxa de falha aumenta.

Uma falha de software se inicia a partir de uma falta. A falta, conhecida também como defeito, pode ser ativada, causando a propagação de erros. Denomina-se como propagação de erros quando há a transformação de erros e eles podem levar a ocorrência de outros erros. A propagação de erros desencadeia a uma falha do sistema se o erro é propagado para a interface do sistema, fazendo com que o serviço prestado se desvie da sua especificação e seja percebido pelo usuário (GROTTKE; MATIAS; TRIVEDI, 2008). A relação entre esses três elementos é mostrada na Figura 3.

No que diz respeito ao envelhecimento de software, muitas falhas de sistemas de computacionais realmente são as consequências de falhas de software que ocorrem. Conceitos específicos foram criados para faltas, erros e falhas em envelhecimento de software. As faltas relacionadas ao envelhecimento de software são nomeadas como **AR-faults** (*Aging Related Faults*). Os erros relacionados ao envelhecimento de software são nomeados como **AR-errors** (*Aging Related Error*). Por fim, as nomeadas **AR-failures** (*Aging Related Failures*) são falhas relacionadas ao envelhecimento de software (GROTTKE; MATIAS; TRIVEDI, 2008).

As faltas relacionadas ao envelhecimento (**AR-faults**) podem ser ativadas na presença de fatores de envelhecimento. Esses fatores podem ser internos ou externos. Os fatores internos são eventos que acontecem internamente em um sistema, enquanto os fatores externos são acionados diretamente por elementos no ambiente do sistema, como por

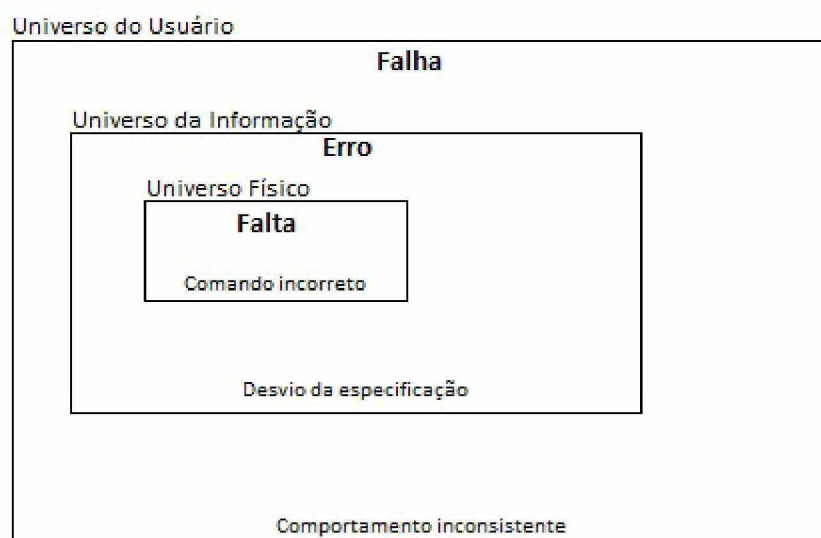


Figura 3 – Relação entre falta, erro e falha.

exemplo, os seus usuários. Um só fator de envelhecimento ou a combinação de fatores levam a ativação de uma falta (por exemplo, chamada de uma função com a presença de uma *AR-fault*).

Uma vez que as **AR-faults** são ativadas, elas dão sequência aos erros relacionados ao envelhecimento (**AR-errors**). Esses erros geram efeitos de envelhecimento que vão se acumulando, mudando o estado correto do funcionamento do software para um estado onde falhas poderão acontecer. O acúmulo desses efeitos de envelhecimento podem causar as falhas relacionadas ao envelhecimento (**AR-Failures**) que podem, inclusive, causar novas faltas no software. A Figura 4 mostra a relação entre faltas, erros e falhas em envelhecimento de software.

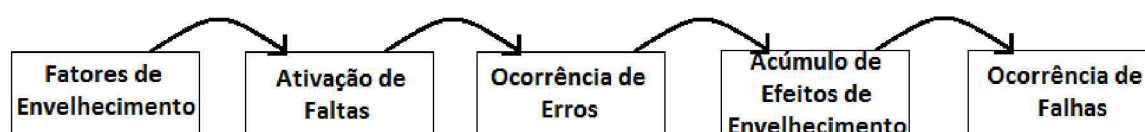


Figura 4 – Relação entre falta, erro e falha em envelhecimento de software. Fonte: (GROTTKE; MATIAS; TRIVEDI, 2008)

Em (GROTTKE; MATIAS; TRIVEDI, 2008), baseado nas diferenças e similaridades entre as características dos efeitos de envelhecimento, os autores criaram uma tabela que apresenta as classes de efeitos de envelhecimento, conforme demonstrado em 5. Os efeitos de envelhecimento também são classificados em efeitos voláteis e não-voláteis. Os efeitos voláteis são aqueles que são removidos ao se realizar a reinicialização do sistema, ou então, algum reinicialização de algum processo que esteja sofrendo de envelhecimento de software,

por exemplo, vazamentos de recursos em um sistema operacional. Por sua vez, os efeitos não voláteis persistem mesmo após uma reinicialização, por exemplo, fragmentação de sistemas de arquivos.

Classe	Extensão	Exemplo
Vazamento de Recursos	(1) Específicos de Sistema Operacional (2) Específicos de Aplicação	Memória (1) (2) Sockets (1) Manipulador de Arquivo (1) Processos (1) Threads (1) (2)
Fragmentação	(1) Específicos de Sistema Operacional (2) Específicos de Aplicação	Memória Física (1) Arquivo de Sistema (1) Arquivo de Banco de Dados (2)
Acúmulo de Erros Numéricos	(1) Específicos de Sistema Operacional (2) Específicos de Aplicação	Arredondamento (1) (2)
Acúmulo de Corrupção de Dados	(1) Específicos de Sistema Operacional (2) Específicos de Aplicação	Sistema de Arquivo (1) Arquivo de Banco de Dados (2)

Figura 5 – Classes de efeitos de envelhecimento. Fonte: (GROTTKE; MATIAS; TRIVEDI, 2008)

Conhecer sobre os efeitos de envelhecimento é essencial. Contudo, eles só podem ser percebidos e monitorados através dos denominados **indicadores de envelhecimento**. Esses indicadores funcionam como marcadores para a detecção de envelhecimento de software. Eles usados individualmente ou combinados com outros indicadores podem levar a conclusão se um software está sofrendo de envelhecimento de software. Eles podem ser considerados em níveis como de, por exemplo, sistemas operacionais, processos de aplicação, *middleware*, máquina virtual (VM) e monitor de máquina virtual (VMM).

Assim como os efeitos de envelhecimento, os indicadores de envelhecimento também foram classificados, nesse caso, em duas categorias: indicadores de todo o sistema (conhecidos como *System Wide*) e indicadores de aplicação específica (conhecidos como *Application Specific*). Os indicadores de todo o sistema são aqueles que fornecem informações sobre o sistema como um todo e não apenas sobre uma aplicação específica, por exemplo, memória física livre. Os indicadores de aplicação específica fornecem informações mais detalhadas do que os de todo o sistema, uma vez, que eles avaliam um processo de aplicação individualmente, por exemplo, tempo de resposta da aplicação.

Assim como o envelhecimento de software pode ser causado pela presença de faltas, erros e falhas, há também o *envelhecimento natural*. O envelhecimento natural pode ocorrer mesmo na ausência de **AR-faults**. Eles acontecem por causas de degradações no comportamento natural (sem interferência humana) do funcionamento do software e podem estar relacionados, principalmente, por efeitos de envelhecimento relacionados a fragmentação (memória, sistema de arquivo e arquivo de banco de dados). De acordo com (GROTTKE; MATIAS; TRIVEDI, 2008), considerando além do próprio sistema de

software, mas considerando também o sistema de nível superior, incluindo seus respectivos usuários, pode-se dizer que o envelhecimento natural é devido a erros por parte dos operadores, por exemplo, no caso de problemas de fragmentação que o operador tenha cometido o erro de não executar rotinas de desfragmentação.

As características apresentadas nessa seção mostraram como acontece o fenômeno denominado envelhecimento de software. Salientando as falhas que podem acontecer, os fatores e indicadores de envelhecimento que auxiliam a evitar que tais falhas venham a ocorrer. É sabido que sem intervenções não é possível eliminar e/ou mitigar os efeitos de envelhecimento. Para tanto, técnicas foram desenvolvidas e são discutidas na Seção 3.2 a seguir.

3.2 Rejuvenescimento de Software

Conforme discutido na Seção 3.1, os efeitos de envelhecimento não são eliminados sem a intervenção externa. Essa intervenção acontece através de um conjunto de técnicas que reduzem ou removem os efeitos de envelhecimento. Esse conjunto de técnicas caracterizam o **rejuvenescimento de software**. Em (HUANG et al., 1995) usaram o termo "rejuvenescimento de software" para falar sobre as técnicas utilizadas para mitigação dos efeitos de degradação dos sistemas de software. O rejuvenescimento de software pode manter a solidez dos sistemas de software e evitar interrupções inesperadas do sistema (JIA; SU; CAI, 2008).

O rejuvenescimento de software envolve redefinições no funcionamento do sistema, deixando-o livre de condições de erros acumulados que poderão gerar falhas de software. Este procedimento visa evitar ou adiar uma falha de maneira inesperada, causando perda de dados ou até mesmo grandes consequências negativas. O acionamento do rejuvenescimento de um software pode ser realizado em intervalos, derivados de modelos de sistemas analíticos, que modelam o comportamento de um sistema. Ou então se baseando no monitoramento dos indicadores de envelhecimento.

Rejuvenescer um software pode envolver diversos níveis de granularidade, por exemplo, a níveis de nós físicos, de aplicação, de componentes de aplicação, de virtualização e de sistema operacional (ALONSO et al., 2013). Além disso, o rejuvenescimento de software pode ser aplicado a muitos elementos de um sistema, por exemplo, processos de software individuais ou objetos de dados persistentes (arquivos de metadados de sistemas de arquivos e índice de banco de dados) (GROTTKE; MATIAS; TRIVEDI, 2008).

De acordo com (GROTTKE; MATIAS; TRIVEDI, 2008), alguns pesquisadores questionam se, em vez de mitigar efeitos de envelhecimento, ou seja, realizar o rejuvenescimento, o ideal seria detectar e corrigir os problemas causadores do envelhecimento. Contudo, esse questionamento enfraquece se considerar que, em diversas situações, detectar e eliminar as faltas relacionadas ao envelhecimento não é um procedimento trivial. Rejuvenescer um

software pode ser uma solução ideal quando se tem limitações de técnicas para corrigir defeitos, por questões econômicas ou até mesmo por questões de tempo. Quanto ao envelhecimento natural, onde o envelhecimento acontece de acordo com o comportamento normal do sistema, rejuvenescer o software não se trata de uma opção, mas sim a única solução.

Para elucidar o processo de rejuvenescimento de software, a Figura 6 mostra dois estados de um sistema. Um estado com rejuvenescimento e outro estado sem rejuvenescimento, onde $R1$ representa a taxa de reparo, λ representa a taxa de falha e $R2$ representam a taxas de possível falha no sistema, $R3$ representa a taxa de reparo após rejuvenescimento e $R4$ representa a taxa de rejuvenescimento. De acordo com o ciclo com rejuvenescimento, quando o estado inicial de um sistema ($S0$) é alterado para um possível estado de falha (Sp), um outro caminho pode ser feito em vez de um estado de falha (Sf), onde um novo estado é criado, o estado de rejuvenescimento (Sr). Observa-se que tanto os estado de queda quanto estado de rejuvenescimento, o sistema fica inoperante. A grande diferença é que no estado de rejuvenescimento, a interrupção do sistema é planejada.

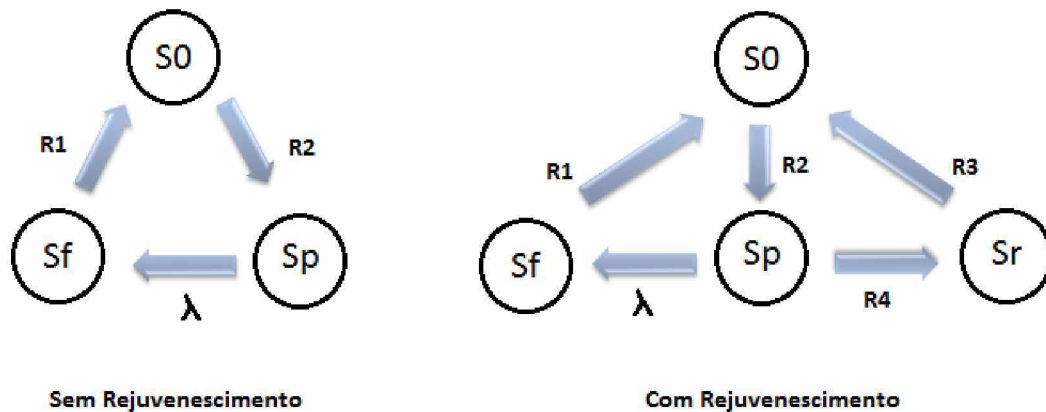


Figura 6 – Software com e sem rejuvenescimento de software.

A decisão sobre o momento ideal para realizar o rejuvenescimento de software é uma grande preocupação para os pesquisadores da área de SAR. As abordagens de rejuvenescimento software podem ser classificadas em dois grupos principais: a estratégias baseadas em tempo e estratégias baseadas em inspeção. Nas estratégias baseadas em tempo o rejuvenescimento é aplicado regularmente e em intervalos de tempo pré-definidos. Essas estratégias são bastante utilizadas em ambientes reais, tais como servidores web (JIA; SU; CAI, 2008). Nas estratégias baseadas em inspeção o rejuvenescimento é realizado de acordo com alguns critérios observados por limiares, predições ou por ambas (mista) em um sistema, realizando o rejuvenescimento de software de acordo o mecanismo escolhido (ALONSO et al., 2013). A Figura 7 mostra a divisão detalhada das abordagens.

Ao detalhar as estratégias baseadas em inspeção, é percebido que no rejuvenescimento

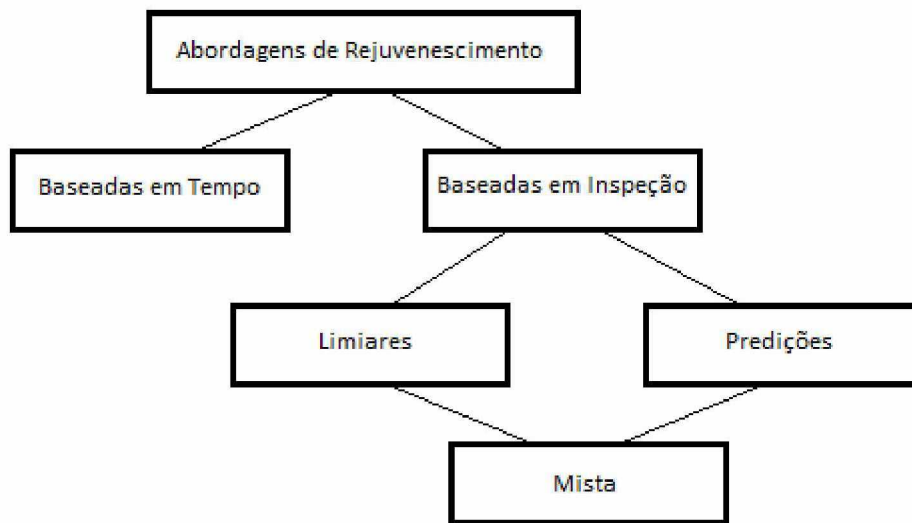


Figura 7 – Abordagens de rejuvenescimento de software.

baseado em limiaries, um especialista estabelece um limite para o sistema, a partir de análise de indicadores de envelhecimento. Quando esse limite é alcançado, o rejuvenescimento é realizado. No caso de rejuvenescimento baseado em previsões, é aplicado um método para prever o tempo gasto até o esgotamento de recursos ou o tempo gasto até a falha provocada pelo envelhecimento de software. Tendo a ciência desse tempo, realiza-se o rejuvenescimento. Para previsão, os métodos como aprendizagem de máquina, abordagens estatísticas ou modelos estruturais são utilizados (MATIAS; FILHO, 2006); (ANDRZEJAK; SILVA, 2008), (ALONSO et al., 2010). Uma abordagem mais recente de estratégias baseada em inspeção é denominada mista. Nesta abordagem, utiliza-se tanto os métodos de previsão quanto método de limiaries para definir quando se realizará o rejuvenescimento (MATOS et al., 2011).

Durante o processo de rejuvenescimento do software, é comum haver um inconveniente que se trata de um tempo de inatividade que ocorre no serviço fornecido pelo software a ser rejuvenescido. Em (GROTTKE et al., 2006), os autores desenvolveram um trabalho focado em reduzir ao máximo esse tempo de inatividade do serviço enquanto o rejuvenescimento é realizado. Por isso, é tão importante não apenas realizar o rejuvenescimento de software afim de não desencadear em falhas de software, mas também programar o momento ideal para esse rejuvenescimento.

Os estudos sobre envelhecimento e rejuvenescimento de software (SAR), com base nas principais ideias introduzidas em (HUANG et al., 1995), aumentaram rapidamente, principalmente nos últimos 10 anos desde a sua introdução. O conceito de SAR atraiu não apenas a área acadêmica, mas também a indústria, resultando em numerosas publicações científicas e patentes registradas. Em geral, os principais temas investigados na pesquisa SAR estão relacionados à detecção (GARG et al., 1998), caracterização (COTRONEO et al., 2010) e modelagem (ZHAO; TRIVEDI, 2011) do fenômeno do envelhecimento de software. Avaliar as diferentes técnicas de rejuvenescimento de software e suas respectivas eficácias também são temas presentes em SAR. Muitas abordagens têm sido propostas e discutidas para diferentes cenários de uso e categorias de software, incluindo os sistemas de cliente e servidor (por exemplo, servidores web (JIA; SU; CAI, 2008), banco de dados (BOVENZI et al., 2012)), sistemas embarcados (KINTALA, 2009) e nos últimos anos sistemas baseados em computação móvel e em nuvem (ARAUJO et al., 2011).

Muitas metodologias foram adotadas para investigações em SAR. Os modelos analíticos têm sido extensivamente utilizados, com base em diferentes métodos, para analisar o comportamento de sistemas, incluindo uma ampla variedade de processos de Markov (por exemplo, CTMC e MRPG), Redes de Petri e Modelos Combinatórios. Além de abordagens analíticas, a pesquisa empírica também está presente na investigação SAR desde o início, incluindo estudos observacionais, estudos exploratórios e experimentos controlados.

Embora o envelhecimento e rejuvenescimento de software seja um campo de pesquisa considerado novo, muito conhecimento já foi produzido. Atualmente, muitas conferências e periódicos incluem temas relacionados a SAR no seu âmbito de interesse. O crescimento rápido da investigação SAR e sua grande variedade de locais de disseminação representam um desafio para os pesquisadores para manter o acompanhamento das novas descobertas e tendências neste campo de pesquisa. Neste trabalho, foram coletados e analisados dados de pesquisa de SAR, utilizando análises para detectar tendências, padrões e lacunas. O objetivo é proporcionar uma visão abrangente das pesquisas nessa área. Para este efeito, foi adotada a abordagem sistemática avaliação Mapeamento Sistemático (MIAN et al., 2005).

Metodologia

Neste capítulo será apresentado o protocolo que foi utilizado na realização deste trabalho. O método para a avaliação adotado para a elaboração deste trabalho seguiu as diretrizes propostas em (MIAN et al., 2005). O mapeamento sistemático é dividido em três fases principais: planejamento, condução e resultados. As fases do mapeamento sistemático são descritas e detalhadas nas subseções a seguir.

4.1 Planejamento

A fase de planejamento de um mapeamento sistemático consiste em definir como os passos do protocolo (MIAN et al., 2005) serão executados. Os detalhes da fase de planejamento do mapeamento sistemático realizado nesta pesquisa são detalhadas na subseções a seguir.

4.1.1 Questões de Pesquisa

Especificar as questões de pesquisa é considerada a parte mais importante de qualquer mapeamento sistemático. Elas auxiliam a condução de toda a metodologia do mapeamento sistemático, uma vez que, respondidas elas permitem a geração de informações relevantes sobre envelhecimento e rejuvenescimento de software (SAR). As questões de pesquisa desta dissertação são mostradas a seguir.

1. **Quantos estudos foram específicos para envelhecimento de software e para rejuvenescimento de software?**

Tradicionalmente os estudos na área de SAR não fazem distinção entre rejuvenescimento de software e envelhecimento de software. Por isso, surgiu a questão de pesquisa acima para saber se os estudos na área de SAR estão mais voltados para envelhecimento de software ou para rejuvenescimento de software.

2. Qual é o tipo mais prevalente de estudo na amostra de trabalho?

Os estudos científicos em geral são de três tipos: empíricos, analíticos ou híbridos. Nos estudos empíricos, os fatos científicos são determinados por meio de trabalhos experimentais. Nos estudos analíticos, os fatos científicos são determinados através de análises sem estudos experimentais. Por fim, os estudos híbridos (termo utilizado neste trabalho) são estudos onde se realizam análises e experimentos em um mesmo trabalho. Dessa forma, a questão de pesquisa acima buscou saber quais dos três tipos de estudo são mais prevalentes na amostra obtida.

3. Quais são as técnicas de modelagem mais prevalentes nos estudos?

Em envelhecimento e rejuvenescimento de software várias técnicas de modelagem são utilizadas para verificar o comportamento do sistema. Essas técnicas ajudam os investigadores a compreender os fenômenos de envelhecimento e rejuvenescimento de software. Elas são utilizadas para caracterização, simulação e previsão de envelhecimento de software. Além disso, no que diz respeito ao rejuvenescimento de software, essas técnicas podem ser utilizadas para a avaliação e programação e seus respectivos custos do rejuvenescimento. O objetivo dessa questão de pesquisa foi detectar quais são as técnicas de modelagem mais utilizadas nos estudos da amostra.

4. Quais foram as classes de efeitos de envelhecimento mais investigados? Quais foram os efeitos mais comuns destas classes?

Quando um software está sofrendo de envelhecimento, ele apresenta algumas características. Essas características são denominadas "efeitos de envelhecimento". Sendo eles tão importantes para o entendimento sobre envelhecimento de software, essa questão de pesquisa objetivou contabilizar o estudos que trataram sobre efeitos de envelhecimento e as suas classes (GROTTKE; MATIAS; TRIVEDI, 2008).

5. Quais foram os indicadores de envelhecimento mais utilizados?

Quando é necessário confirmar a presença ou ausência do fenômeno envelhecimento em um software, é essencial a utilização dos indicadores de envelhecimento. Esses indicadores atuam como medidores. Com isso, a questão acima buscou contabilizar quantos estudos utilizaram esses indicadores de envelhecimento em suas pesquisas, mostrando quais são os indicadores mais utilizados.

6. Quais técnicas de rejuvenescimento têm sido usadas com mais frequência?

As técnicas de rejuvenescimento são importantes restauradoras do estado interno do software. Muitos estudos buscam utilizar e melhorar essas técnicas. Baseado nisso, a questão acima buscou responder quais foram as técnicas de rejuvenescimento mais utilizadas pelos pesquisadores em seus respectivos trabalhos para realizar o rejuvenescimento de software.

7. Quais estratégias de rejuvenescimento foram mais prevalentes entre os estudos?

As estratégias de rejuvenescimento orientam pesquisadores sobre como realizar o rejuvenescimento de software. Conforme apresentado Capítulo 3, essas estratégias podem ser baseadas em tempo ou em inspeção. Baseado nisso, a questão acima visou responder quais foram as estratégias têm sido mais utilizadas pelos pesquisadores.

8. Quais foram as plataformas de sistemas operacionais mais prevalentes?

Os estudos na área de SAR, especialmente os dos tipos empíricos e híbridos, são realizados utilizando algum sistema operacional como instrumento de pesquisa, e não como objeto de estudo, para executarem os seus trabalhos. Com isso, a questão de pesquisa acima teve como objetivo saber quais foram os sistemas operacionais mais utilizados pelos pesquisadores em seus trabalhos.

4.1.2 Estratégia de Busca de Estudos Primários

A estratégia de busca consiste na abordagem utilizada para a construção da amostra para a realização do mapeamento sistemático. Neste trabalho, a estratégia foi orientada por especialistas na área de SAR. Foi realizada uma consulta aos especialistas a 13 investigadores experientes em SAR e de diferentes países (Brasil, China, Alemanha, Índia, Itália, Japão, Taiwan e Estados Unidos da América), apresentados na Tabela 2. Essa consulta objetivou resgatar as sugestões desses especialistas sobre veículos de publicação onde se deveria procurar material para compor a amostra utilizada.

Dos especialistas consultados, quatro (30,76% dos especialistas) sugeriram veículos de publicação para coleta dos estudos, a partir de 1995, ano em que o tema de SAR foi cunhado. As sugestões de todos os especialistas foram unidas e apresentadas na Tabela 3 para aqueles veículos que são conferências e na Tabela 4 para aqueles veículos que são periódicos.

Tabela 2 – Relação de especialistas consultados e seus respectivos locais de trabalho

Especialista	Afiliação
Alberto Avritzer	Sonatype, McLean, VA, EUA
Antonio Puliafito	Università di Messina, Messina ME, Itália
Artur Andrzejak	Institute of Computer Science, Heidelberg
Fumio Machida	NEC Laboratory for Analysis of System Dependability (LASD), Minato-ku, Tóquio, Japão
Hiroyuki Okamura	Hiroshima University, Hiroshima, Japão
Kai-Yuan Cai	Department of Automatic Control, Beijing University of Aeronautics and Astronautics, China
Kalyan Vaidyanathan	Oracle Corporation, Redwood City, EUA
Katinka Wolter	Institut für Informatik, Fachbereich Mathematik und Informatik, Berlim, Alemanha
Kishor S. Trivedi	Department of Electrical and Computer Engineering, Duke University, Durham, NC, EUA
Michael Grottke	Friedrich-Alexander University, Nürnberg, Alemanha
Rivalino Matias	Faculdade de Computação, Universidade Federal de Uberlândia, Minas Gerais, Brasil
Tadashi Dohi	Hiroshima University, Hiroshima, Japão
Yennun Y. Huang	Academia Sinica, Taipei, Taiwan

Tabela 3 – Lista de conferências indicadas pelos especialistas

Nome da Conferência
Conference Object-Oriented Programming Systems, Languages and Applications (OOPSLA)
Conference on Programming Language Design and Implementation (PLDI)
International Computer Software and Applications Conference (COMPSAC)
International Conference on Availability, Reliability and Security (ARES)
International Conference on Dependable Systems and Networks (DSN)
International Conference on Software Engineering (ICSE)
International Symposium on Empirical Software Engineering (ISESE)
International Symposium on Fault-Tolerant Computing (FTCS)
International Symposium on Software Reliability Engineering (ISSRE)
International Workshop on Software Aging and Rejuvenation (WoSAR)
Pacific Rim International Symposium on Dependable Computing (PRDC)
Symposium on Reliable Distributed Systems (SRDS)
Workshop on Software and Performance (WOSP)

Tabela 4 – Lista de periódicos indicados pelos especialistas

Nome do Periódico
ACM Computing Surveys (CSUR)
ACM Journal on Emerging Technologies in Computing Systems (JETC)
IBM Systems Journal
IEEE Computer
IEEE Software
IEEE Transactions on Dependable and Secure Computing (TDSC)
IEEE Transactions on Reliability (TR)
IEEE Transactions on Software Engineering (TSE)
Journal of Systems and Software (JSS)
Performance Evaluation (PEVA)
Software - Practice and Experience (SPE)

Os repositórios digitais utilizados na coleta dos estudos primários foram selecionados considerando a relevância e a popularidade de cada um. A partir desses repositórios, foram localizados os veículos de publicação que os especialistas sugeriram. Em seguida, foi realizada a busca de todos os estudos que tratavam sobre a área de envelhecimento e rejuvenescimento de software. Com o intuito de saber se um estudo era, de fato, sobre envelhecimento ou rejuvenescimento de software, foram analisados o resumo, as palavras chaves e o texto de cada estudo.

□ Repositórios Digitais

IEEE Xplore

ACM Digital Library

Science Direct

4.1.3 Critérios de Seleção

Os critérios de exclusão eliminam estudos que não são de interesse para a pesquisa. No caso desta pesquisa, foram desconsiderados os estudos não continham *informações detalhadas* sobre a área de SAR que os tornassem relevantes para serem inseridos na amostra. Os critérios de exclusão estabelecidos foram: *short paper's*, estudos secundários, terciários, textos de apresentação.

Após a definição dos critérios de exclusão, são definidos os critérios de inclusão. Esses critérios determinam que os estudos a serem incluídos na amostra devem ser, obrigatoriamente, sobre o tema investigado. No caso desta pesquisa, foram inseridos os estudos sobre envelhecimento de software e/ou Rejuvenescimento a partir do ano de 1995, ano em que o tema foi cunhado e com *informações suficientes* para ser considerado como um estudo sobre SAR. Foi utilizado também um procedimento conhecido como *Snowballing* (WOHLIN, 2014). Assim, as referências bibliográficas dos estudos foram analisadas e, quando

já não presentes na amostra, foram adicionados desde que respeitassem os critérios de inclusão.

4.2 Condução

Uma vez que a fase de planejamento do mapeamento sistemático esteja concluída. A fase de condução pode ser iniciada. A subseções a seguir mostram os detalhes de como essa fase foi executada. A amostra resultou em 188 estudos.

4.2.1 Procedimento de Seleção de Estudo

A seleção dos estudos foi um processo de dois estágios para coletar os estudos da amostra. Os estudos dos veículos indicados pelos pesquisadores (Tabela 2) foram coletados e os critérios de exclusão e inclusão foram aplicados. O resultado disso foi uma amostra inicial com 146 estudos. Em seguida, todos os 146 estudos da amostra tiveram as suas referências bibliográficas analisadas. Essa análise foi realizada através de um programa desenvolvido na linguagem de programação Java que consistiu em abrir cada arquivo de cada estudo, detectando e contabilizando as referências de cada estudo. Os estudos mais citados que ainda não estavam na amostra foram coletados. Todos esses estudos foram selecionados conforme os critérios estabelecidos, com uma amostra final de 188 estudos.

4.2.2 Estratégia de Extração de Dados

Após a amostra estar completa, contendo 188 estudos sobre SAR, foi realizada a extração de dados. Essa extração consistiu em ler os estudos, registrando os dados obtidos em cada um desses estudos. Esses dados foram registrados através da criação de **tags** nos próprios estudos. Essas **tags** marcaram os estudos sobre o que exatamente eles se tratavam na área de SAR e tiveram o intuito de melhorar a identificação de cada um deles.

4.2.3 Síntese dos Dados Extraídos

Após a extração dos dados (4.2.2), foi necessário recolher os dados obtidos, contabilizá-los, produzindo informações que pudessem ser úteis para confirmar a hipótese (1.3) deste trabalho de dissertação.

4.3 Resultados

As duas primeiras fases principais do mapeamento sistemático foram realizadas, restando apenas a realização da fase final. Essa fase envolveu escrever os resultados da

avaliação e divulgação dos resultados a potenciais interessados.

❑ **Escrita de Artigo**

❑ **Escrita da Dissertação**

❑ **Defesa da Dissertação**

Os resultados obtidos no mapeamento sistemático são divulgados nesta dissertação de mestrado, mais precisamente no Capítulo 5, a seguir. Divulgou-se também os resultados obtidos neste trabalho em uma conferência específica da área de envelhecimento e rejuvenescimento de software, o WoSAR (Workshop on Software Aging and Rejuvenation) no ano de 2016.

Resultados e Análises

A aplicação do protocolo apresentação no Capítulo 4 resultou em uma amostra de 188 estudos primários sobre envelhecimento e rejuvenescimento de software. A lista total desses estudos está disponível no Apêndice A.1 neste trabalho. Cada seção a seguir está relacionada a uma questão de pesquisa do mapeamento sistemático que foi realizado, retomando-as em negritos com os seus respectivos resultados e análises.

5.1 Temas dos Estudos

Entre os 188 estudos primários selecionados, realizou-se a separação desses estudos por temas de estudos, ou seja, por envelhecimento de software ou por rejuvenescimento de software. Desses estudos, **quantos estudos foram específicos tanto para o envelhecimento e para o rejuvenescimento?** Foram encontrados 86 (45,74% de todos os estudos da amostra) estudos voltados para o envelhecimento de software e 102 (54,26% de todos os estudos da amostra) estudos se concentraram em rejuvenescimento de software. O critério utilizado para definir o tema de um estudo foi a análise da predominância desse tema em cada um dos estudos primários. Os estudos que geraram dúvidas quanto a sua predominância, foram analisados por outros pesquisadores participantes do mapeamento sistemático para definir se o estudo era sobre envelhecimento ou rejuvenescimento de software.

A Figura 8 mostra a distribuição destes estudos ao longo do período investigado. Nessa figura foram definidos quatro períodos distintos: os primeiros cinco anos (1995 a 1999), onde o número de estudos é menor quando comparado ao número de estudos dos períodos seguintes. O segundo período e terceiro período, que abrangem de 2000 a 2004 e 2005 a 2009, respectivamente, houve um aumento gradual e constante no número de estudos. Nota-se que em 2009, ano em que não houve edição do WoSAR (International Workshop on Software Aging and Rejuvenation), o número de estudos foi consideravelmente menor, deixando claro quão importante é esse evento para a pesquisa na área de SAR. O terceiro período, variando de 2010 a 2014, apresentou a maior atividade na história da pesquisa em

SAR. Em todos os períodos houve uma série bem equilibrada relativa de estudos focados em envelhecimento (45,74%) e rejuvenescimento (54,26%).

O desenvolvimento desta pesquisa teve início no ano de 2015. Contudo, é importante ressaltar que os estudos referentes ao ano de 2015 não fizeram parte da amostra obtida. Considerando o período em 2015 em que os estudos foram coletados, caso o ano de 2015 tivesse sido considerado, os resultados poderiam não condizer com a realidade devido a indisponibilidade de alguns estudos.

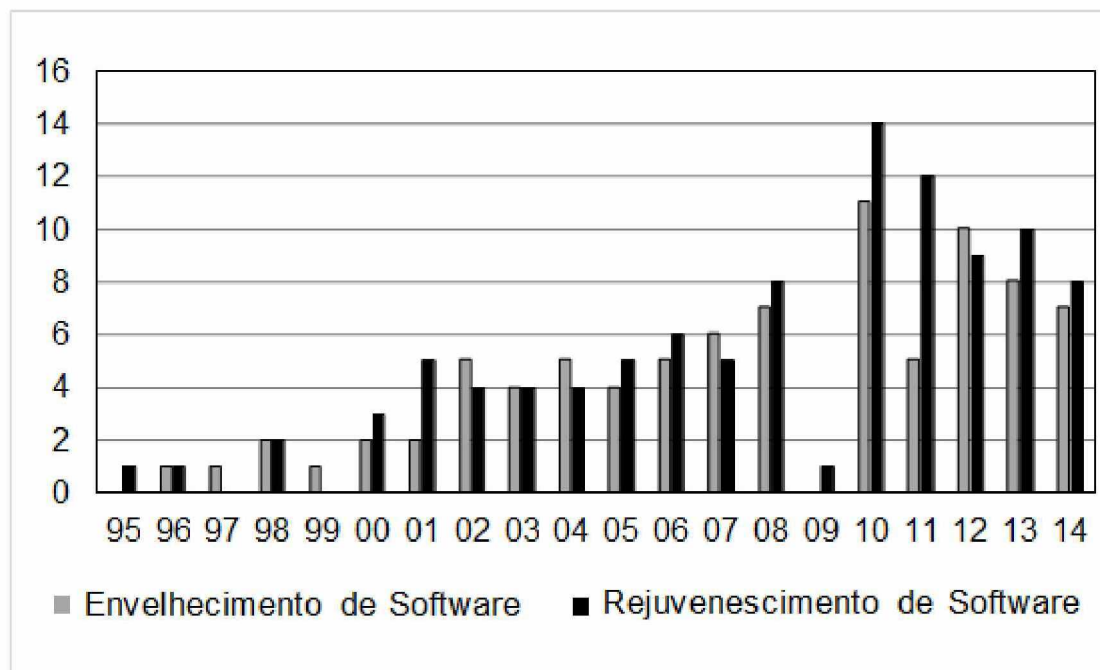


Figura 8 – Distribuição dos estudos ao longo dos anos.

Realizou-se também a análise da distribuição de estudos por diferentes autores a cada cinco anos. O objetivo da Tabela 5 é apresentar dados sobre a quantidade de estudos e de autores. Essa tabela mostra períodos que foram definidos a partir da observação do gráfico da Figura 8. A coluna "Novos Autores" mostra a quantidade de autores que não possuem nome em qualquer estudo dos períodos anteriores.

Considerando os 9 estudos do período 1995 a 1999, foram detectados 21 autores. Como esse é o primeiro período, então todos os 21 autores foram considerados como novos. Considerando os 38 estudos do período entre 2000 a 2004, foi contabilizado 43 autores. Entre esses autores, 39 não constam na relação de autores do período 1995 a 1999. Assim, o período de 2000 a 2004 experimentou um aumento de novos autores de 65% ($39/(39+21)$). O período 2005 a 2009 experimentou um aumento de novos autores de 53,12% ($68/(68+39+21)$). E, por fim, o período de 2010 a 2014 teve um aumento de 47,10% ($114/(114+68+39+21)$).

Os resultados disso é que, em média, a cada cinco anos foi observado um aumento de mais de 54% no número de investigadores. Os 188 estudos analisados foram de autoria de

Tabela 5 – Distribuição de autores por ano

Período	Estudos	Autores	Novos Autores
1995-1999	9	21	21 (100%)
2000-2004	38	43	39 (65%)
2005-2009	47	70	62 (53,12%)
2010-2014	94	136	114 (47,10%)

236 autores diferentes, onde a cada novo período, o número de autores foi crescente. Esta descoberta é uma evidência de que a comunidade de envelhecimento e rejuvenescimento de software está em expansão.

5.2 Tipos de Estudo

Na amostra coletada, os estudos foram classificados em três tipos: analítico, empírico ou híbrido. Os estudos do tipo analítico não incluem qualquer atividade experimental/empírica relacionada. Os estudos do tipo empíricos são opostos aos do tipo analítico. Por fim, os estudos do tipo híbrido envolvem ambas as atividades (analíticas e empíricas). Com isso, a seguinte pergunta surgiu: **Qual é o tipo mais prevalente de estudo na amostra de trabalho?** A resposta a esta questão é apresentada na Tabela 6, que mostra as porcentagens de tipos de estudo por tema de estudo (envelhecimento ou rejuvenescimento de software).

Tabela 6 – Porcentagem de tipos de estudo por temas de estudo

Tema	Analítico	Empírico	Híbrido
Envelhecimento	35 (40,70%)	36 (41,86%)	15 (17,44%)
Rejuvenescimento	53 (51,96%)	32 (31,37%)	17 (16,67%)

Como pode ser visto, os estudos analíticos foram, em geral, os mais prevalentes com 46,81% (88 de 188 estudos). Os estudos empíricos foram prevalentes (por apenas um estudo) em envelhecimento de software. Em rejuvenescimento de software, o número de estudos foi consideravelmente menor (21 estudos a menos) que os estudos analíticos. Os estudos híbridos foram os menos prevalentes na amostra, contudo, apresentam aumento gradativo ao longo dos anos. Por isso, foi importante analisar a distribuição desses estudos ao longo dos anos, que são apresentados na Figura 9 para envelhecimento de software e na Figura 10 para rejuvenescimento de software, respectivamente.

Ainda que a maioria de todos os estudos em todo o período tenham sido analíticos, o número de estudos empíricos e estudos híbridos aumentaram ao longo dos anos. Na Figura 9, por exemplo, foi observado que o número total de estudos empíricos e híbrido de envelhecimento de software de 2003 a 2014, representou 63,89% (46 de 72 estudos) dos estudos no período. Por outro lado, ao considerar os estudos de rejuvenescimento

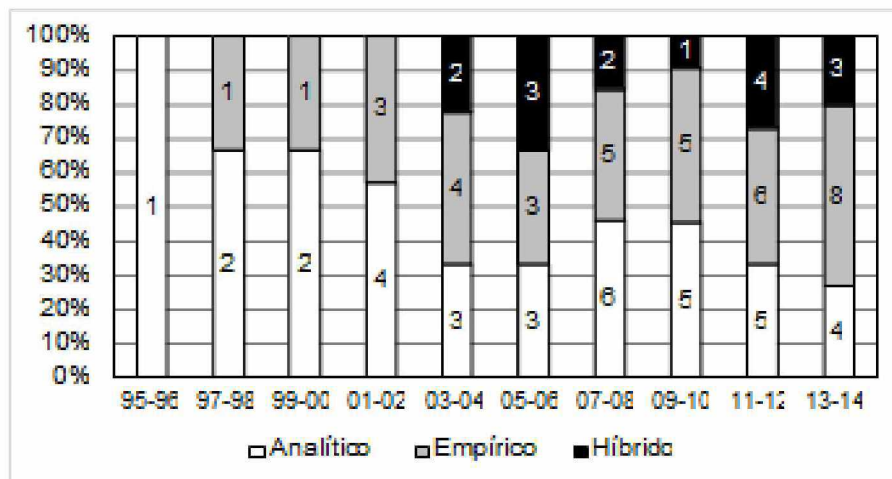


Figura 9 – Distribuição de estudos de envelhecimento por tipo de estudo.

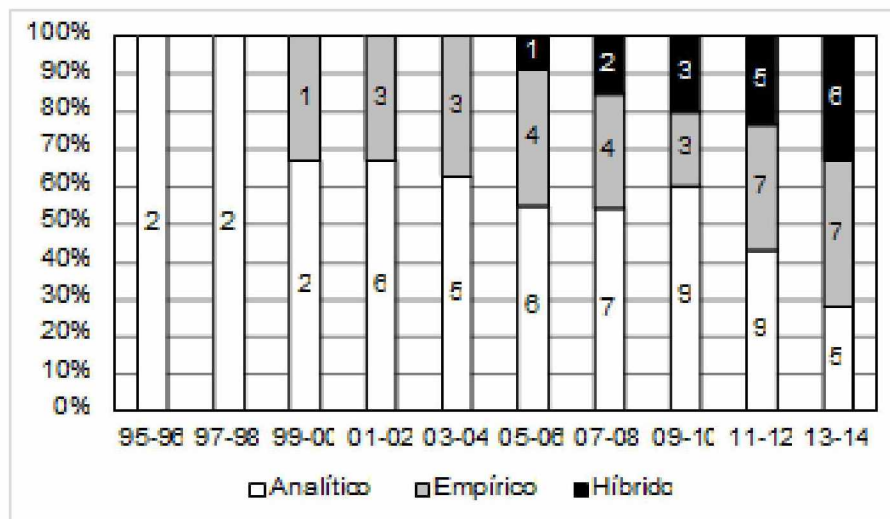


Figura 10 – Distribuição de estudos de rejuvenescimento por tipo de estudo.

de software (Figura 10), a porcentagem de 64,10% dos estudos empíricos e híbridos só ultrapassa os estudos analíticos a partir de 2011.

Esses resultados sugerem uma tendência no desenvolvimento de estudos empíricos ou híbridos, tanto de envelhecimento de software quanto de rejuvenescimento de software. Essa tendência pode ser observada com um aumento visível em estudos sobre rejuvenescimento de software, do tipo híbrido, nos últimos dez anos.

5.3 Técnicas de Modelagem

As técnicas de Modelagem desempenham um importante papel que consiste em ajudar os investigadores a compreender os fenômenos de envelhecimento e rejuvenescimento de software. Estas técnicas têm sido utilizadas para fins diferentes, tais como: caracterização,

simulação e previsão de envelhecimento de software. Além disso, técnicas de modelagem podem ser utilizadas para a avaliação e programação de rejuvenescimento de software e seus respectivos custos.

Assim, foi investigada a seguinte questão de pesquisa: **Quais são as técnicas de modelagem mais prevalentes nos estudos?** Analisando os estudos que tratam sobre técnicas de modelagem, descobriu-se que 65,42% (123 de 188 estudos) de todos os estudos analisados se referem à técnicas de modelagem. Observou-se que estes estudos foram envolvidos com uma ou mais das seguintes técnicas: processos de Markov e suas variantes (DTMC, MRGP, PMS, MDP); redes de Petri e suas variantes (SPN, SRN, GSPN); análise de ajuste de curva, que abrange técnicas como regressão linear e não-linear, os modelos de séries temporais (ARMA, ARIMA, ARX, Holt-Winters), funções lineares, Mann-Kendall e outras análises de tendências.

A Tabela 7 apresenta o número de estudos por técnicas de modelagem e por tema de estudo. Os valores percentuais se referem ao total de estudos relacionados à técnicas de modelagem (123 estudos). Processos de Markov (45 estudos) e análise de ajuste de curva (47 estudos) são as técnicas prevalentes tanto em estudos de envelhecimento de software quanto em estudos de rejuvenescimento de software, seguido por redes de Petri (31 estudos). Embora o número de estudos relacionados com redes de Petri e análise de ajuste de curva tenha ficado bem equilibrado entre estudos envelhecimento e rejuvenescimento de software, o uso de processos de Markov foi mais proeminente nos estudos de rejuvenescimento de software.

Tabela 7 – Técnicas de modelagem mais usadas por tema de estudo.

Técnica de Modelagem	Envelhecimento	Rejuvenescimento
Cadeia de Markov e Variações	19 (15,45%)	26 (21,14%)
Redes de Petri e Variações	14 (11,38%)	17 (13,82%)
Análise de Ajuste de Curva	23 (18,70%)	24 (19,51%)

Ainda relacionado à técnica de modelagem, foi realizada a distribuição de estudos relacionados a técnicas de modelagem ao longo dos anos, conforme a Figura 11. Destaca-se que as técnicas de análise de ajuste de curva (47 estudos) foram observadas somente a partir de 2003-2004 em diante, na maioria das vezes elas ficaram em desvantagem em relação a processos de Markov (38 estudos a partir de 2003) e redes de Petri (23 estudos a partir de 2003), presentes desde 1997.

A Figura 12 apresenta a distribuição das técnicas de modelagem por tipo de estudo ao longo dos anos. Como esperado, devido a técnicas de modelagem estarem relacionadas a análises, os estudos analíticos (92 estudos) foram os mais numerosos. Em menor número, os estudos híbridos (31 estudos). Não houve incidência de estudos empíricos que envolvessem técnicas de modelagem. Mesmo os estudos analíticos sendo a maioria, nota-se um padrão de tendência de crescimento no número de estudos híbridos, o que é condizente

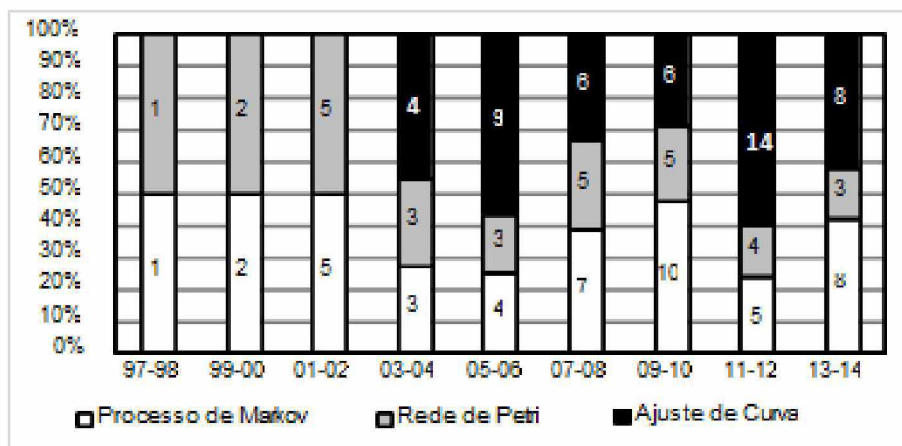


Figura 11 – Distribuição de estudos relacionados com técnicas de modelagem.

com os resultados apresentados na Figura 11. Com base nessas evidências, é notória uma tendência na pesquisa em SAR para estudos híbridos sobre técnicas de modelagem.

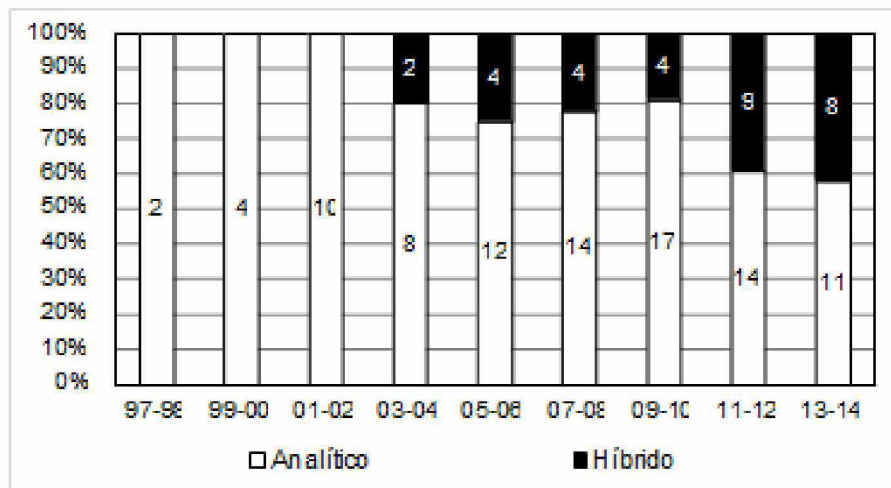


Figura 12 – Distribuição de estudos relacionados com técnicas de modelagem e o tipo de estudo.

5.4 Efeitos de Envelhecimento

Conforme introduzido no Capítulo 3, uma aplicação que está sofrendo de envelhecimento software é afetada por efeitos de envelhecimento. Esses efeitos de envelhecimento podem ser classificados de acordo com o proposto em (GROTTKE; MATIAS; TRIVEDI, 2008). Essas classes são: vazamento de recursos, fragmentação, erro numérico e corrupção de dados, conforme mostrado na Figura 5 no Capítulo 3. Por isso, foram respondidas as seguintes perguntas: **Quais foram as classes de efeitos de envelhecimento mais investigados?** **Quais foram os efeitos mais comuns destas classes?** Foram encon-

trados 64 estudos (10 analíticos, 38 empíricos e 16 híbridos) relacionados com os efeitos de envelhecimento mencionados. A Figura 13 apresenta a distribuição destes estudos ao longo dos anos. Os anos sem incidências no gráfico são de anos que não constava na amostra estudos que tratassem sobre efeitos de envelhecimento. Como pode ser visto, estudos empíricos e híbridos foram predominantes, uma vez que efeitos de envelhecimento se encaixam em estudos que realizam experimentos.

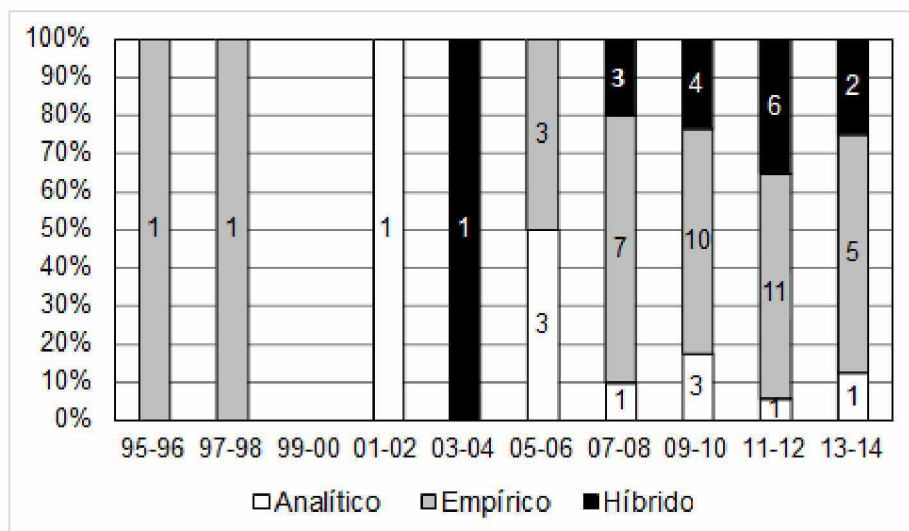


Figura 13 – Distribuição de estudos sobre envelhecimento efeitos por tipo de estudo.

Dado o corpo de conhecimento sobre os efeitos de envelhecimento construído em vinte anos de pesquisa na área de SAR, foi observada uma tendência no desenvolvimento de estudos híbridos que investigam diferentes aspectos dos efeitos de envelhecimento. Uma oportunidade de pesquisa poderia ser o desenvolvimento de algoritmos para reconhecimento de efeitos de envelhecimento, similar ao que foi proposto em (BARADA; SWAIN, 2014). Analisou-se também a forma como as classes de efeitos de envelhecimento apareceram nos estudos de diferentes tipos, mostradas na Figura 14.

Observou-se que classes de **vazamento de Recursos** e **fragmentação** foram as mais recorrentes. Especificamente, sobre os estudos relacionados a vazamento de recursos (44 estudos) trataram dos seguintes recursos: memória principal, manipuladores de arquivos e processos. A Figura 15 apresenta a ocorrência desses recursos entre os estudos de vazamento de recursos por tipo de estudo.

Foram encontrados 39 estudos relacionados vazamento de memória e este número excede o total de todos os estudos para outros recursos. Por isso, foi analisada a distribuição ao longo dos anos de estudos relacionados a vazamento desse recurso e mostrado na Figura 16. Estudos sobre vazamento de memória não constavam na amostra nos períodos de 1997 a 2000 e de 2003 a 2004. A partir de 2007, o período mais prolífico no que diz respeito aos estudos de vazamento de memória, analisando o tipo desses estudos, foi observado que os

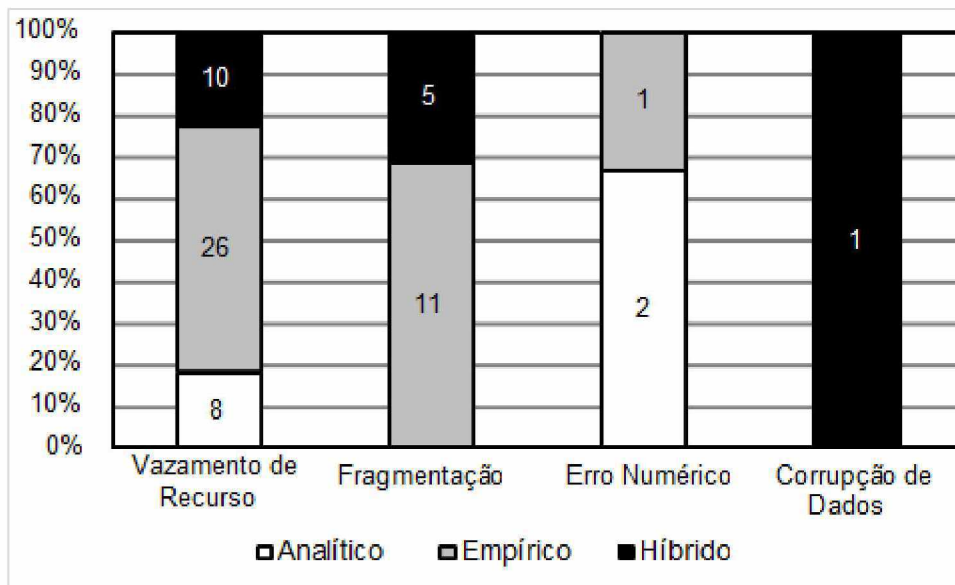


Figura 14 – Porcentagem de classes de efeitos de envelhecimento por tipo de estudo.

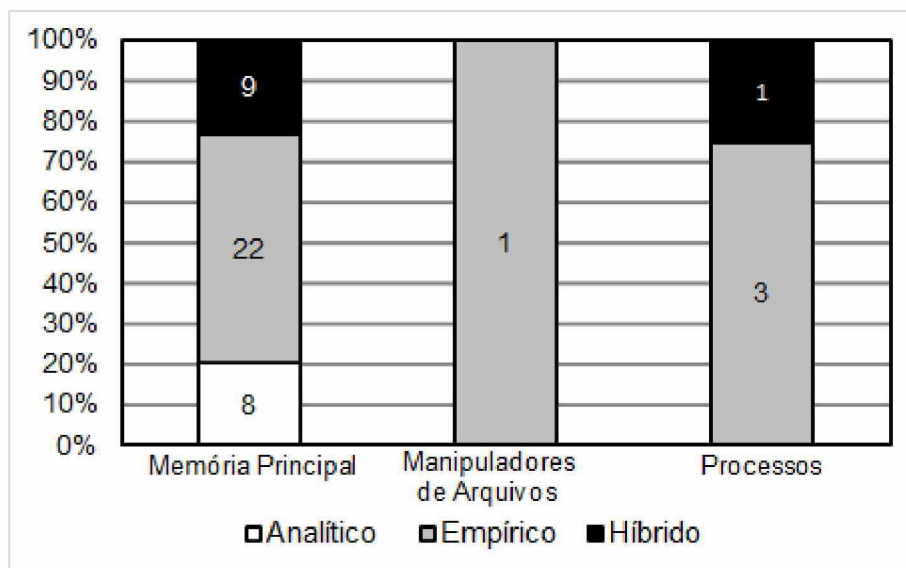


Figura 15 – Recursos investigados por tipo de estudos relacionados com vazamento de recurso.

estudos empíricos predominaram. Contudo, os estudos do tipo híbrido também estão se tornando uma tendência nos trabalhos de pesquisa sobre o vazamento de memória.

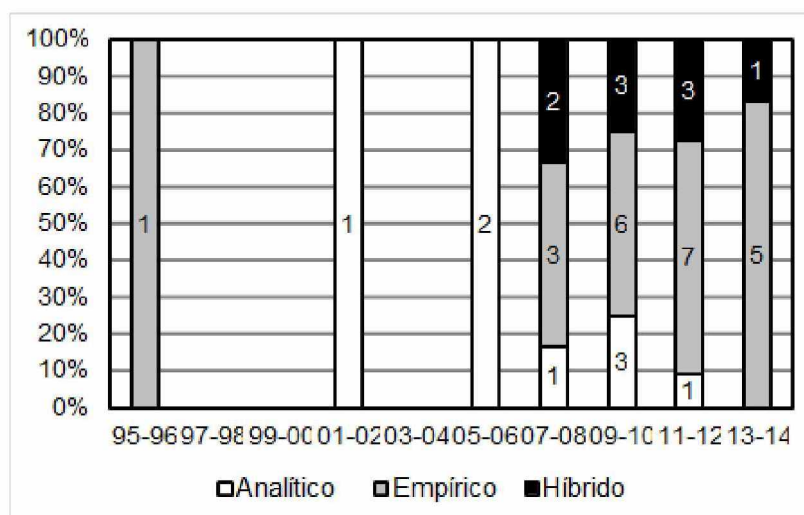


Figura 16 – Distribuição de estudos relacionados com o vazamento de memória.

Em seguida, a mesma análise foi realizada sobre os trabalhos relacionados à fragmentação e foram encontrados estudos que abordam os seguintes recursos: memória principal, sistema de arquivos e arquivos de banco de dados. A Figura 17 mostra este resultado. A Figura 18 mostra a distribuição desses estudos sobre fragmentação ao longo dos anos. Na amostra esses estudos apareceram a partir de 2003. Nota-se que a fragmentação da memória apareceu em praticamente todo o período, exceto em 2013-2014.

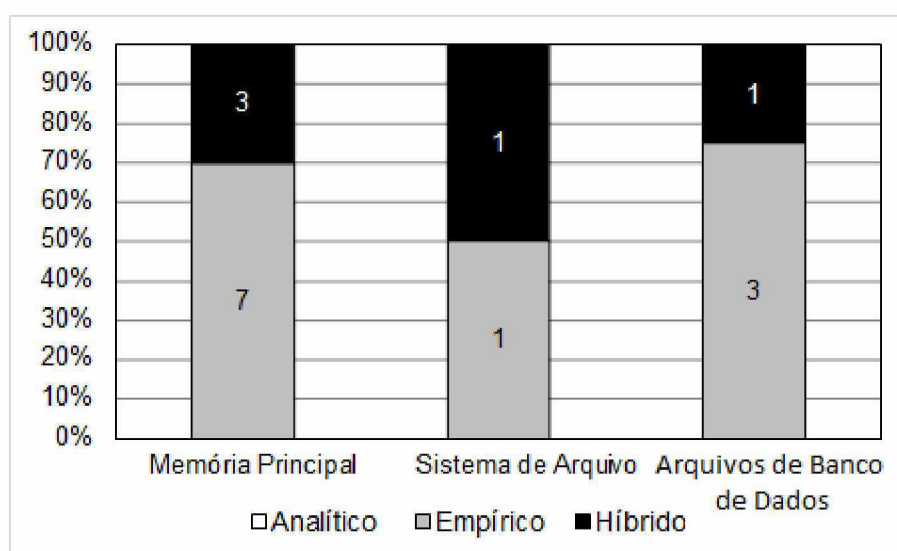


Figura 17 – Recursos investigados por tipo de estudo relacionados com fragmentação.

Semelhante a fragmentação de sistema de arquivo, os estudos sobre arquivos de banco de dados aparentam ser promissores como oportunidades de investigação, dado o intenso desenvolvimento de grandes aplicações de dados. Isso pode levar a descoberta de novos problemas relacionados com o fenômeno de envelhecimento de software em banco de dados. Em relação às demais classes, isto é, acúmulo de erros numéricos e corrupção de

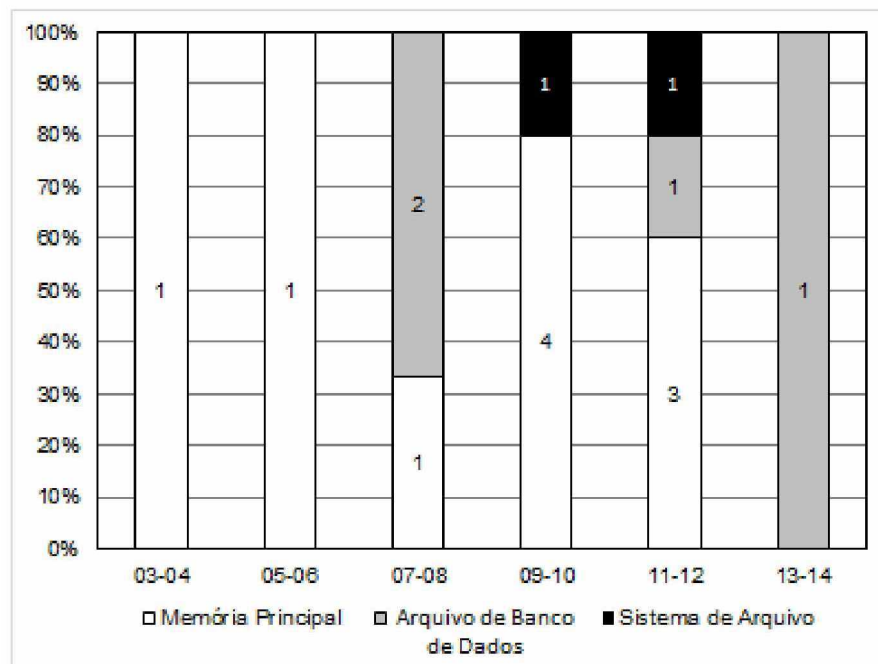


Figura 18 – Distribuição de estudos ao longo dos anos sobre fragmentação

dados, todos os três estudos relacionados ao erro numérico investigaram o problema de arredondamento, onde um deles é do tipo empírico e os outros são do tipo analítico. O estudo sobre corrupção de dados abordaram sistema de arquivos (um estudo híbrido) e arquivos de banco de dados (dois estudos empíricos).

5.5 Indicadores de Envelhecimento

No Capítulo 3, foi falado que os efeitos do envelhecimento de software são medidos por meio de indicadores de envelhecimento. Descobrir um conjunto eficiente e eficaz dos melhores indicadores de envelhecimento é um importante tópico para pesquisa. A qualidade dos indicadores de envelhecimento influencia diretamente a precisão do tempo com o qual o rejuvenescimento é acionado. Esses indicadores podem determinar custos (por exemplo, tempo de inatividade durante o rejuvenescimento) e benefícios (por exemplo, o tempo de inatividade evitada por falhas inesperadas) do rejuvenescimento de software (GROTTKE; MATIAS; TRIVEDI, 2008).

Desde 1995, diferentes indicadores de envelhecimento têm sido propostos e utilizados, por isso, espera-se responder à seguinte questão de pesquisa: **Quais foram os indicadores de envelhecimento mais utilizados?** Encontrou-se 64 estudos que experimentaram pelo menos um dos seguintes indicadores de envelhecimento: memória física livre (Mem), espaço SWAP (Swap), *Resident Set Size* (RSS), Heap utilizada (HU), *JVM Heap Size* (JVMHS), taxa de falhas (FRate), tempo de resposta (RTime), a carga do sistema (Load), CPU usado (CPU), tamanho da tabela de arquivo (FTSize) e tamanho

da memória virtual (VMSize).

A Tabela 8 mostra o **número** e a **porcentagem** de ocorrências dos indicadores de envelhecimento acima que apareceram na amostra. Considerando os temas de estudo, os indicadores de envelhecimento foram encontrados em 48 estudos relacionados com o envelhecimento de software e 16 estudos relacionados com o rejuvenescimento de software. A memória física livre foi usada em 26 dos 64 estudos (representando 40,62%). Com isso, é notado que um mesmo indicador de envelhecimento pode ter aparecido em diferentes estudos.

A Figura 19 apresenta a distribuição dos indicadores de envelhecimento ao longo dos anos. É possível observar como foi a evolução dos indicadores de envelhecimento dos tipos *System Wide* e *Application Specific* ao longo do anos.

Tabela 8 – Ocorrência de indicadores de envelhecimento

Indicadores de Envelhecimento	Ocorrências
Memória Física Livre (Mem)	26 (40,62%)
Espaço SWAP (Swap)	18 (28,12%)
JVM Heap Size (JVMHS)	12 (18,75%)
Resident set size (RSS)	11 (17,18%)
Taxa de Falha (FRate)	8 (12,50%)
Tempo de Resposta (RTime)	8 (12,50%)
Carga de Sistema (Load)	7 (10,93%)
Heap utilizada(HU)	5 (7,81%)
Tamanho de Memória Virtual (VMsize)	4 (6,25%)
CPU Usado (CPU)	3 (4,68%)
File Table Size (FTsize)	1 (1,56%)

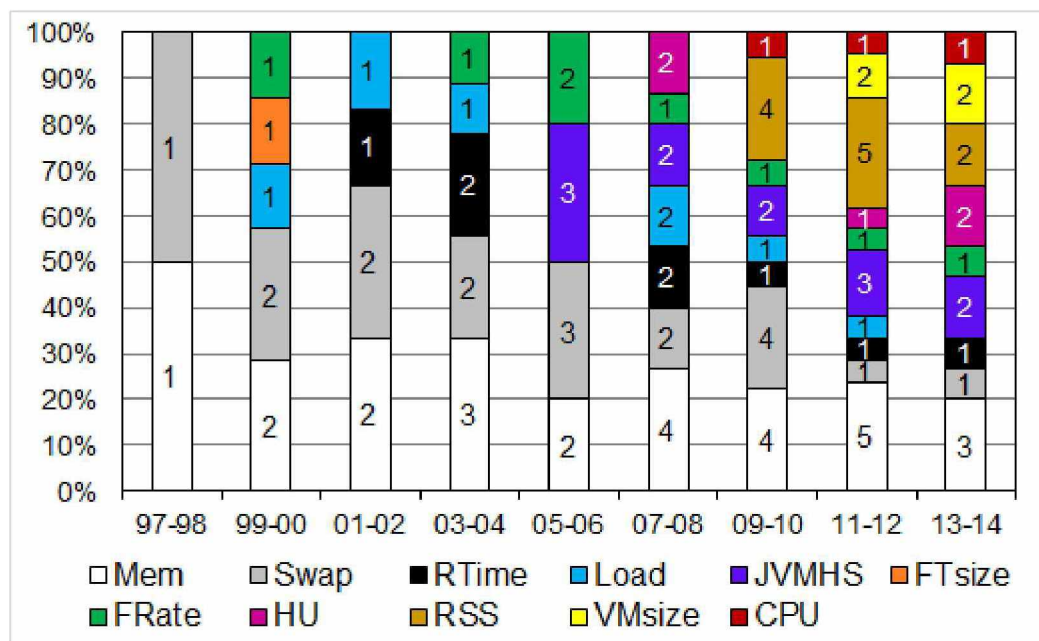


Figura 19 – Distribuição dos estudos utilizando indicadores de envelhecimento.

Os indicadores globais do sistema, também conhecidos como *System Wide* (Capítulo 3) foram predominantes nos primeiros anos de pesquisa SAR. No entanto, nos últimos anos, a utilização de indicadores específicos de aplicação, conhecidos como *Application Specific* (Capítulo 3) ganha cada vez mais importância. Por exemplo, a memória física livre e espaço SWAP foram predominantes nos primeiros anos e, embora ainda estejam presentes nos estudos mais atuais, eles apareceram nos últimos anos em uma fração menor, enquanto outros indicadores começaram a aparecer, sendo eles específicos de aplicação.

Especificamente o indicador memória física livre (Mem) diminuiu de 50% (em 1997-1998) para 20% (em 2013-2014) considerando os estudos que utilizaram indicadores de envelhecimento. Esta diminuição é consequência da introdução de indicadores de envelhecimento novos e que podem ser considerados melhores. Desde 2007-2008 uma multidão de indicadores de envelhecimento tem sido experimentada. Estudos recentes (por exemplo, (MATIAS; FILHO, 2006), (ANDRZEJAK; SILVA, 2008)) demonstraram que o uso de indicadores específicos tem muito maior precisão do que os indicadores globais do sistema, para diferentes cenários de carga de trabalho.

Curiosamente, alguns indicadores de envelhecimento são citados, mas não são frequentemente utilizados nos estudos. Por exemplo, o tamanho da tabela de arquivo (FTsize) foi citado em estudos na amostra, mas foi utilizado em apenas um deles. As evidências sugerem oportunidades de pesquisa, pelo menos, em duas direções. A primeira direção é investigar novos e mais eficazes indicadores de envelhecimento, ou qualificar os existentes que têm sido pouco explorados até o presente momento. A segunda direção é combinar a variedade de indicadores de envelhecimento testados e de uma forma eficaz.

5.6 Técnicas de Rejuvenescimento

Conforme mostrado no Capítulo 3, sobre envelhecimento e rejuvenescimento de software (SAR), em termos de rejuvenescimento de software, existem técnicas que restauram o software, onde o mais comum é o reiniciar do software (sistema operacional ou aplicação específica). O ato de reiniciar uma aplicação ou um sistema operacional, normalmente, incorre em algum grau de paralisação de serviço, o que é um efeito colateral comum em típicas técnicas de rejuvenescimento. No entanto, desde que a arquitetura de sistemas de computação tem evoluído, o reinício foi aplicado a diferentes níveis de granularidade, como, por exemplo, o componente de aplicação (ALONSO et al., 2010), máquina virtual e *hypervisor*.

Assim, considerou-se a seguinte questão de pesquisa: **Quais as técnicas de rejuvenescimento têm sido usadas com mais frequência?** Foram encontrados 40 estudos com as seguintes técnicas de rejuvenescimento: reinicialização de Processo/App, reinicialização de Sistema Operacional (OS), reinicialização de máquina virtual (VM) e reinicialização de *hypervisor*. Considerando também o tipo de estudo, os estudos relacionados

com técnicas de rejuvenescimento foram bem equilibrados: 13 analíticos, 13 empíricos e 14 híbridos. A Tabela 9 resume essas descobertas. Foi considerado importante realizar a distribuição dos estudos ao longo dos anos. Essa distribuição pode ser observada na Figura 20.

Tabela 9 – Técnicas de Rejuvenescimento por Tipo de Estudo

Técnica de Rejuvenescimento	Analítico	Empírico	Híbrido
Reinicialização de Processo/App	8	7	6
Reinicialização de OS	2	2	2
Reinicialização de VM	2	3	4
Reinicialização de Hypervisor	1	1	2

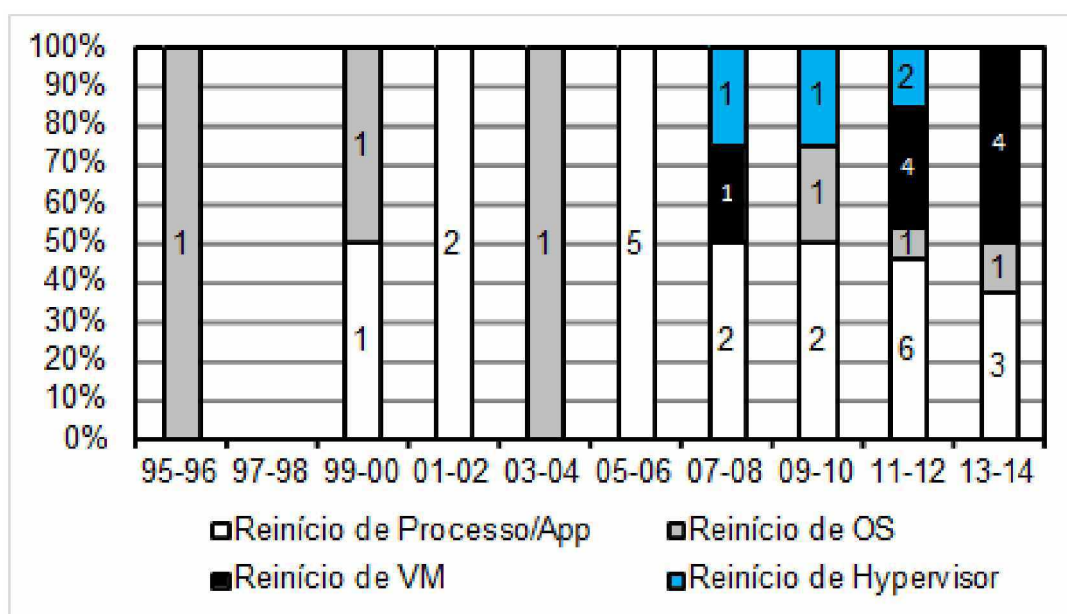


Figura 20 – Distribuição de estudos utilizando técnicas de rejuvenescimento.

É importante ressaltar que um único estudo pode ter usado mais de uma técnica de rejuvenescimento. Além disso, não estavam presentes na amostra os estudos sobre técnicas de rejuvenescimento nos anos de 1997 e 1998. As evidências sugerem que técnicas de rejuvenescimento com base em virtualização foram aplicadas de forma consistente a partir de 2008, onde o reinício de máquina de virtual teve grande destaque. A reinicialização do sistema operacional (OS) foi timidamente investigado nos últimos anos, o que representa uma oportunidade de investigação nesta área, especialmente na exploração de variantes desta abordagem como a reinicialização rápida de sistemas operacionais introduzido em (VAIDYANATHAN et al., 2001).

5.7 Estratégias de Rejuvenescimento

Conforme descrito no Capítulo 3, as técnicas de rejuvenescimento podem ser aplicadas para utilização de dois tipos de estratégia: a base em inspeção ou base no tempo. Com base no tempo é uma estratégia em que a técnica de rejuvenescimento é aplicada em intervalos de tempo regulares. Por outro lado, na estratégia à base em inspeção a técnica de rejuvenescimento depende da avaliação de indicadores de envelhecimento. A avaliação dos indicadores de envelhecimento é realizada de acordo com uma das seguintes abordagens: baseada em limiar, baseada em predição ou então, uma mistura das duas abordagens, nomeada aqui neste trabalho como mista. Os detalhes sobre essas abordagens foram apresentados no Capítulo 3.

Com base nessas alternativas, é perguntado: **Quais estratégias de rejuvenescimento foram mais prevalentes entre os estudos?** Foram analisados os 102 estudos de rejuvenescimento em relação à sua estratégia de rejuvenescimento, e em dois deles as descrições não eram claras o suficiente para classificá-los. Desconsiderando então esses dois estudos, 100 estudos da amostra sobre rejuvenescimento de software foram analisados. A Tabela 10 resume o resultado obtido, onde a coluna "Mista" corresponde ao número de estudos que aplicaram concomitantemente as abordagens baseadas em inspeção (limiar e predição).

Tabela 10 – Número de Estudos por Estratégia de Rejuvenescimento.

Baseado em Tempo	Baseado em Inspeção		
	Limiar	Predição	Mista
26	28	41	5

Com os resultados apresentados, nota-se que o número de estudos que usaram estratégias baseadas em tempo é consideravelmente menor do que o número de estudos que usaram estratégias baseadas em inspeção. A explicação para o número de estudos que usaram estratégias baseadas em inspeção é o fato de que elas investigam diversos fatores de um software para realização do rejuvenescimento, não considerando apenas o tempo, assim como as estratégias baseadas em tempo. Apesar de nos primeiros anos da introdução ao tema SAR as estratégias baseadas em tempo terem sido muito utilizadas, elas se baseiam apenas em um intervalo de tempo fixo para a realização do rejuvenescimento. Este rejuvenescimento pode acontecer em momentos não exatamente ideais, gerando custos e tempo de inatividades maiores do que o desejado. As estratégias mistas foram pouco utilizadas nos estudos da amostra. Esse resultado sugere uma oportunidade de investigação das vantagens e desvantagens na combinação de estratégias de rejuvenescimento.

5.8 Plataformas de Sistemas Operacionais

Na pesquisa realizada, foi considerado importante realizar a análise das plataformas de sistemas operacionais (OS) dos estudos da amostra. Contudo, diferentemente das análises anteriores, nesta pesquisa foram consideradas as plataformas de sistemas operacionais não como sujeito de pesquisa, mas sim como parte de instrumentação dos estudos. Ou seja, analisou-se quais foram os sistemas operacionais que os autores usaram para realizar as suas respectivas pesquisas.

Assim, a seguinte questão de pesquisa surgiu: **Quais foram as plataformas de sistemas operacionais mais prevalentes?** Encontrou-se 67 estudos que utilizaram as seguintes plataformas de sistema operacional: Linux/Unix (57 estudos), Windows (8 estudos), Android (1 estudo), e TinyOS (1 estudo). A distribuição de plataformas de OS, ao longo dos anos, é mostrado na Figura 21. A distribuição desses estudos de acordo com os seus tipos, ao longo dos anos, é mostrado na Figura 22. O número de estudos empíricos (33 estudos) foram prevalentes, seguidos dos estudos híbridos (20 estudos). Os estudos analíticos foram em menor número (14 estudos).

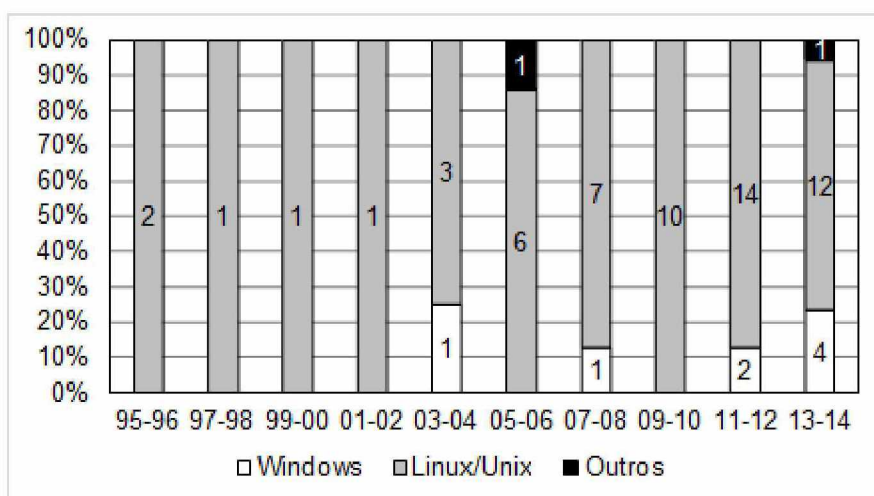


Figura 21 – Distribuição de estudos envolvendo plataformas de sistemas operacionais.

O envelhecimento de software se tornam mais evidentes em aplicações do lado do servidor, devido ao seu maior tempo em operação. Isso se torna ainda mais evidente quando, nos estudos da amostra, houve uma predominância de pesquisas em SAR no lado do servidor. Além disso, as plataformas de sistemas operacionais mais prevalentes foram Linux e Unix (utilizadas, preferencialmente, em sistemas de servidor).

É importante ressaltar que a nova geração de aplicações relacionadas à "Internet das Coisas" apresentam novos desafios, uma vez que é esperado que elas sejam executadas ubiquamente em regime 24/7, ou seja, 24 horas por dias nos 7 dias da semana. Estas aplicações relacionadas com internet estão fortemente baseadas em sistemas embarcados e móveis. Por isso, pode se considerar como oportunidades de pesquisa em SAR, realizar

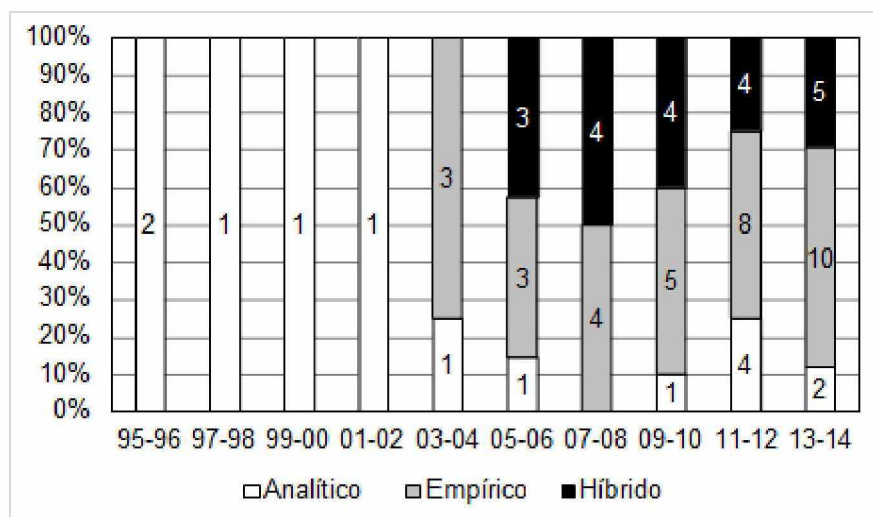


Figura 22 – Distribuição de estudos envolvendo plataformas de sistemas operacionais por tipo de estudo.

estudos considerando aplicações e sistemas operacionais para sistemas embarcados e dispositivos móveis ainda que, na amostra coletada, eles tenham sido em um número pouco considerável.

Conclusão

Este Capítulo apresenta as conclusões deste trabalho. As principais contribuições resumem os benefícios dos resultados obtidos. Os trabalhos futuros apresentam intenções de pesquisas futuras descobertas a partir desta pesquisa. As contribuições em produção bibliográfica mostram a publicação realizada a partir dos resultados obtidos.

6.1 Principais Contribuições

Este trabalho apresentou um mapeamento sistemático dos primeiros 20 anos de pesquisa na área de envelhecimento e rejuvenescimento de software (SAR). Esse mapeamento considerou 188 estudos (Apêndice A.1) do ano de 1995 até o ano de 2014. Com as respostas das questões de pesquisa, este trabalho revelou perspectivas promissoras.

Esta pesquisa contribui com pesquisadores inciantes ou que estão iniciando seus estudos na área de SAR, fornecendo uma visão geral do estado da arte, ajudando-os a encontrar temas de pesquisa em potencial. Contribuirá também com os pesquisadores que já são experientes na área de SAR, identificando lacunas e oportunidades pesquisa.

Os principais resultados mostraram um equilíbrio entre estudos focados em envelhecimento (45,74%) e rejuvenescimento (54,26%). No período de 2010 a 2014, o número de estudos duplicou em relação aos cinco anos anteriores. Além disso, o número de novos autores aumentou 83,87% no mesmo período. Isso mostra que a pesquisa na área de envelhecimento e rejuvenescimento de software e a sua comunidade vem crescendo significativamente nos últimos anos investigados.

Analisando os tipos de estudos, foram obtidas evidências que sugerem que os estudos híbridos estão se tornando abrangentes, com crescimento significativo nos últimos anos investigados. Contudo, os estudos analíticos e empíricos mostram que, mesmo presentes durante todos os períodos, eles ainda podem ser considerados tipos de estudos importantes para os próximos anos.

Em termos de efeitos de envelhecimento, a classe de vazamento de recursos é visivelmente a classe dominante nos estudos da amostra, onde a memória principal é o recurso

mais investigado durante todos os períodos. Uma oportunidade de pesquisa seria o desenvolvimento de algoritmos para reconhecimento de efeitos de envelhecimento. Sobre estudos que abordaram a classe de **fragmentação**, os de arquivos de banco de dados e os de sistema de arquivo, aparentam ser oportunidades de investigação, dado o intenso desenvolvimento de grandes aplicações de dados.

No caso dos indicadores de envelhecimento, é notável que os indicadores de aplicações específicas (*Application Specific*) estão ganhando espaço em relação aos indicadores globais de sistema (*System Wide*). Indicadores tais como *Resident Set Size*, *Heap Utilizada*, *Tamanho da Memória Virtual* e *JVM Heap Size*, aparecem como uma tendência nos estudos. Enquanto indicadores como *Memória Física Livre* e *Espaço SWAP*, continuam sendo utilizados nos estudos, contudo, em uma fração menor. As evidências sugerem oportunidades de pesquisa como investigar novos e mais eficazes indicadores de envelhecimento, ou qualificar os existentes que têm sido pouco explorados até o presente momento. Além disso, combinar com eficácia a variedade de indicadores de envelhecimento testados.

Em relação as técnicas de rejuvenescimento, a reinicialização da Máquina Virtual (VM) teve o maior aumento nos últimos anos. Surpreendentemente, há muito a ser feito a nível de sistema operacional e de *hypervisor*, que foram timidamente investigados nos estudos da amostra. Avanços nestes níveis beneficiariam sistemas baseados em máquinas virtuais e, conseqüentemente, a computação em nuvem que é um conceito em ascensão e está relacionado à virtualização.

Dada a tendência dos estudos híbridos, o rejuvenescimento à base de inspeção mista foi uma estratégia pouco utilizada nos estudos da amostra. Esse resultado sugere uma oportunidade de investigação sobre as vantagens e desvantagens na combinação de estratégias de rejuvenescimento.

Investigou-se também as plataformas de sistemas operacionais utilizadas por pesquisadores da área. No caso, as plataformas consideradas são aquelas que serviram como instrumento de pesquisa para os autores, para que eles pudessem realizar os seus experimentos. Não foram consideradas as plataformas que foram o objeto de pesquisa. Os resultados mostraram que, claramente, as plataformas de sistemas operacionais mais prevalentes foram Linux e Unix. Esses sistemas operacionais são utilizados, preferencialmente, em sistemas de servidor, o que sugere uma predominância de pesquisas em SAR no lado do servidor. Além disso, a nova geração de aplicações está fortemente relacionada com a "Internet das Coisas" e muitas vezes são aplicações baseadas em sistemas embarcados e móveis. Com isso, foi considerada também como oportunidade de pesquisa em SAR, realizar estudos considerando aplicações e sistemas operacionais para sistemas embarcados e dispositivos móveis.

A plataforma de sistema operacional Windows, apesar de ser conhecida e utilizada mundialmente, mesmo por usuários comuns, não foi uma plataforma prevalente nos estudos da amostra. Uma oportunidade de pesquisa seria aprofundar os estudos sobre essa

plataforma, utilizando-a como ferramenta de pesquisa nos estudos realizados, assim como tem sido realizado com as plataformas Unix e Linux.

6.2 Ameaças à Validade

Este trabalho foi realizado com base em uma amostra de 188 estudos que foram coletados seguindo rigorosos critérios de seleção definidos em um protocolo. Neste protocolo, especialistas da área de SAR foram consultados para que sugerissem veículos de publicação para a coleta desses estudos. Dessa forma, foram acatadas apenas as sugestões desses especialistas, buscando os estudos iniciais da amostra apenas nos veículos sugeridos. Assim, tem-se a ciência de que bons veículos de publicação, com renome internacional, não foram incluídos na lista de veículos deste trabalho. Como forma de mitigação, foi realizado o procedimento de *Snowballing*. Esse procedimento permitiu adicionar à amostra outros estudos de relevância (grande número de citações nos estudos da amostra inicial) e que respeitassem os critérios de seleção definidos.

6.3 Trabalhos Futuros

Os resultados obtidos nesta pesquisa e que foram apresentados no Capítulo 5 deixaram motivação para novas investigações em trabalhos futuros. Conforme dito no Capítulo 2, um mapeamento sistemático é projetado com intuito de fornecer uma ampla visão de uma área de pesquisa. Com isso, há a intenção de realizar investigações a partir das oportunidades de pesquisa que este trabalho encontrou e que foram citadas na Seção 6.1.

6.4 Contribuições em Produção Bibliográfica

Com o desenvolvimento deste trabalho, um artigo com os resultados da pesquisa realizada foi publicado em 2016 no WoSAR (*Workshop On Software Aging And Rejuvenation*), o principal evento da área de envelhecimento e rejuvenescimento de software.

Referências

- ALONSO, J.; GAVALDA, R.; BERRAL, J. L.; TORRES, J. Adaptive on-line software aging prediction based on machine learning. **IEEE/IFIP International Conference on Dependable Systems Networks (DSN)**, IEEE Computer Society, p. 507–516, 2010.
- ALONSO, J.; MATIAS, R.; VICENTE, E.; MARIA, A. A comparative experimental study of software rejuvenation overhead. **Performance Evaluation**, Elsevier Science Publishers B. V., Amsterdam, The Netherlands, The Netherlands, v. 70, n. 3, p. 231–250, mar. 2013. ISSN 0166-5316.
- ANDRZEJAK, A.; SILVA, L. Using machine learning for non-intrusive modeling and prediction of software aging. In: IEEE. **NOMS 2008-2008 IEEE Network Operations and Management Symposium**. [S.l.], 2008. p. 25–32.
- ARAÚJO, J.; MATOS, R.; MACIEL, P.; VIEIRA, F.; MATIAS, R.; TRIVEDI, K. S. Software rejuvenation in eucalyptus cloud computing infrastructure: A method based on time series forecasting and multiple thresholds. In: IEEE. **Software Aging and Rejuvenation (WoSAR), 2011 IEEE Third International Workshop on**. [S.l.], 2011. p. 38–43.
- BARADA, S.; SWAIN, S. K. A survey report on software aging and rejuvenation studies in virtualized environment. **Int J Comput Eng Technol (IJCSET)**, v. 5, n. 5, p. 541–546, 2014.
- BOVENZI, A.; COTRONEO, D.; PIETRANTUONO, R.; RUSSO, S. On the aging effects due to concurrency bugs: a case study on mysql. In: IEEE. **2012 IEEE 23rd International Symposium on Software Reliability Engineering**. [S.l.], 2012. p. 211–220.
- BRERETON, P.; KITCHENHAM, B.; BUDGEN, D.; TURNER, M.; KHALIL, M. Lessons from applying the systematic literature review process within the software engineering domain. **Journal of systems and software**, Elsevier, v. 80, n. 4, p. 571–583, 2007.
- COTRONEO, D.; NATELLA, R.; PIETRANTUONO, R.; RUSSO, S. Software aging analysis of the linux operating system. In: IEEE. **2010 IEEE 21st International Symposium on Software Reliability Engineering**. [S.l.], 2010. p. 71–80.

- COTRONEO, D.; NATELLA, R.; PIETRANTUONO, R.; RUSSO, S. A survey of software aging and rejuvenation studies. **J. Emerg. Technol. Comput. Syst.**, ACM, New York, NY, USA, v. 10, n. 1, p. 8:1–8:34, jan. 2014. ISSN 1550-4832.
- FREITAS, H.; OLIVEIRA, M.; SACCOL, A. Z.; MOSCAROLA, J. O método de pesquisa survey. **Revista de administração**, v. 35, n. 3, p. 105–112, 2000.
- GARG, S.; MOORSEL, A. V.; VAIDYANATHAN, K.; TRIVEDI, K. S. A methodology for detection and estimation of software aging. In: IEEE. **Software Reliability Engineering, 1998. Proceedings. The Ninth International Symposium on**. [S.l.], 1998. p. 283–292.
- GROTTKE, M.; LI, L.; VAIDYANATHAN, K.; TRIVEDI, K. S. Analysis of software aging in a web server. **IEEE Transactions on reliability**, IEEE, v. 55, n. 3, p. 411–420, 2006.
- GROTTKE, M.; MATIAS, R.; TRIVEDI, K. S. The fundamentals of software aging. In: **IEEE Proceedings of Workshop on Software Aging and Rejuvenation, in conjunction with ISSRE**. Seattle, WA. [S.l.: s.n.], 2008.
- HUANG, Y.; KINTALA, C.; KOLETTIS, N.; FULTON, N. D. Software rejuvenation: Analysis, module and applications. In: IEEE. **Fault-Tolerant Computing, 1995. FTCS-25. Digest of Papers., Twenty-Fifth International Symposium on**. [S.l.], 1995. p. 381–390.
- JIA, Y.-F.; SU, J.-Y.; CAI, K.-Y. A feedback control approach for software rejuvenation in a web server. In: IEEE. **Software Reliability Engineering Workshops, 2008. ISSRE Wksp 2008. IEEE International Conference on**. [S.l.], 2008. p. 1–6.
- KEELE, S. Guidelines for performing systematic literature reviews in software engineering. In: **Technical report, Ver. 2.3 EBSE Technical Report. EBSE**. [S.l.: s.n.], 2007.
- KINTALA, C. M. Software rejuvenation in embedded systems. **Journal of Automata, Languages and Combinatorics**, Otto-von-Guericke-Universitat, v. 14, n. 1, p. 63–73, 2009.
- KITCHENHAM, B. Procedures for performing systematic reviews. **Keele, UK, Keele University**, v. 33, n. 2004, p. 1–26, 2004.
- KITCHENHAM, B.; CHARTERS, S. Procedures for performing systematic literature reviews in software engineering. **Keele University & Durham University, UK**, 2007.
- KITCHENHAM, B.; DYBA, T.; JORGENSEN, M. Evidence-based software engineering. In: IEEE COMPUTER SOCIETY. **Proceedings of the 26th international conference on software engineering**. [S.l.], 2004. p. 273–281.
- KITCHENHAM, B.; PFLEEGER, S. L.; PICKARD, L. M.; JONES, P. W.; HOAGLIN, D. C.; EMAM, K. E.; ROSENBERG, J. Preliminary guidelines for empirical research in software engineering. **IEEE Transactions on software engineering**, IEEE, v. 28, n. 8, p. 721–734, 2002.

- KUO, S.-Y.; HUANG, C.-Y.; LYU, M. R. Framework for modeling software reliability, using various testing-efforts and fault-detection rates. **IEEE Transactions on Reliability**, IEEE, v. 50, n. 3, p. 310–320, 2001.
- LIMA, M. M. de; LIMA, A. R. de; MONTEIRO, A. C. C.; JÚNIOR, E. H. C.; GOMES, L. d. Q. L. Uma revisão sistemática da literatura dos processos de desenvolvimento de software educativo. In: **Anais do Simpósio Brasileiro de Informática na Educação**. [S.l.: s.n.], 2012. v. 23, n. 1.
- MALCHER, P. R. C.; OLIVEIRA, S. R. B. Um mapeamento sistemático sobre abordagens de apoio à rastreabilidade de requisitos no contexto de projetos de software.
- MATIAS, R.; FILHO, J. P. An experimental study on software aging and rejuvenation in web servers. In: **IEEE. 30th Annual International Computer Software and Applications Conference (COMPSAC'06)**. [S.l.], 2006. v. 1, p. 189–196.
- MATOS, R.; ARAUJO, J.; MACIEL, P.; SOUZA, F. V. de; MATIAS, R.; TRIVEDI, K. Software rejuvenation in eucalyptus cloud computing infrastructure: a hybrid method based on multiple thresholds and time series prediction. **Int. Trans. Syst. Sci. Appl.**, v. 7, p. 295–303, 2011.
- MIAN, P.; CONTE, T.; NATALI, A.; BIOLCHINI, J.; TRAVASSOS, G. A systematic review process for software engineering. In: **ESELAW'05: 2nd Experimental Software Engineering Latin American Workshop**. [S.l.: s.n.], 2005.
- PEK, E.; LÄMMEL, R. A literature survey on empirical evidence in software engineering. **CoRR**, abs/1304.1002, 2013.
- PINSONNEAULT, A.; KRAEMER, K. Survey research methodology in management information systems: an assessment. **Journal of management information systems**, Taylor & Francis, v. 10, n. 2, p. 75–105, 1993.
- RAMOS, E.; BRASIL, M. Um mapeamento sistemático sobre padrões de software para reengenharia de sistemas. **80f. Monografia (Especialização em Engenharia de Software com Ênfase em Padrões de Software)–Universidade Estadual do Ceará, Fortaleza-CE**, 2011.
- ROTHER, E. T. Systematic literature review x narrative review. **Acta Paulista de Enfermagem**, SciELO Brasil, v. 20, n. 2, p. v–vi, 2007.
- VAIDYANATHAN, K.; HARPER, R. E.; HUNTER, S. W.; TRIVEDI, K. S. Analysis and implementation of software rejuvenation in cluster systems. In: **ACM. ACM SIGMETRICS Performance Evaluation Review**. [S.l.], 2001. v. 29, n. 1, p. 62–71.
- WOHLIN, C. Guidelines for snowballing in systematic literature studies and a replication in software engineering. In: **ACM. Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering**. [S.l.], 2014. p. 38.
- XAVIER, J.; MACÊDO, A.; MATIAS, R.; BORGES, L. Engenharia de confiabilidade de software: Um mapeamento sistemático da última década. **IV Simpósio Brasileiro de Engenharia de Sistemas Computacionais (SBESC)**, 2014.

ZHAO, J.; TRIVEDI, K. S. Performance modeling of apache web server affected by aging. In: IEEE. **Software Aging and Rejuvenation (WoSAR), 2011 IEEE Third International Workshop on**. [S.l.], 2011. p. 56–61.

Apêndices

APÊNDICE **A**

Lista de Estudos

A amostra utilizada neste trabalho foi composta pelos estudos na área de envelhecimento e rejuvenescimento de software a seguir.

A.1 Lista de Estudos da Amostra

Título do estudo	DOI
A comparative evaluation of software rejuvenation strategies.	10.1109/Wosar.2011.16
A comparative experimental study of software rejuvenation overhead.	10.1016/J.Peva.2012.09.002
A comprehensive approach to optimal software rejuvenation.	10.1016/J.Peva.2013.05.010
A comprehensive model for software rejuvenation.	10.1109/Tdsc.2005.15
A fast rejuvenation technique for server consolidation with virtual machines.	10.1109/Dsn.2007.6
A feedback control approach for software rejuvenation in a web server.	10.1109/Issrew.2008.5355514
A framework of survivability model for wireless sensor network.	10.1109/Ares.2006.6
A measurement-based aging analysis of the java virtual machine.	10.1002/Stvr.467
A measurement-based model for estimation of resource exhaustion in operational software systems.	10.1109/Issre.1999.809313
A methodology for detection and estimation of software aging.	10.1109/Issre.1998.730892
A neural-wavelet based methodology for software aging forecasting.	10.1109/Icsmc.2005.1571122
A nonlinear approach to modeling of software aging in a web server.	10.1109/Apsec.2007.38
A performability-oriented software rejuvenation framework for distributed applications.	10.1109/Dsn.2005.12
A pomdp formulation of multistep failure model with software rejuvenation.	10.1109/Wosar.2011.11

A proactive fault-detection mechanism in large-scale cluster systems.	10.1109/Ipdp.1639332
A queuing model for service availability of systems with rejuvenation.	10.1109/Issrew.2008.5355520
A rejuvenation methodology of cluster recovery.	10.1109/Ccgrid.2005.1558539
A rough wavelet network model with genetic algorithm and its application to aging forecasting of application server.	10.1109/Icmlc.4370668
A self-stabilizing autonomic recoverer for eventual byzantine software.	10.1016/J.Jss.028
A simulation study on the effectiveness of restart and rejuvenation to mitigate the effects of software ageing.	10.1109/Wosar.2010.5722100
A systematic differential analysis for fast and robust detection of software aging.	10.1109/Srds.2014.38
A threshold autoregressive model for software aging.	10.1109/Sose.2006.5
A workload-based analysis of software aging, and rejuvenation.	10.1109/Tr.2005.853442
Accelerated degradation tests applied to software aging experiments.	10.1109/Tr.2034292
Achieving fault-tolerant software with rejuvenation and reconfiguration.	10.1109/Ms.2001.936217
Adaptation mechanisms for survivable sensor networks against denial of service attack.	10.1109/Ares.2007.32
Adaptive on-line software aging prediction based on machine learning.	10.1109/Dsn.2010.5544275
Adaptive software rejuvenation : degradation model and rejuvenation scheme.	10.1109/Dsn.2003.1209934
Advanced pattern recognition for detection of complex software aging phenomena in online transaction processing servers.	10.1109/Dsn.2002.1028933
Aging-related bugs in cloud computing software.	10.1109/Issrew.2012.97
An advanced methodology for measuring and characterizing software aging.	10.1109/Issrew.2012.81
An approach for estimation of software aging in a web server.	10.1109/Isese.2002.1166929
An experiment in memory leak analysis with a mission-critical middleware for air traffic control.	10.1109/Issrew.2008.5355513
An experimental study on software aging and rejuvenation in web servers.	10.1109/Compsac.2006.25
An overview of existing tools for fault-injection and dependability benchmarking in grids.	10.1.1.108.6033
Analysis and application of conditional software rejuvenation — a new approach.	10.1109/Issrew.2008.5355522
Analysis of a two-level software rejuvenation policy.	10.1016/J.Ress.2004.02.011
Analysis of inspection-based preventive maintenance in operational software systems.	10.1109/Reldis.2002.1180199

Analysis of preventive maintenance in transactions based software systems.	10.1109/12.656092
Analysis of restart mechanisms in software systems.	10.1109/Tse.2006.73
Analysis of service availability for time-triggered rejuvenation policies.	10.1016/J.Jss.2010.05.022
Analysis of software aging in a web server.	10.1109/Tr.2006.879609
Analysis of software rejuvenation using markov regenerative stochastic petri net.	10.1109/Issre.1996.497656
Analysis of software rejuvenation via a multi-agent approach.	10.1016/J.Jss.2015.02.017
Application server aging prediction model based on wavelet network with adaptive particle swarm optimization algorithm.	10.1007/978-3-540-74205-0_3
Applying partial and full rejuvenation in different degradation levels.	10.1109/Wosar.2011.14
Availability analysis and improvement of active/standby cluster systems using software rejuvenation.	10.1016/S0164-1212(01)00107-8
Availability analysis of application servers using software rejuvenation and virtualization.	10.1007/S11390-009-9228-1
Availability and reliability estimation for a system undergoing minimal, perfect and failed rejuvenation.	10.1109/Issrew.2008.5355519
Availability modeling and analysis on virtualized clustering with rejuvenation.	10.1.1.583.2945
Availability optimization in operational software system with aperiodic time-based software rejuvenation scheme.	10.1109/Issrew.2008.5355516
Availability study on cloud computing environments : live migration as a rejuvenation mechanism.	10.1109/Dsn.2013.6575322
Can linux be rejuvenated without reboots?.	10.1109/Wosar.2011.12
Chaotic analysis of software aging in web server.	10.1109/Sose.2006.10
Characterization of software aging effects in elastic storage mechanisms for private clouds.	10.1109/Issrew.2012.82
Characterizing aging phenomena of the java virtual machine.	10.1109/Srds.2007.22
Combined server rejuvenation in a virtualized data center.	10.1109/Uic-Atc.52
Comparing four case studies on bohr-mandel characteristics using odc.	10.1109/Issrew.2013.6688908
Comparison of expected failure times for several replacement policies.	10.1109/Tr.2006.879602
Compositional fluid stochastic petri net model for operational software system performance.	10.1109/Issrew.2008.5355517
Comprehensive evaluation of aperiodic checkpointing and rejuvenation schemes in operational software system.	10.1016/J.Jss.06.058
Cost optimality in testing and rejuvenation.	10.1109/Issrew.2012.84
Critical-reliability path identification and delay analysis.	10.1145/2564926

Decentralized deduplication in san cluster file systems.	10.1.1.149.292
Dependability analysis of a client/server software system with rejuvenation.	10.1109/Issre.2002.1173241
Detecting software aging of web servers with real-valued negative.	10.1016/J.Peva.2013.10.001
Detection and analysis of performance deterioration in mobile offloading system.	10.1109/Issrew.2014.15
Detection and prediction of resource-exhaustion vulnerabilities.	10.1109/Issre.2008.47
Detection and root cause analysis of memory-related software aging defects by automated tests.	10.1109/Mascots.2013.53
Deterministic models of software aging and optimal rejuvenation schedules.	10.1109/Inm.374780
Discouraging software piracy using software aging	10.1007/3-540-47870-1_1
Discrete availability models to rejuvenate a telecommunication billing application.	10.1109/Hase.2002.1173117
Dynamic software rejuvenation policies in a transaction-based system under markovian arrival processes.	10.1016/J.Peva.2012.07.004
Ensuring stable performance for systems that degrade.	10.1145/1071021.1071026
Ensuring system performance for cluster and single server systems.	10.1016/J.Jss.2006.07.020
Ensuring the performance of apache http server affected by aging.	10.1109/Tdsc.2013.38
Entropy-based detection of incipient faults in software systems.	10.1109/Prdc.2012.14
Estimating discrete-time periodic software rejuvenation.	10.1142/S0218539306002446
Estimating software rejuvenation schedules in high-assurance systems.	10.1093/Comjnl/44.6.473
Evaluation of software performance affected by aging.	10.1109/Wosar.2010.5722093
Fast software rejuvenation of virtual machine monitors.	10.1109/Tdsc.2010.20
Fault-tolerant execution of mobile agents.	10.1109/Icdsn.857527
Fine grained software degradation models for optimal rejuvenation policies.	10.1016/S0166-5316(01)00037-2
High availability solution: resource usage management in virtualized software aging	10.1145/2078861.2078864
High-available grid services through the use of virtualized clustering.	10.1109/Grid.2007.4354113
How does testing affect the availability of aging software systems?.	10.1016/J.Peva.2012.05.007
Identifying silent failures of saas services using finite state machine based invariant analysis.	10.1109/Issrew.2013.6688909
Improving dependability using shared supplementary memory and opportunistic micro rejuvenation in multi-tasking embedded systems	10.1109/Prdc.2007.18
Improving fault tolerance by virtualization and software rejuvenation.	10.1109/Ams.2008.75

Injecting memory leaks to accelerate software failures.	10.1109/Issre.2011.24
Is software aging related to software metrics?.	10.1109/Wosar.2010.5722096
Job completion time on a virtualized server subject to software aging and rejuvenation.	10.1109/Wosar.2011.10
Job completion time on a virtualized server with software rejuvenation.	10.1145/2539121
Lifetime reliability for load-sharing redundant systems with arbitrary failure distributions.	10.1109/Tr.2010.2048679
Machine learning techniques for predicting web server anomalies.	10.1109/Ncca.2011.25
Managing performance of aging applications via synchronized replica rejuvenation.	10.1007/978-3-540-75694-1_9
Maximizing interval reliability in operational software system with rejuvenation.	10.1109/Issre.2003.1251068
Measuring software aging effects through os kernel instrumentation.	10.1109/Wosar.2010.5722094
Memory leak analysis of mission-critical middleware.	10.1016/J.Jss.2010.05.027
Memory leak detection in sun solaris os.	10.1109/Iscsct.2008.183
Methods and opportunities for rejuvenation in aging distributed software systems.	10.1016/J.Jss.2010.05.026
Minimizing completion time of a program by checkpointing and rejuvenation.	10.1145/233008.233050
Mining software aging patterns by artificial neural networks.	10.1007/978-3-540-69939-2_24
Mining software aging: a neural network approach	10.1109/ISCC.4625660
Model based approach for autonomic availability management.	10.1007/11955498_1
Modeling and analysis of software aging and rejuvenation.	10.1109/Simsym.2000.844925
Modeling and analysis of software aging and software failure.	10.1016/J.Jss.2006.06.029
Modeling and analysis of software rejuvenation in a server virtualized system with live vm migration.	10.1016/J.Peva.2012.09.003
Modeling and analysis of software rejuvenation in a server virtualized system.	10.1109/Wosar.2010.5722098
Modeling and analysis of software rejuvenation in cable modem termination systems.	10.1109/Issre.2002.1173239
Modeling and analyzing server system with rejuvenation through sysml and stochastic reward nets.	10.1109/Ares.2011.28
Modeling and performance analysis of software rejuvenation policies for multiple degradation systems.	10.1109/Compsac.2009.39
Modeling perfect and minimal rejuvenation for client server systems with heterogeneous load.	10.1109/Prdc.2008.22
Modeling software rejuvenation on a redundant system using monte carlo simulation.	10.1109/Issrew.2012.89
Modeling software systems with rejuvenation, restoration and checkpointing through fluid stochastic petri nets.	10.1109/Pnpm.796555

Monitoring memory-related software aging: an exploratory study.	10.1109/Issrew.2012.90
Monitoring of aging software systems affected by integer overflows.	10.1109/Issrew.2012.91
Multiformalism to support software rejuvenation modeling.	10.1109/Issrew.2012.92
Multi-granularity software rejuvenation policy based on continuous time markov chain.	10.1109/Wosar.2011.9
Multi-scale entropy: one metric of software aging	10.3850/ 978-981-07-8859-9_10
Non-parametric predictive inference of adaptive software rejuvenation schedule.	10.1109/Issrew.2008.5355521
Non-parametric predictive inference of preventive rejuvenation schedule in operational software systems.	10.1109/Issre.2007.13
On the aging effects due to concurrency bugs: a case study on mysql.	10.1109/Issre.2012.50
On the effectiveness of mann-kendall test for detection of software aging.	10.1109/Issrew.2013.6688905
On the potential of software rejuvenation for long-running sensor network deployments.	10.1145/1809111.1809124
Online anomaly prediction for robust cluster systems.	10.1109/Icde.2010.128
Optimal resource allocation in a virtualized software aging platform with software rejuvenation.	10.1109/Issre.2011.30
Optimal software rejuvenation for tolerating soft failures	10.1109/32.553701
Optimal software rejuvenation policy with discounting.	10.1109/Prdc.2001.992684
Optimization of opportunity-based software rejuvenation policy.	10.1109/Issrew.2012.93
Optimizing software rejuvenation policies under interval reliability criteria.	10.1109/UIC-ATC.111
Performability analysis of clustered systems with rejuvenation under varying workload.	10.1016/J.Peva.2006.04.002
Performance assurance via software rejuvenation: monitoring, statistics and algorithms.	10.1109/Dsn.2006.58
Performance degradation analysis of a supercomputer.	10.1109/Issrew.2013.6688904
Performance modeling of apache web server affected by aging.	10.1109/Wosar.2011.13
Performance-aware software rejuvenation strategies in a queueing system.	10.1109/Wosar.2010.5722099
Predicting aging-related bugs using software complexity metrics.	10.1016/J.Peva.2012.09.004
Predicting software anomalies using machine learning techniques.	10.1109/Nca.2011.29
Predicting web server crashes: a case study in comparing prediction algorithms.	10.1109/Icas.2008.56
Prediction of performance anomalies in web-applications based-on software aging scenarios.	10.1109/Wosar.2010.5722095

Preventing secret leakage from fork(): securing privilege-separated applications.	10.1109/Icc.255108
Proactive management of software aging.	10.1147/Rd.452.0311
Proactive software rejuvenation based on machine learning techniques.	10.1007/978-3-642-12636-9_13
Rejuvenating high available virtualized systems.	10.1109/Ares.2010.69
Rejuvenation and failure detection in partitionable systems.	10.1109/Prdc.2001.992692
Rejuvenation and the spread of epidemics in general topologies.	10.1109/Issrew.2014.61
Robust and adaptive modeling of software aging.	10.1.1.71.196
Semi-markov availability modeling of a redundant system with partial and full rejuvenation actions.	10.1109/Depcos-Relcomex.2008.13
Software aging analysis of the linux operating system.	10.1109/Issre.2010.24
Software aging analysis of web server using neural networks.	10.5121/Ijaia.3302
Software aging and multifractality of memory resources.	10.1109/Dsn.2003.1209987
Software aging and rejuvenation in a soap-based server.	10.1109/Nca.2006.51
Software aging and rejuvenation: where we are and where we are going.	10.1109/Wosar.2011.15
Software aging estimation and prediction of a real vod system based on pca and neural networks.	10.1109/Icinf.5204903
Software aging forecasting model of service-oriented application server based on wavelet network with adaptive genetic algorithm	10.1155/2007/57086
Software aging in the eucalyptus cloud computing infrastructure : characterization and rejuvenation.	10.1145/2539122
Software aging prediction model based on fuzzy wavelet network with adaptive genetic algorithm.	10.1109/Ictai.2005.104
Software aging trend analysis of server virtualized system.	10.1109/Icoin.2014.6799478
Software life-extension : a new countermeasure to software aging.	10.1109/Issre.2012.19
Software maintenance optimization based on stackelberg game methods.	10.1109/Issrew.2014.38
Software rejuvenation and reconfiguration for enhancing survivability of sensor networks.	10.1109/Waina.2007.231
Software rejuvenation impacts on a phased-mission system for mars exploration.	10.1109/Issrew.2013.6688906
Software rejuvenation in eucalyptus cloud computing infrastructure: a method based on time series forecasting and multiple thresholds.	10.1109/Wosar.2011.18
Software rejuvenation on a pki.	10.1109/Wosar.2010.5722102
Software rejuvenation policies for cluster systems under varying workload.	10.1109/Prdc.2004.1276563

Software rejuvenation scheduling using accelerated life testing.	10.1145/2539118
Software rejuvenation strategy based on components.	10.1109/Wcse.2010.113
Software rejuvenation: analysis, module and applications.	10.1109/Ftcs.1995.466961
Software rejuvenation: do it & industries use it?.	10.1109/Issrew.2012.96
Statistical non-parametric algorithms to estimate the optimal software rejuvenation schedule.	10.1109/Prdc.2000.897287
The fundamentals of software aging.	10.1109/Issrew.2008.5355512
The mechanics of memory-related software aging.	10.1109/Wosar.2010.5722097
The nature of the times to flight software failure during space missions.	10.1109/Issre.2012.32
The relationship research between usage of resource and performance of computer system.	10.1109/Wcse.2.169
Time and prediction based software rejuvenation policy.	10.1109/Itcs.2010.34
Toward optimal virtual machine placement and rejuvenation scheduling in a virtualized data center.	10.1109/Issrew.2008.5355515
Towards a practical survivable intrusion tolerant replication system.	10.1109/Srds.2014.16
Towards fast os rejuvenation: an experimental evaluation of fast os reboot techniques.	10.1109/Issre.2013.6698905
Towards patching memory leak bugs in off-the-shelf software.	10.1109/Issrew.2014.44
Towards performability modeling of software rejuvenation.	10.1.1.141.5327
Transient analysis of software rejuvenation policies in virtualized system: Phase-type expansion approach.	10.1080/16843703.2014.11673349
Understanding bohr-mandel bugs through odc triggers and a case study with empirical estimations of their field proportion.	10.1109/Wosar.2011.17
Using accelerated life tests to estimate time to software aging failure.	10.1109/Issre.2010.42
Using machine learning for non-intrusive modeling and prediction of software aging.	10.1109/Noms.4575113
Using performance signatures and software rejuvenation for worm mitigation in tactical manets.	10.1145/1216993.1217023
Using virtualization to improve software rejuvenation.	10.1109/Tc.2010.119
Workload characterization for software aging analysis.	10.1109/Issre.2011.18
Workload-aware anomaly detection for web applications.	10.1016/J.Jss.2013.03.060