

# RESSALVA

Atendendo solicitação do autor,  
o texto completo desta dissertação  
será disponibilizado somente a partir  
de 27/03/2026.



UNIVERSIDADE ESTADUAL PAULISTA "JÚLIO  
DE MESQUITA FILHO"  
INSTITUTO DE GEOCIÊNCIAS E CIÊNCIAS  
EXATAS – RIO CLARO



---

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA  
COMPUTAÇÃO

**DYNASCHEMA: UMA BIBLIOTECA PARA EVOLUÇÃO DE BANCO  
DE DADOS RELACIONAL PARA O DOMÍNIO DE SOFTWARE  
AUTOADAPTATIVO**

**GABRIEL NAGASSAKI CAMPOS**

**RIO CLARO-SP  
2024**

UNIVERSIDADE ESTADUAL PAULISTA

“Júlio de Mesquita Filho”

Instituto de Geociências e Ciências Exatas

Câmpus de Rio Claro

GABRIEL NAGASSAKI CAMPOS

**DynaSchema: Uma Biblioteca para Evolução de Banco de Dados  
Relacional para o Domínio de Software Autoadaptativo**

Dissertação de Mestrado apresentada ao Instituto de Geociências e Ciências Exatas do Câmpus de Rio Claro, da Universidade Estadual Paulista “Júlio de Mesquita Filho”, como parte dos requisitos para obtenção do título de Mestre em Ciência da Computação.

Orientador: Prof. Dr. Frank José Affonso

Rio Claro – SP

2024

C198d Campos, Gabriel Nagassaki  
DynaSchema : uma biblioteca para evolução de banco de dados relacional para o domínio de software autoadaptativo / Gabriel Nagassaki Campos. -- Rio Claro, 2024  
152 f.  
  
Dissertação (mestrado) - Universidade Estadual Paulista (Unesp), Instituto de Geociências e Ciências Exatas, Rio Claro  
Orientador: Frank José Affonso  
  
1. Evolução do esquema de dados. 2. Software autoadaptativo. 3. Biblioteca. I. Título.

Sistema de geração automática de fichas catalográficas da Unesp. Biblioteca do Instituto de Geociências e Ciências Exatas, Rio Claro. Dados fornecidos pelo autor(a).

Essa ficha não pode ser modificada.

UNIVERSIDADE ESTADUAL PAULISTA  
“Júlio de Mesquita Filho”  
Instituto de Geociências e Ciências Exatas  
Câmpus de Rio Claro

GABRIEL NAGASSAKI CAMPOS

**DynaSchema: Uma Biblioteca para Evolução de Banco de Dados  
Relacional para o Domínio de Software Autoadaptativo**

Dissertação de Mestrado apresentada ao Instituto de Geociências e Ciências Exatas do Câmpus de Rio Claro, da Universidade Estadual Paulista “Júlio de Mesquita Filho”, como parte dos requisitos para obtenção do título de Mestre em Ciência da Computação.

**Comissão Examinadora**

- Prof. Dr. Frank José Affonso (Orientador)  
Departamento de Estatística, Matemática Aplicada e Computação (DEMAC)  
Universidade Estadual Paulista “Júlio De Mesquita Filho” (UNESP) – Câmpus de Rio Claro – SP
- Profa. Dra. Marcela Xavier Ribeiro  
Departamento de Computação  
Universidade Federal de São Carlos – SP
- Prof. Dr. Rogério Eduardo Garcia  
Departamento de Matemática e Computação  
Universidade Estadual Paulista “Júlio De Mesquita Filho” (UNESP) – Câmpus de Presidente Prudente – SP

**Conceito: Aprovado.**

Rio Claro (SP), 27 de março de 2024.

# AGRADECIMENTOS

Agradeço a Deus por todas as graças conquistadas ao longo da vida, que inclui esta jornada de mestrado acadêmico. Também agradeço à intercessão de Nossa Senhora, Santa Mãe de Deus, que atende de maneira misericordiosa a todas as nossas suplicas.

Agradeço a minha família por encorajar minha carreira acadêmica, provendo todo o suporte necessário para minha dedicação exclusiva aos estudos.

Agradeço ao Prof. Dr. Frank José Afonso, pela parceria, paciência e sabedoria em orientar minha formação acadêmica, desde a iniciação científica, graduação, e agora, a pós-graduação.

Agradeço aos membros da banca examinadora pela avaliação cuidadosa e pelos comentários valiosos que contribuíram com este projeto de mestrado acadêmico.

Agradeço a todos os colaboradores da Universidade Estadual Paulista “Júlio de Mesquita Filho” que colaboraram direta ou indiretamente com minha formação acadêmica, desde a graduação até a pós-graduação.

Agradeço pelo apoio financeiro da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES N. 88887.670715/2022-00) na pós-graduação, além do Projeto LINEAS – Lineamentos e Zonas de Fraturas Continentais: Possíveis Corredores de Deformação da Bacia de Santos e seu Embasamento (FUNDUNESP N. 2700/2017) e da Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP N. 2019/21510-3) na graduação.

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001.

# RESUMO

Vários problemas podem surgir quando a adaptação de software é realizada de maneira manual, principalmente com relação à inserção involuntária de incertezas por parte dos desenvolvedores. Portanto, a adaptação manual é um tipo de atividade propensa a erros que pode consumir muito tempo, esforço e dinheiro. Nesse contexto, o Software Autoadaptativo (em inglês, *Self-adaptive Software* – SaS) é um tipo especial de sistema de software que é capaz de se modificar em tempo de execução para lidar com as mudanças no seu ambiente operacional. De acordo com essa característica, o SaS pode ser adotado com uma solução para lidar com a automação da adaptação de software ao mesmo tempo que minimiza o envolvimento dos desenvolvedores. Na literatura científica podem ser encontrados vários exemplos de SaS relacionados com diferentes domínios de software, porém os tipos de SaS associados com a persistência de dados ainda são pouco investigados. Tais tipos de SaS abordam um cenário de execução em que os ciclos de autoadaptação estrutural do sistema quebram a compatibilidade com o esquema de banco de dados que foi estabelecido na etapa de projeto. Tendo em vista esse problema, foi conduzido um mapeamento sistemático da literatura (em inglês, *Systematic Mapping Study* – SMS) para estabelecer um panorama abrangente sobre a evolução do esquema de dados no domínio de SaS. A partir dos resultados desse SMS, foram levantadas as evidências sobre a importância desse tema de pesquisa, porém não aplicadas de maneira direta no domínio de SaS. Assim, o principal objetivo deste projeto de mestrado acadêmico é contribuir com a área de SaS através da proposição da biblioteca DynaSchema, que consiste em uma nova solução para lidar com a evolução do esquema de dados nesse domínio de software. Em síntese, essa solução atua como uma camada intermediária entre o SaS e seu banco de dados relacional, permitindo que o SaS possa adaptar seu esquema de dados ao mesmo tempo que a biblioteca gerencia as mudanças necessárias no banco de dados. Além disso, vale destacar que o SMS conduzido nesta dissertação de mestrado pode ser utilizado como um norte para o desenvolvimento de soluções que lidam com a evolução do esquema de dados. Por fim, foi elaborado um estudo de caso para avaliar a biblioteca DynaSchema, onde foram explorados cenários de adaptação que lidam com a evolução do esquema de dados do SaS.

**Palavras-chave:** evolução do esquema de dados; software autoadaptativo; biblioteca; dynaschema.

# ABSTRACT

Several problems can arise when software adaptation is performed manually, mainly regarding developers' involuntary insertion of uncertainties. Therefore, manual adaptation is an error-prone activity that consumes time, effort, and money. In this context, Self-addaptive Software (SaS) is a special type of software system capable of modifying itself at runtime to deal with changes in its operating environment. According to this feature, SaS can be adopted as a solution to handle the automation of software adaptation while minimizing developer involvement. Several examples of SaS related to different software domains can be found in the scientific literature, but the SaS types associated with data persistence are still slightly investigated. Such systems address an execution scenario in which the system's structural self-adaptation cycles break up compatibility with the database schema established in the design stage. Because of this problem, a Systematic Mapping Study (SMS) was conducted to establish a comprehensive overview of the evolution of the data schema in the SaS domain. Based on the results of this SMS, evidence was retrieved about the importance of this research topic, but not directly applied in the SaS domain. Thus, the main goal of this academic master's project is to contribute to the SaS area by proposing the DynaSchema library, which consists of a new solution to deal with the evolution of the data schema in this software domain. In short, this solution acts as an intermediate layer between SaS and its relational database, enabling SaS to adapt its data schema at the same time as the library manages the necessary changes in the database. Furthermore, it is worth highlighting that the SMS conducted in this master's dissertation can be used as a guide to support the development of solutions that deal with the evolution of the data schema. Finally, a case study was developed to evaluate the DynaSchema library, where adaptation scenarios that deal with the evolution of the SaS data schema were explored.

**Keywords:** data schema evolution; self-adaptive software; library; dynaschema.

# LISTA DE ILUSTRAÇÕES

|                                                                                       |    |
|---------------------------------------------------------------------------------------|----|
| Figura 1 – Processo de obtenção de uma arquitetura concreta . . . . .                 | 20 |
| Figura 2 – Ambiente operacional de um SaS . . . . .                                   | 22 |
| Figura 3 – Organização hierárquica das propriedades <i>self</i> -* . . . . .          | 22 |
| Figura 4 – Laço de controle MAPE . . . . .                                            | 24 |
| Figura 5 – Módulos da arquitetura de referência RA4SaS . . . . .                      | 26 |
| Figura 6 – Processo de geração do código fonte das entidades de software . . . . .    | 27 |
| Figura 7 – Metamodelo de uma entidade de software . . . . .                           | 28 |
| Figura 8 – Ferramenta de modelagem DSLModeler4SaS . . . . .                           | 29 |
| Figura 9 – Arquitetura simplificada de um sistema de banco de dados . . . . .         | 30 |
| Figura 10 – Arquitetura de três esquemas para um SGBD . . . . .                       | 32 |
| Figura 11 – Distribuição dos estudos por ano de publicação . . . . .                  | 37 |
| Figura 12 – Distribuição dos estudos por tipo de publicação . . . . .                 | 38 |
| Figura 13 – Distribuição dos estudos por base de pesquisa e frente de busca . . . . . | 39 |
| Figura 14 – Distribuição dos estudos por instituição e país . . . . .                 | 40 |
| Figura 15 – Distribuição dos estudos por tipo de solução . . . . .                    | 44 |
| Figura 16 – Domínios de aplicação abordados pelos estudos . . . . .                   | 45 |
| Figura 17 – Tipos de modelos de dados abordados pelos estudos . . . . .               | 50 |
| Figura 18 – Abordagem para a evolução do esquema de dados . . . . .                   | 53 |
| Figura 19 – Atributos não funcionais abordados pelos estudos . . . . .                | 54 |
| Figura 20 – Métodos de avaliação abordados pelos estudos . . . . .                    | 56 |
| Figura 21 – Resumo comparativo entre os estudos . . . . .                             | 58 |
| Figura 22 – Exemplo de execução da autoadaptação de um SaS . . . . .                  | 63 |
| Figura 23 – Contexto de funcionamento da biblioteca DynaSchema . . . . .              | 65 |
| Figura 24 – Arquitetura da biblioteca DynaSchema . . . . .                            | 71 |
| Figura 25 – Diagrama de classes do módulo Gerenciador de Estado . . . . .             | 74 |
| Figura 26 – Diagrama de classes do módulo Gerenciador de Notificação . . . . .        | 77 |
| Figura 27 – Funcionamento de mecanismo de notificação . . . . .                       | 79 |
| Figura 28 – Diagrama de classes do módulo de Dialeto . . . . .                        | 80 |
| Figura 29 – Processo de instanciação de um objeto de dialeto SQL . . . . .            | 82 |

|                                                                                                                 |     |
|-----------------------------------------------------------------------------------------------------------------|-----|
| Figura 30 – Diagrama de classes do módulo Gerenciador de Esquema . . . . .                                      | 83  |
| Figura 31 – Diagrama de classes do Módulo Gerenciador de Persistência . . . . .                                 | 88  |
| Figura 32 – Funcionamento do mecanismo de persistência . . . . .                                                | 90  |
| Figura 33 – Diagrama de classes do Módulo Gerenciador de Migração . . . . .                                     | 92  |
| Figura 34 – Passos para as duas primeiras etapas do exemplo de funcionamento para migração de esquema . . . . . | 95  |
| Figura 35 – Passos para terceira etapa do exemplo de funcionamento para migração de esquema . . . . .           | 97  |
| Figura 36 – Passos para quarta etapa do exemplo de funcionamento para migração de esquema . . . . .             | 98  |
| Figura 37 – Passos para quinta etapa do exemplo de funcionamento para migração de esquema . . . . .             | 99  |
| Figura 38 – Diagrama de classes da interface da biblioteca DynaSchema . . . . .                                 | 100 |
| Figura 39 – Modelo UML do sistema SisVenda . . . . .                                                            | 104 |
| Figura 40 – Projeto do sistema SisVenda na ferramenta de modelagem DSLModeler4SaS                               | 110 |
| Figura 41 – Diagrama de classes do sistema SisVenda . . . . .                                                   | 111 |
| Figura 42 – Geração do sistema SisVenda através da ferramenta de modelagem DSLModeler4SaS . . . . .             | 113 |
| Figura 43 – Contexto de funcionamento do sistema SisVenda . . . . .                                             | 115 |
| Figura 44 – Primeira versão do esquema de banco de dados para o sistema SisVenda . .                            | 118 |
| Figura 45 – Segunda versão do modelo UML do sistema SisVenda . . . . .                                          | 119 |
| Figura 46 – Esquema de banco de dados no estado misto . . . . .                                                 | 122 |
| Figura 47 – Segunda versão do esquema de banco de dados para o sistema SisVenda . .                             | 124 |
| Figura 48 – Dados da tabela de clientes do tipo pessoa física em estado misto . . . . .                         | 125 |
| Figura 49 – Quadro comparativo entre a biblioteca DynaSchema e os estudos mais completos do SMS . . . . .       | 126 |
| Figura 50 – Quantidade de estudos primários por fase de seleção . . . . .                                       | 149 |

# LISTA DE TABELAS

|                                                                                                                                     |     |
|-------------------------------------------------------------------------------------------------------------------------------------|-----|
| Tabela 1 – Estudos primários selecionados pelo SMS . . . . .                                                                        | 142 |
| Tabela 2 – Lista das bases de pesquisa consultadas pelo SMS . . . . .                                                               | 146 |
| Tabela 3 – <i>String</i> de pesquisa baseada na palavra-chave <i>self-adaptive</i> . . . . .                                        | 147 |
| Tabela 4 – <i>String</i> de pesquisa na palavra-chave <i>zero-downtime</i> . . . . .                                                | 148 |
| Tabela 5 – Quantidade de estudos primários selecionados por fase de filtragem, base de<br>publicação e frente de pesquisa . . . . . | 150 |
| Tabela 6 – Formulário de extração de dados . . . . .                                                                                | 150 |

# LISTA DE ABREVIATURAS E SIGLAS

|                |                                                                                                                                       |
|----------------|---------------------------------------------------------------------------------------------------------------------------------------|
| API            | <i><u>A</u>pplication <u>P</u>rogramming <u>I</u>nterface</i>                                                                         |
| CEP            | <i><u>C</u>ódigo de <u>E</u>ndereçamento <u>P</u>ostal</i>                                                                            |
| CE             | <i><u>C</u>ritério de <u>E</u>xclusão</i>                                                                                             |
| CI             | <i><u>C</u>ritério de <u>I</u>nclusão</i>                                                                                             |
| CNPJ           | <i><u>C</u>adastro <u>N</u>acional da <u>P</u>essoa <u>J</u>urídica</i>                                                               |
| CPF            | <i><u>C</u>adastro de <u>P</u>essoa <u>F</u>ísica</i>                                                                                 |
| DDL            | <i><u>D</u>ata <u>D</u>efinition <u>L</u>anguage</i>                                                                                  |
| DSLModeler4SaS | <i><u>D</u>omain-<u>S</u>pecific <u>L</u>anguage <u>M</u>odeler for <u>S</u>elf-<u>a</u>daptive <u>S</u>oftware</i>                   |
| DSL            | <i><u>D</u>omain-<u>S</u>pecific <u>L</u>anguage</i>                                                                                  |
| eLanguage      | <i><u>e</u>ntity <u>L</u>anguage</i>                                                                                                  |
| ETL            | <i><u>E</u>xtraction, <u>T</u>ransformation and <u>L</u>oading</i>                                                                    |
| FB             | <i><u>F</u>rente de <u>B</u>usca</i>                                                                                                  |
| IBM            | <i><u>I</u>nternational <u>B</u>usiness <u>M</u>achines</i>                                                                           |
| IE             | <i><u>I</u>nscrição <u>E</u>stadual</i>                                                                                               |
| JDK            | <i><u>J</u>ava <u>D</u>evelopment <u>K</u>it</i>                                                                                      |
| JPA            | <i><u>J</u>ava <u>P</u>ersistence <u>A</u>PI</i>                                                                                      |
| MAPE-K         | <i><u>M</u>onitor, <u>A</u>nalyze, <u>P</u>lan, <u>E</u>xecute over <u>K</u>nowledge</i>                                              |
| MAPE           | <i><u>M</u>onitor, <u>A</u>nalyze, <u>P</u>lan, <u>E</u>xecute</i>                                                                    |
| OLAP           | <i><u>O</u>n-<u>L</u>ine <u>A</u>nalytical <u>P</u>rocessing</i>                                                                      |
| ORM            | <i><u>O</u>bject-<u>R</u>elational <u>M</u>apping</i>                                                                                 |
| QP             | <i><u>Q</u>uestão de <u>P</u>esquisa</i>                                                                                              |
| RA4SaS         | <i><u>R</u>eference <u>A</u>rchitecture for <u>S</u>elf-<u>a</u>daptive <u>S</u>oftware</i>                                           |
| RA4SelfMobApps | <i><u>R</u>eference <u>A</u>rchitecture for <u>S</u>elf-<u>a</u>daptive <u>S</u>ervice-oriented <u>M</u>obile <u>A</u>pplications</i> |
| SAS            | <i><u>S</u>elf-<u>a</u>daptive <u>S</u>oftware</i>                                                                                    |
| SDL            | <i><u>S</u>torage <u>D</u>efinition <u>L</u>anguage</i>                                                                               |
| SGBD           | <i><u>S</u>istema <u>G</u>erenciador de <u>B</u>anco de <u>D</u>ados</i>                                                              |
| SMS            | <i><u>S</u>ystematic <u>M</u>apping <u>S</u>tudy</i>                                                                                  |
| SQL            | <i><u>S</u>tructured <u>Q</u>uery <u>L</u>anguage</i>                                                                                 |
| UML            | <i><u>U</u>nified <u>M</u>odeling <u>L</u>anguage</i>                                                                                 |
| VDL            | <i><u>V</u>iew <u>D</u>efinition <u>L</u>anguage</i>                                                                                  |
| XML            | <i><u>e</u>Xtensible <u>M</u>arkup <u>L</u>anguage</i>                                                                                |
| ZDT            | <i><u>Z</u>ero-<u>D</u>owntime</i>                                                                                                    |

# SUMÁRIO

|            |                                                                                                                               |           |
|------------|-------------------------------------------------------------------------------------------------------------------------------|-----------|
| <b>1</b>   | <b>INTRODUÇÃO . . . . .</b>                                                                                                   | <b>11</b> |
| <b>1.1</b> | <b>Contextualização . . . . .</b>                                                                                             | <b>11</b> |
| <b>1.2</b> | <b>Motivação . . . . .</b>                                                                                                    | <b>12</b> |
| <b>1.3</b> | <b>Objetivos . . . . .</b>                                                                                                    | <b>14</b> |
| <b>1.4</b> | <b>Organização da Dissertação . . . . .</b>                                                                                   | <b>15</b> |
| <b>2</b>   | <b>FUNDAMENTAÇÃO TEÓRICA . . . . .</b>                                                                                        | <b>16</b> |
| <b>2.1</b> | <b>Arquitetura de Software . . . . .</b>                                                                                      | <b>16</b> |
| <b>2.2</b> | <b>Sistemas Autoadaptativos . . . . .</b>                                                                                     | <b>21</b> |
| <b>2.3</b> | <b>Reference Architecture for Self-adaptive Software . . . . .</b>                                                            | <b>25</b> |
| <b>2.4</b> | <b>Sistemas de Banco de Dados . . . . .</b>                                                                                   | <b>29</b> |
| <b>2.5</b> | <b>Considerações Finais . . . . .</b>                                                                                         | <b>34</b> |
| <b>3</b>   | <b>EVOLUÇÃO DO ESQUEMA DE DADOS PARA O DOMÍNIO DE SISTEMAS AUTOADAPTATIVOS . . . . .</b>                                      | <b>35</b> |
| <b>3.1</b> | <b>Mapeamento Sistemático da Literatura . . . . .</b>                                                                         | <b>35</b> |
| 3.1.1      | QP1: Quais são os aspectos demográficos dos estudos? . . . . .                                                                | 36        |
| 3.1.2      | QP2: Quais tipos de soluções são abordadas pelos estudos? . . . . .                                                           | 41        |
| 3.1.3      | QP3: Em quais domínios de aplicação as soluções propostas pelos estudos estão inseridas? . . . . .                            | 44        |
| 3.1.4      | QP4: Quais tipos de modelos de dados são abordados pelos estudos? . . . . .                                                   | 47        |
| 3.1.5      | QP5: Quais tipos de abordagens são utilizadas pelos estudos para tratar o processo de evolução do esquema de dados? . . . . . | 49        |
| 3.1.6      | QP6: Quais atributos não funcionais foram considerados no projeto das soluções propostas pelos estudos? . . . . .             | 54        |
| 3.1.7      | QP7: Como as soluções propostas pelos estudos foram avaliadas? . . . . .                                                      | 55        |
| <b>3.2</b> | <b>Discussão de Resultados . . . . .</b>                                                                                      | <b>57</b> |
| <b>3.3</b> | <b>Considerações Finais . . . . .</b>                                                                                         | <b>61</b> |
| <b>4</b>   | <b>BIBLIOTECA DYNASCHEMA . . . . .</b>                                                                                        | <b>62</b> |

|            |                                                               |            |
|------------|---------------------------------------------------------------|------------|
| <b>4.1</b> | <b>Projeto da Biblioteca</b>                                  | <b>62</b>  |
| 4.1.1      | Exemplo de Execução                                           | 63         |
| 4.1.2      | Componentes Funcionais                                        | 65         |
| 4.1.3      | Conjunto de Funcionalidades                                   | 67         |
| 4.1.4      | Arquitetura Interna                                           | 70         |
| <b>4.2</b> | <b><i>Design</i> dos Módulos e da Interface da Biblioteca</b> | <b>73</b>  |
| 4.2.1      | Módulo Gerenciador de Estado                                  | 73         |
| 4.2.2      | Módulo Gerenciador de Notificação                             | 76         |
| 4.2.3      | Módulo de Dialeto                                             | 78         |
| 4.2.4      | Módulo Gerenciador de Esquema                                 | 82         |
| 4.2.5      | Módulo Gerenciador de Persistência                            | 87         |
| 4.2.6      | Módulo Gerenciador de Migração                                | 91         |
| 4.2.7      | Interface da Biblioteca                                       | 98         |
| <b>4.3</b> | <b>Considerações Finais</b>                                   | <b>102</b> |
| <b>5</b>   | <b>ESTUDO DE CASO</b>                                         | <b>103</b> |
| <b>5.1</b> | <b>Sistema Alvo</b>                                           | <b>103</b> |
| <b>5.2</b> | <b>Implementação do Sistema Alvo</b>                          | <b>108</b> |
| <b>5.3</b> | <b>Primeira Versão do Esquema de Dados do Sistema Alvo</b>    | <b>114</b> |
| <b>5.4</b> | <b>Alteração do Esquema de Dados do Sistema Alvo</b>          | <b>119</b> |
| <b>5.5</b> | <b>Discussão de Resultados</b>                                | <b>125</b> |
| <b>5.6</b> | <b>Considerações Finais</b>                                   | <b>131</b> |
| <b>6</b>   | <b>CONCLUSÃO</b>                                              | <b>132</b> |
| <b>6.1</b> | <b>Contribuições</b>                                          | <b>133</b> |
| <b>6.2</b> | <b>Dificuldades e Limitações</b>                              | <b>134</b> |
| <b>6.3</b> | <b>Trabalhos Futuros</b>                                      | <b>135</b> |
|            | <b>REFERÊNCIAS</b>                                            | <b>137</b> |
|            | <b>APÊNDICE A – LISTA DE ESTUDOS PRIMÁRIOS</b>                | <b>142</b> |
|            | <b>APÊNDICE B – PROTOCOLO DE PESQUISA</b>                     | <b>144</b> |
| <b>B.1</b> | <b>Questões de Pesquisa</b>                                   | <b>144</b> |

|            |                                               |            |
|------------|-----------------------------------------------|------------|
| <b>B.2</b> | <b>Estratégia de Pesquisa . . . . .</b>       | <b>145</b> |
| <b>B.3</b> | <b>Condução da Pesquisa . . . . .</b>         | <b>148</b> |
| <b>B.4</b> | <b>Extração e Síntese dos Dados . . . . .</b> | <b>150</b> |
| <b>B.5</b> | <b>Limitações . . . . .</b>                   | <b>151</b> |

# 1 Introdução

## 1.1 Contextualização

Os sistemas de software têm desempenhado um papel fundamental na nossa sociedade, sendo responsáveis pela automatização da execução de tarefas em diferentes segmentos, a saber: instituições públicas, privadas e financeiras; sistemas de comunicação; aeroportos; meios de entretenimento; entre outras áreas de aplicação. Além disso, tais sistemas têm sido projetados para atuar em situações adversas e/ou sujeitas às mudanças, permitindo a manutenção da sua disponibilidade de execução em 24/7 (ou seja, 24 horas por dia e 7 dias por semana), sem interrupções ou intervenções humanas. De acordo com essas demandas, é essencial que esses sistemas de software possuam a capacidade de lidar com as incertezas, cujas causas podem variar desde mudanças no seu ambiente operacional até em alterações nas necessidades de seus usuários. O Software Autoadaptativo (em inglês, *Self-adaptive Software* – SaS) é um tipo especial de sistema de software que é capaz de lidar com as mudanças em tempo de execução para atender as novas necessidades dos usuários ou reagir de forma autônoma às modificações em seu ambiente de execução (por exemplo, degradação de qualidade, sobrecarga de operações, entre outras adversidades) (CHENG *et al.*, 2009; LEMOS *et al.*, 2013; WEYNS, 2019).

Vários problemas podem surgir quando a adaptação de software é realizada de maneira manual, principalmente devido à inserção involuntária de incertezas por parte dos desenvolvedores (ou seja, dos seres humanos). Nesse sentido, pode-se dizer que a adaptação manual de software é uma atividade propensa a erros que pode consumir muito tempo, esforço e dinheiro (HILLENBRAND *et al.*, 2020). Tendo em vista esse cenário problemático, abordagens automatizadas de adaptação têm sido adotadas com uma alternativa viável para maximizar a rapidez de implementação do SaS e, em paralelo, minimizar o envolvimento dos seres humanos (CHENG *et al.*, 2009; SALEHIE; TAHVILDARI, 2009; LEMOS *et al.*, 2013). Por outro lado, a preservação da integridade entre os modelos gerados na fase de projeto e o código fonte do SaS durante seus ciclos de autoadaptação em tempo de execução ainda é um desafio que foi pouco investigado pela comunidade científica de SaS e pelos profissionais da indústria que trabalham com esse tipo de sistema de software (HILLENBRAND *et al.*, 2020; CHENG *et al.*, 2009; AFFONSO; PASSINI; NAKAGAWA, 2019; WEYNS, 2019; LEMOS *et al.*, 2013).

Uma iniciativa da comunidade de SaS tem mantido um repositório que é atualizado ao longo dos anos para fornecer exemplos, desafios e soluções para a área de SaS (VOGEL, 2023). Os exemplos de SaS contidos nesse repositório estão relacionados com diferentes domínios de software, a saber: serviços *web*, veículos autônomos, sistemas ciber-físicos, processamento de vídeo, entre outras áreas de aplicação. Embora esse repositório seja uma referência para a comunidade SaS, outros tipos desse sistema de software também podem ser encontrados na literatura. Como exemplo, esta dissertação de mestrado se concentra especificamente nos tipos de SaS que precisam armazenar suas informações em um banco de dados relacional. Nesse cenário, um SaS pode passar por ciclos de autoadaptação estrutural que quebram a compatibilidade com o esquema de banco de dados que foi estabelecido na sua etapa de projeto. Para lidar com esse problema de integridade, esse SaS deve apresentar um mecanismo para lidar com as diferentes versões do seu esquema de dados em tempo de execução (HILLENBRAND *et al.*, 2020; STÖRL; KLETTKE; SCHERZINGER, 2020). Portanto, esse mecanismo deve alterar o esquema de banco de dados para acompanhar os ciclos de autoadaptação do SaS ao longo do tempo.

## 1.2 Motivação

Existem vários estudos na literatura científica que exploram diferentes aspectos de um SaS. Tais estudos contextualizam os pontos principais que definem esse tipo de sistema de software (BRUN *et al.*, 2009; CHENG *et al.*, 2009; KRUPITZER *et al.*, 2015; SALEHIE; TAHVILDARI, 2009; WEYNS *et al.*, 2013), as propriedades de autoadaptação para resolver um problema específico (KRUPITZER *et al.*, 2015; SALEHIE; TAHVILDARI, 2009), as técnicas para o controle do processo de autoadaptação (BRUN *et al.*, 2009; CHENG *et al.*, 2009; SALEHIE; TAHVILDARI, 2009; WEYNS *et al.*, 2013), as questões de engenharia envolvidas no desenvolvimento de uma entidade de software (KRUPITZER *et al.*, 2015; SALEHIE; TAHVILDARI, 2009), entre outros assuntos.

Baseado no contexto exposto, Affonso e Nakagawa (2013) propuseram a arquitetura de referência RA4SaS (do inglês, *Reference Architecture for Self-adaptive Software*) para apoiar o desenvolvimento de um SaS. Essa arquitetura tem sido mantida pelo orientador deste projeto de mestrado acadêmico, servindo de base para desenvolvimento de outras pesquisas no domínio de SaS. Como exemplo, pode-se citar o *framework* para apoiar à tomada de decisão em um SaS (AFFONSO *et al.*, 2015), a ferramenta de modelagem para apoiar à tomada de decisão fornecida

pelo *framework* anterior (AFFONSO; LEITE; NAKAGAWA, 2016), o gerador de código fonte baseado em *templates* (BENATO; AFFONSO; NAKAGAWA, 2017) e a arquitetura de referência para a criação de aplicativos móveis autoadaptativos orientados a serviços (AFFONSO; PASSINI; NAKAGAWA, 2019). O tema desta dissertação de mestrado está inserido no contexto da linha de pesquisa de aprimoramento da arquitetura de referência RA4SaS, cuja última evolução permitiu que o SaS possa raciocinar sobre a persistência dos seus dados. Nesse sentido, um SaS que utiliza a arquitetura RA4SaS está preparado para autoadaptar seu modelo de dados, porém o esquema do banco de dados não acompanha essas mudanças de maneira automática.

Motivado pela problemática apresentada nesta seção, pelo cenário exposto na Seção 1.1 e pela necessidade de se obter uma visão geral sobre a temática desta dissertação de mestrado, foi conduzida uma revisão da literatura sobre a evolução do esquema de dados para o domínio de SaS. Para isso, foi adotada a técnica de mapeamento sistemático (em inglês, *Systematic Mapping Study* – SMS), que permite uma avaliação mais completa e justa sobre um tópico de pesquisa através de uma metodologia confiável, rigorosa, auditável e isenta de influências pessoais dos pesquisadores (KITCHENHAM; DYBA; JORGENSEN, 2004; KITCHENHAM; CHARTERS, 2007; PETERSEN; VAKKALANKA; KUZNIARZ, 2015). Em relação aos interesses de investigação, esse SMS buscou traçar um panorama abrangente em relação aos atributos não funcionais, modelos de dados, estratégias de migração e aspectos de *design* que foram abordados no projeto das soluções que lidam com a evolução do esquema de dados.

Dentre os resultados desse SMS, vale destacar a seleção de 17 estudos primários (listados na Tabela 1 do Apêndice A) para sistematizar os conhecimentos sobre a temática de pesquisa explorado nesta dissertação de mestrado. A partir de uma análise geral, pode-se dizer que a evolução do esquema de dados no domínio de SaS é um tema amplo e precisa ser investigado com maior profundidade. Nesse sentido, o panorama reportado no SMS do Capítulo 3 apresenta uma importante contribuição para a comunidade de SaS e seus *stakeholders*. Embora os estudos selecionados nesse SMS apresentem um conjunto comum de características sobre diferentes aspectos de investigação (ou seja, domínios de aplicação, modelos de dados, estratégias de migração e atributos não funcionais), não foi observado um consenso sobre o que deve ser aceito como uma solução no contexto da evolução do esquema de dados no domínio de SaS.

Diante do conteúdo reportado nesta seção, a persistência de dados para o domínio de SaS e os problemas relacionadas à evolução do esquema de dados desse tipo de sistema são assuntos que vêm sendo investigados nos últimos anos e ainda possuem potencial de exploração pela

comunidade científica. Nesse sentido, pode-se dizer que nenhuma das iniciativas, encontradas na investigação conduzida neste projeto de mestrado acadêmico, contempla totalmente as lacunas e temas em aberto que foram expostas nesta seção, a saber: (i) *a dependência entre a estrutura do SaS e o esquema de banco de dados*; (ii) *a evolução do esquema de banco de dados*; (iii) *a migração dos dados da versão antiga do esquema de banco de dados para a nova versão*; (iv) *a manutenção simultânea das versões antiga e nova do esquema de banco de dados em um estado transitório* e (v) *a capacidade de recuperação em relação a uma operação mal executada*. Diante do exposto, tais características foram consideradas como fatores motivadores para a proposição de uma nova solução para a evolução do esquema de dados no domínio de SaS.

## 6 Conclusão

Neste projeto de mestrado acadêmico foi apresentada uma nova solução que é capaz de lidar com o problema da evolução do esquema de dados no domínio de SaS. Em síntese, essa solução foi batizada com o nome DynaSchema e suas funcionalidades foram empacotadas em uma biblioteca que pode ser reutilizada nos projetos de SaS. Para isso, tal biblioteca atua como uma camada intermediária (ou seja, um *middleware*) entre o SaS e seu banco de dados relacional, onde o SaS deve informar a biblioteca quais mudanças foram feitas no esquema de dados da aplicação e quando devem ser executadas etapas da evolução desse esquema de dados. Nesse sentido, a biblioteca DynaSchema possui três responsabilidades principais, a saber: (i) estabelecer o meio para persistir as informações no banco de dados; (ii) modificar o esquema de dados de acordo com as alterações estruturais exigidas pelo SaS; e (iii) notificar o SaS em relação à execução das etapas para a evolução do esquema de dados da aplicação.

Do ponto de vista operacional, o SaS que utiliza a biblioteca DynaSchema como um mecanismo de persistência tem a capacidade de inserir, alterar, remover e consultar suas informações em um banco de dados relacional. Durante os ciclos de autoadaptação do SaS, essa biblioteca é utilizada para evoluir o esquema de banco de dados em tempo de execução, permitindo que o SaS possa acessar suas informações armazenadas mesmo após uma mudança estrutural no seu modelo de dados. Por sua vez, essa biblioteca notifica o SaS em relação ao estado de execução das etapas para a evolução do esquema de dados da aplicação, assim o SaS pode detectar as mudanças no seu ambiente operacional.

De acordo com o projeto apresentado nesta dissertação de mestrado, o desenvolvimento desse tipo de solução envolve uma atividade complexa com uma carga elevada de conceitos e conhecimentos de mais de um domínio de software. Inicialmente, no Capítulo 2 foram apresentados os conceitos e as definições abordados no projeto da biblioteca DynaSchema. Em seguida, no Capítulo 3 foi conduzido um SMS sobre a evolução do esquema de dados para o domínio de SaS, cujos resultados evidenciaram a relevância do tema de pesquisa desta dissertação de mestrado e contribuíram para o projeto da biblioteca DynaSchema. Nos Capítulos 4 e 5 foram reportados os detalhes de projeto e da avaliação dessa biblioteca, respectivamente. Por fim, neste capítulo são apresentadas as conclusões desta dissertação de mestrado da seguinte maneira: na Seção 6.1 são abordadas as contribuições dessa dissertação; na Seção 6.2 são reportadas as prin-

cipais dificuldades enfrentadas durante o desenvolvimento da biblioteca DynaSchema e suas limitações; e na Seção 6.3 são apresentados as atividades para trabalhos futuros.

## 6.1 Contribuições

No Capítulo 3 foi conduzido um SMS baseado em sete QPs que seguiu as diretrizes propostas por Petersen, Vakkalanka e Kuzniarz (2015). Com base nas evidências levantadas nos resultados desse SMS, foi possível estabelecer um panorama amplo sobre a evolução do esquema de dados no domínio de SaS (ou seja, a temática de pesquisa deste projeto de mestrado acadêmico) que revelou as lacunas e os assuntos que precisam ser investigados com mais detalhes. Embora a principal contribuição deste projeto de mestrado seja a biblioteca DynaSchema, a revisão da literatura apresentada no Capítulo 3 também pode ser destacada com uma contribuição tanto para os pesquisadores quanto para os profissionais da indústria de software, pois esse SMS pode ser utilizado com um guia para nortear o desenvolvimento e/ou aprimoramento de novas soluções relacionadas com a evolução do esquema de dados no domínio de SaS.

A biblioteca DynaSchema desenvolvida neste projeto de mestrado acadêmico constitui a principal contribuição para os pesquisadores e os profissionais da indústria de software. Nesse sentido, essa biblioteca apresenta uma nova solução para a evolução do esquema de dados no domínio de SaS que condensa as melhores evidências desse tema de pesquisa, além de abordar as principais lacunas de investigação relacionadas com essa temática. Vale lembrar que a biblioteca DynaSchema foi projetada para atuar como uma camada intermediária entre o SaS e seu banco de dados relacional, assim estabelecendo uma interface de utilização mais abstrata e transparente aos desenvolvedores de software. Além disso, vale destacar que essa biblioteca adotou alguns padrões de projeto e boas práticas de engenharia, permitindo a extensão dessa solução sem a necessidade de grandes esforços de desenvolvimento. De acordo com o escopo deste projeto de mestrado, o *design* da biblioteca DynaSchema foi planejado para maximizar a rapidez de ampliação das capacidades dessa solução para lidar com novos interesses (por exemplo, como a facilidade de adição de novas estratégias de dialeto SQL).

O estudo de caso conduzido no Capítulo 5 também pode ser considerado uma contribuição deste projeto de mestrado, pois fornece alguns aspectos importantes sobre a aplicabilidade da biblioteca DynaSchema em um cenário real de funcionamento. Apesar do estudo de caso abordar a aplicação dessa biblioteca em um sistema específico, as operações de migração do

esquema e dos dados contidas no processo de evolução do esquema de dados do SaS são consideradas essenciais e devem ser executadas em qualquer tipo de sistema, independentemente do seu domínio de software. Além disso, o estudo de caso apresentado neste projeto de mestrado acadêmico teve como objetivo primário a validação da biblioteca DynaSchema como uma solução factível para lidar com a evolução do esquema de dados no domínio de SaS, cujos resultados permitiram a coleta de parâmetros relevantes sobre seu comportamento e a aderência aos requisitos de projeto.

A biblioteca DynaSchema foi projetada originalmente para operar no domínio de SaS, porém acredita-se que essa solução possui o potencial para atuar em outros domínios de software. Nesse sentido, tal biblioteca apresenta um conjunto de funcionalidades que atende as necessidades de um sistema de software tradicional, além de possuir uma interface abstrata que pode ser utilizada de maneira genérica em qualquer aplicação (ou seja, independente de projeto). Portanto, a biblioteca DynaSchema pode servir como uma referência na elaboração de futuras iniciativas para lidar com a evolução do esquema de dados em diferentes domínios de software.

## 6.2 Dificuldades e Limitações

De acordo com as evidências obtidas no SMS conduzido no Capítulo 3, foram observados vários aspectos que podem ser explorados em relação a evolução do esquema de dados no domínio de SaS. Devido a restrição de tempo para a conclusão deste projeto de mestrado acadêmico, foi estabelecido um escopo de trabalho (Seção 4.1) para projetar e desenvolver a biblioteca DynaSchema que constituiu uma nova solução para lidar com esse problema. Nesse sentido, tal escopo priorizou a compatibilidade dessa biblioteca com diferentes tipos de SaS existentes na literatura (como os exemplos contidos no repositório mantido por Vogel (2023)), além de considerar as necessidades do nosso grupo de pesquisa. Vale salientar que o projeto de *design* da biblioteca DynaSchema foi direcionado para a possibilidade de generalização e flexibilização dessa solução em relação ao suporte de novas funcionalidades. Apesar desses cuidados, essa biblioteca não pode ser considerada uma solução do tipo “bala de prata”, que consegue cobrir todas as necessidades exigidas por um SaS do mundo real.

Devido à natureza multidisciplinar da biblioteca DynaSchema, foram encontrados vários desafios durante a implementação prática dessa solução, a saber: o desenvolvimento de modelos eficazes para representar a estrutura das entidades do SaS e o plano de migração; a necessidade

de implementar a interface da JPA como maneira de aumentar a aceitação da biblioteca através do reaproveitamento de conceitos bem estabelecidos; a execução das operações de evolução do esquema de dados da aplicação em tempo de execução; a utilização intensiva da API de reflexão da linguagem Java para obter as informações estruturais e de persistência das entidades de software do SaS; a criação de múltiplas versões “virtuais” de esquema de banco de dados que precisam de um mecanismo de sincronização de dados entre si; e o processo de migração de dados que deve lidar com duas versões estruturais diferentes do esquema de banco de dados e que pode exigir algum tipo de mapeamento de dados.

Por fim, o estudo de caso conduzido no Capítulo 5 apresentou uma maneira de avaliar a biblioteca DynaSchema, cujos resultados revelaram um cenário favorável sobre o funcionamento e a viabilidade dessa solução em uma aplicação prática. Entretanto não foi possível realizar neste momento uma análise em relação às questões de desempenho dessa biblioteca. Essa limitação ocorreu por diferentes fatores, a saber: (i) a restrição de tempo inerente a um projeto de mestrado acadêmico; (ii) a complexidade do projeto e implementação da biblioteca DynaSchema; e (iii) pela própria natureza desse tipo de atividade.

## 6.3 Trabalhos Futuros

Durante a condução das atividades prevista para este projeto de mestrado, foram identificadas algumas tarefas que podem ser realizadas como trabalhos futuros, a saber:

- **Condução de outros estudos de casos e experimentos:** esta tarefa está relacionada com a avaliação do comportamento da biblioteca DynaSchema em outros cenários de utilização, como na verificação da experiência do usuário em relação a facilidade de utilização dessa biblioteca e a investigação do seu funcionamento em exemplos práticos da vida real, por exemplo. Nesse sentido, tais experimentos estão interessados na observação da completude dos processos de migração do esquema e dos dados da aplicação, na validação da suficiência das operações de manipulação do esquema e/ou dos dados e na aferição da performance da biblioteca em lidar com diferentes tamanhos de massas de dados;
- **Condução de estudos de casos em outros domínios de software:** no Capítulo 5 foi apresentado um estudo de caso para avaliar a biblioteca DynaSchema em relação a um sistema no contexto da arquitetura de referência RA4SaS voltada para o domínio de

sistemas da informação. Nesse sentido, avalia-se como uma atividade viável a verificação do comportamento dessa biblioteca em aplicações voltadas para o domínio de sistemas críticos, como um sistema para cuidado de pacientes dentro de um ambiente doméstico (CAMARGO *et al.*, 2024), por exemplo. Esses sistemas possuem níveis de criticidade mais elevado em relação a aplicação explorada no Capítulo 5, pois demandam alta confiabilidade e resiliência. No caso do sistema citado como exemplo, a ocorrência de falhas/anomalias pode resultar em sérios danos ao paciente que está sendo monitorado; e

- **Acompanhar a evolução do tema de pesquisa:** esta tarefa consiste na atualização do SMS reportado no Capítulo 3. Dessa maneira, espera-se observar as tendências sobre a evolução do esquema de dado no domínio de SaS, além de monitorar os interesses da comunidade científica e da indústria em relação a esse tipo de solução.

## REFERÊNCIAS

AFFONSO, F. J.; LEITE, G.; NAKAGAWA, E. Y. Dms-modeler: A tool for modeling decision-making systems for self-adaptive software domain. In: *The 28th International Conference on Software Engineering & Knowledge Engineering (SEKE' 16)*. [s.n.], 2016. p. 617–622. ISSN 2325-9086. Disponível em: <<https://doi.org/10.18293/SEKE2016-184>>. Citado na página 13.

AFFONSO, F. J. *et al.* A framework based on learning techniques for decision-making in self-adaptive software. In: *The 27th International Conference on Software Engineering and Knowledge Engineering. (SEKE' 2015)*. Wyndham Pittsburgh University Center, Pittsburgh, USA: Knowledge Systems Institute Graduate School, 2015. p. 24–29. ISSN 2325-9086. Disponível em: <<https://doi.org/10.18293/seke2015-125>>. Citado na página 12.

AFFONSO, F. J.; NAKAGAWA, E. Y. A reference architecture based on reflection for self-adaptive software. In: *2013 VII Brazilian Symposium on Software Components, Architectures and Reuse*. [s.n.], 2013. p. 129–138. Disponível em: <<https://doi.org/10.1109/SBCARS.2013.24>>. Citado 4 vezes na(s) página(s) 12, 20, 25 e 26.

AFFONSO, F. J.; PASSINI, W. F.; NAKAGAWA, E. Y. A reference architecture to support the development of mobile applications based on self-adaptive services. *Pervasive and Mobile Computing*, v. 53, p. 33–48, 2019. ISSN 1574-1192. Disponível em: <<https://doi.org/10.1016/j.pmcj.2019.01.001>>. Citado 3 vezes na(s) página(s) 11, 13 e 20.

ANGELOV, S.; GREFFEN, P.; GREEFHORST, D. A framework for analysis and design of software reference architectures. *Information and Software Technology*, v. 54, n. 4, p. 417 – 431, 2012. Disponível em: <<https://doi.org/10.1016/j.infsof.2011.11.009>>. Citado na página 20.

ASSOUDI, H.; LOUNIS, H. Self-healing data exchange process under evolving schemas: A new mapping adaptation approach based on self-optimization. In: *2011 IEEE 13th International Symposium on High-Assurance Systems Engineering*. [s.n.], 2011. p. 188–190. ISSN 1530-2059. Disponível em: <<https://doi.org/10.1109/HASE.2011.42>>. Citado 2 vezes na(s) página(s) 43 e 49.

ATZENI, P. *et al.* A runtime approach to model-generic translation of schema and data. *Information Systems*, v. 37, n. 3, p. 269–287, 2012. ISSN 0306-4379. Disponível em: <<https://doi.org/10.1016/j.is.2011.11.003>>. Citado 3 vezes na(s) página(s) 41, 48 e 51.

AZEROUAL, O.; JHA, M. Without data quality, there is no data migration. *BIG DATA AND COGNITIVE COMPUTING*, v. 5, n. 2, JUN 2021. Disponível em: <<https://doi.org/10.3390/bdcc5020024>>. Citado na página 51.

BASS, L.; CLEMENTS, P.; KAZMAN, R. *Software architecture in practice*. 2. ed. [S.l.]: Addison-Wesley Professional, 2003. ISBN 978-03-211-5495-8. Citado na página 20.

BENATO, G. S.; AFFONSO, F. J.; NAKAGAWA, E. Y. Infrastructure based on template engines for automatic generation of source code for self-adaptive software domain. In: *The 29th International Conference on Software Engineering & Knowledge Engineering (SEKE' 17)*. Wyndham Pittsburgh University Center, Pittsburgh, USA: [s.n.], 2017. p. 1–6. ISSN 2325-9086. Disponível em: <<https://doi.org/10.18293/SEKE2017-147>>. Citado 4 vezes na(s) página(s) 13, 26, 27 e 28.

BEURER-KELLNER, L. *et al.* A transformational approach to managing data model evolution of web services. *IEEE Transactions on Services Computing*, p. 1–1, 2022. ISSN 1939-1374. Disponível em: <<https://doi.org/10.1109/TSC.2022.3144613>>. Citado 3 vezes na(s) página(s) 41, 48 e 51.

BHATTACHERJEE, S. *et al.* Bullfrog: Online schema evolution via lazy evaluation. In: \_\_\_\_\_. *Proceedings of the 2021 International Conference on Management of Data*. New York, NY, USA: Association for Computing Machinery, 2021. p. 194–206. ISBN 9781450383431. Disponível em: <<https://doi.org/10.1145/3448016.3452842>>. Citado 3 vezes na(s) página(s) 43, 48 e 52.

BRUN, Y. *et al.* Engineering self-adaptive systems through feedback loops. In: \_\_\_\_\_. *Software Engineering for Self-Adaptive Systems*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009. p. 48–70. Disponível em: <[https://doi.org/10.1007/978-3-642-02161-9\\_3](https://doi.org/10.1007/978-3-642-02161-9_3)>. Citado 3 vezes na(s) página(s) 12, 21 e 23.

CAMARGO, M. P. d. O. *et al.* Ra4self-cps: A reference architecture for self-adaptive cyber-physical systems. *IEEE Latin America Transactions*, v. 22, n. 2, p. 113–125, 2024. Disponível em: <<https://doi.org/10.1109/TLA.2024.10412036>>. Citado na página 136.

CHENG, B. H. C. *et al.* Software engineering for self-adaptive systems: A research roadmap. In: \_\_\_\_\_. *Software Engineering for Self-Adaptive Systems*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009. p. 1–26. ISBN 978-3-642-02161-9. Disponível em: <[https://doi.org/10.1007/978-3-642-02161-9\\_1](https://doi.org/10.1007/978-3-642-02161-9_1)>. Citado 6 vezes na(s) página(s) 11, 12, 21, 23, 24 e 35.

COCKBURN, A. *Hexagonal architecture*. 2005. Disponível em: <<https://web.archive.org/web/20220919205430/https://alistair.cockburn.us/hexagonal-architecture/>>. Acesso em: 26 out. 2022. Citado na página 18.

DATE, C. J. *Sistemas de banco de dados*. 8. ed. [S.l.]: Elsevier Editora Ltda., 2004. ISBN 978-85-3521-273-0. Citado 4 vezes na(s) página(s) 30, 31, 32 e 33.

DIESTE, O.; GRIMÁN, A.; JURISTO, N. Developing search strategies for detecting relevant experiments. *Empirical Software Engineering*, v. 14, n. 5, p. 513–539, Oct 2009. ISSN 1573-7616. Disponível em: <<https://doi.org/10.1007/s10664-008-9091-7>>. Citado na página 146.

DYBA, T.; DINGSOYR, T.; HANSEN, G. K. Applying systematic reviews to diverse study types: An experience report. In: *First International Symposium on Empirical Software Engineering and Measurement (ESEM 2007)*. [s.n.], 2007. p. 225–234. Disponível em: <<https://doi.org/10.1109/ESEM.2007.59>>. Citado na página 146.

ELMASRI, R.; NAVATHE, S. B. *Sistemas de banco de dados*. 7. ed. [S.l.]: Pearson Education do Brasil, 2019. ISBN 978-85-430-2500-1. Citado 4 vezes na(s) página(s) 30, 31, 32 e 33.

FOWLER, M. *Padrões de arquitetura de aplicações corporativas*. [S.l.]: Bookman, 2006. ISBN 978-85-363-0638-4. Citado 3 vezes na(s) página(s) 16, 17 e 18.

GAMMA, E. *et al.* *Padrões de projetos: soluções reutilizáveis de software orientado a objetos*. [S.l.]: Bookman, 2000. ISBN 978-85-7307-610-3. Citado 2 vezes na(s) página(s) 18 e 19.

GÖTZ, S.; KÜHN, T. Models@run.time for object-relational mapping supporting schema evolution. In: \_\_\_\_\_. [s.n.], 2015. v. 1474, p. 41–50. Cited By 1. Disponível em: <<https://www.scopus.com/inward/record.uri?eid=2-s2.0-84954564351&partnerID=40&md5=ff323df66409bddb8c99cfda6e4737da>>. Citado 2 vezes na(s) página(s) 41 e 48.

HILLENBRAND, A. *et al.* Self-adapting data migration in the context of schema evolution in nosql databases. *Distributed and Parallel Databases*, 2021. ISSN 1573-7578. Disponível em: <<https://doi.org/10.1007/s10619-021-07334-1>>. Citado 2 vezes na(s) página(s) 42 e 52.

HILLENBRAND, A. *et al.* Towards self-adapting data migration in the context of schema evolution in nosql databases. In: *2020 IEEE 36th International Conference on Data Engineering Workshops (ICDEW)*. [s.n.], 2020. p. 133–138. Disponível em: <<https://doi.org/10.1109/ICDEW49219.2020.00013>>. Citado 4 vezes na(s) página(s) 11, 12, 51 e 52.

IBM. *An architectural blueprint for autonomic computing*. 2005. [On-line], World Wide Web. Disponível em: <<https://web.archive.org/web/20200209045605/http://www-03.ibm.com/autonomic/pdfs/ACBlueprintWhitePaperV7.pdf>>. Acesso em: 15 mar. 2020. Citado na página 25.

JONG, M. de; DEURSEN, A. van; CLEVE, A. Zero-downtime sql database schema evolution for continuous deployment. In: *2017 IEEE/ACM 39th International Conference on Software Engineering: Software Engineering in Practice Track (ICSE-SEIP)*. [s.n.], 2017. p. 143–152. Disponível em: <<https://doi.org/10.1109/ICSE-SEIP.2017.5>>. Citado 4 vezes na(s) página(s) 42, 52, 122 e 123.

KITCHENHAM, B.; CHARTERS, S. Guidelines for performing systematic literature reviews in software engineering. v. 2, jan 2007. Disponível em: <[https://www.researchgate.net/publication/302924724\\_Guidelines\\_for\\_performing\\_Systematic\\_Literature\\_Reviews\\_in\\_Software\\_Engineering](https://www.researchgate.net/publication/302924724_Guidelines_for_performing_Systematic_Literature_Reviews_in_Software_Engineering)>. Citado 2 vezes na(s) página(s) 13 e 35.

KITCHENHAM, B. A.; DYBA, T.; JORGENSEN, M. Evidence-based software engineering. In: *Proceedings of the 26th International Conference on Software Engineering*. USA: IEEE Computer Society, 2004. (ICSE '04), p. 273–281. ISBN 0769521630. Disponível em: <<https://doi.org/10.1109/ICSE.2004.1317449>>. Citado 2 vezes na(s) página(s) 13 e 35.

KRUPITZER, C. *et al.* A survey on engineering approaches for self-adaptive systems. *Pervasive and Mobile Computing*, v. 17, p. 184–206, 2015. ISSN 1574-1192. 10 years of Pervasive Computing' In Honor of Chatschik Bisdikian. Disponível em: <<https://doi.org/10.1016/j.pmcj.2014.09.009>>. Citado 3 vezes na(s) página(s) 12, 21 e 23.

LEMOES, R. de *et al.* Software engineering for self-adaptive systems: A second research roadmap. In: \_\_\_\_\_. *Software Engineering for Self-Adaptive Systems II: International Seminar, Dagstuhl Castle, Germany, October 24-29, 2010 Revised Selected and Invited Papers*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013. p. 1–32. Disponível em: <[https://doi.org/10.1007/978-3-642-35813-5\\_1](https://doi.org/10.1007/978-3-642-35813-5_1)>. Citado na página 11.

MAES, P. Concepts and experiments in computational reflection. In: . New York, NY, USA: Association for Computing Machinery, 1987. (OOPSLA '87), p. 147–155. ISBN 0897912470. Disponível em: <<https://doi.org/10.1145/38765.38821>>. Citado na página 25.

MARKS, R. M.; STERRITT, R. A metadata driven approach to performing complex heterogeneous database schema migrations. *Innovations in Systems and Software Engineering*, v. 9, n. 3, p. 179, 2013. ISSN 1614-5054. Disponível em: <<https://doi.org/10.1007/s11334-013-0217-8>>. Citado 3 vezes na(s) página(s) 41, 47 e 51.

MARTIN, R. C. *Arquitetura limpa: O Guia do Artesão para Estrutura e Design de Software*. [S.l.]: Alta Books, 2019. ISBN 978-85-5080-460-6. Citado na página 18.

MITRANONT, J. L.; FUGKEAW, S. Direct access versioning for multidimensional database schema creation. In: *The Sixth IEEE International Conference on Computer and Information Technology (CIT'06)*. [s.n.], 2006. p. 17–17. Disponível em: <<https://doi.org/10.1109/CIT.2006.79>>. Citado 4 vezes na(s) página(s) 43, 48, 51 e 52.

NAMDEO, B.; SUMAN, U. A model for relational to nosql database migration: Snapshot-live stream db migration model. In: *2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS)*. [s.n.], 2021. v. 1, p. 199–204. ISSN 2575-7288. Disponível em: <<https://doi.org/10.1109/ICACCS51430.2021.9441829>>. Citado 3 vezes na(s) página(s) 43, 47 e 52.

NEAMTIU, I. *et al.* Improving cloud availability with on-the-fly schema updates. In: *Proceedings of the 19th International Conference on Management of Data*. Mumbai, Maharashtra, IND: Computer Society of India, 2013. (COMAD '13), p. 24–34. Disponível em: <<https://dl.acm.org/doi/10.5555/2694476.2694487>>. Citado 2 vezes na(s) página(s) 43 e 48.

PALERMO, J. *The Onion Architecture*. 2008. Disponível em: <<https://jeffreypalermo.com/2008/07/the-onion-architecture-part-1>>. Acesso em: 26 out. 2022. Citado na página 18.

PETERSEN, K.; VAKKALANKA, S.; KUZNIARZ, L. Guidelines for conducting systematic mapping studies in software engineering: An update. *Information and Software Technology*, v. 64, p. 1–18, 2015. ISSN 0950-5849. Disponível em: <<https://doi.org/10.1016/j.infsof.2015.03.007>>. Citado 5 vezes na(s) página(s) 13, 35, 36, 133 e 150.

PRESSMAN, R. S.; MAXIM, B. R. *Engenharia de software: uma abordagem profissional*. 8. ed. [S.l.]: AMGH Editora Ltda., 2016. ISBN 978-85-8055-533-2. Citado 3 vezes na(s) página(s) 16, 17 e 18.

PUKALL, M. *et al.* Javadaptor – flexible runtime updates of java applications. *Software - Practice and Experience*, v. 43, n. 2, p. 153–185, 2013. Cited By 34. Disponível em: <<https://doi.org/10.1002/spe.2107>>. Citado 2 vezes na(s) página(s) 41 e 48.

RA, Y.-G.; RUNDENSTEINER, E. A. Oodb support for providing transparent schema changes. In: *Proceedings of the 1994 Conference of the Centre for Advanced Studies on Collaborative Research*. IBM Press, 1994. (CASCON '94), p. 57. Disponível em: <<https://dl.acm.org/doi/10.5555/782185.782242>>. Citado 3 vezes na(s) página(s) 43, 48 e 51.

RAHM, E.; BERNSTEIN, P. A. An online bibliography on schema evolution. *SIGMOD Rec.*, Association for Computing Machinery, New York, NY, USA, v. 35, n. 4, p. 30–31, dec 2006. ISSN 0163-5808. Disponível em: <<https://doi.org/10.1145/1228268.1228273>>. Citado na página 36.

SALEHIE, M.; TAHVILDARI, L. Self-adaptive software: Landscape and research challenges. *ACM Transactions on Autonomous and Adaptive Systems*, Association for Computing Machinery, New York, NY, USA, v. 4, n. 2, maio 2009. ISSN 1556-4665. Disponível em: <<https://doi.org/10.1145/1516533.1516538>>. Citado 9 vezes na(s) página(s) 11, 12, 21, 22, 23, 24, 25, 35 e 60.

SANTOS, R. J.; BERNARDINO, J.; VIEIRA, M. 24/7 real-time data warehousing: A tool for continuous actionable knowledge. In: *2011 IEEE 35th Annual Computer Software and Applications Conference*. [s.n.], 2011. p. 279–288. ISSN 0730-3157. Disponível em: <<https://doi.org/10.1109/COMPSAC.2011.44>>. Citado 2 vezes na(s) página(s) 42 e 47.

- SCHULER, R.; KESSELMAN, C. Chisel: a user-oriented framework for simplifying database evolution. *Distributed and Parallel Databases*, v. 39, n. 2, p. 483, 2021. ISSN 1573-7578. Disponível em: <<https://doi.org/10.1007/s10619-020-07314-x>>. Citado 4 vezes na(s) página(s) 46, 50, 51 e 53.
- SILBERSCHATZ, A.; KORTH, H. F.; SUDARSHAN, S. *Sistema de banco de dados*. 7. ed. [S.l.]: GEN LTC, 2021. ISBN 978-85-9515-733-0. Citado 2 vezes na(s) página(s) 29 e 31.
- SOMMERVILLE, I. *Engenharia de software*. 10. ed. [S.l.]: Pearson Education do Brasil Ltda., 2019. ISBN 978-85-430-2497-4. Citado 4 vezes na(s) página(s) 16, 17, 18 e 54.
- SONG, Y. *et al.* Automatic adaptation of software applications to database evolution by graph differencing and aop-based dynamic patching. In: *2012 IEEE 36th Annual Computer Software and Applications Conference*. [s.n.], 2012. p. 111–118. ISSN 0730-3157. Disponível em: <<https://doi.org/10.1109/COMPSAC.2012.21>>. Citado 2 vezes na(s) página(s) 42 e 48.
- SOUNDARARAJAN, G.; AMZA, C. Online data migration for autonomic provisioning of databases in dynamic content web servers. In: *Proceedings of the 2005 Conference of the Centre for Advanced Studies on Collaborative Research*. IBM Press, 2005. (CASCON '05), p. 268–282. Disponível em: <<https://dl.acm.org/doi/10.5555/1105634.1105654>>. Citado 2 vezes na(s) página(s) 43 e 52.
- STIEMER, A. *et al.* Polymigrate: Dynamic schema evolution and data migration in a distributed polystore. In: GADEPALLY, V. *et al.* (Ed.). *Heterogeneous Data Management, Polystores, and Analytics for Healthcare*. Cham: Springer International Publishing, 2021. p. 42–53. ISBN 978-3-030-71055-2. Disponível em: <[https://doi.org/10.1007/978-3-030-71055-2\\_4](https://doi.org/10.1007/978-3-030-71055-2_4)>. Citado 3 vezes na(s) página(s) 50, 51 e 52.
- STÖRL, U.; KLETTKE, M.; SCHERZINGER, S. Nosql schema evolution and data migration: State-of-the-art and opportunities. *Advances in Database Technology - EDBT*, v. 2020-March, p. 655–658, 2020. Disponível em: <<https://doi.org/10.5441/002/edbt.2020.87>>. Citado na página 12.
- VOGEL, T. *Software Engineering for Self-Adaptive Systems exemplars repository*. 2023. [Online]. Disponível em: <<http://self-adaptive.org/exemplars>>. Acesso em: 15 mar. 2024. Citado 2 vezes na(s) página(s) 12 e 134.
- WANG, Q.; DU, Z.; LIU, N. Design and realization of database online migration. In: *Proceedings of 2012 2nd International Conference on Computer Science and Network Technology*. [s.n.], 2012. p. 1195–1198. Disponível em: <<https://doi.org/10.1109/ICCSNT.2012.6526138>>. Citado na página 42.
- WEYNS, D. Software engineering of self-adaptive systems. In: \_\_\_\_\_. *Handbook of Software Engineering*. Cham: Springer International Publishing, 2019. p. 399–443. Disponível em: <[https://doi.org/10.1007/978-3-030-00262-6\\_11](https://doi.org/10.1007/978-3-030-00262-6_11)>. Citado na página 11.
- WEYNS, D. *et al.* On patterns for decentralized control in self-adaptive systems. In: \_\_\_\_\_. *Software Engineering for Self-Adaptive Systems II: International Seminar, Dagstuhl Castle, Germany, October 24-29, 2010 Revised Selected and Invited Papers*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013. p. 76–107. ISBN 978-3-642-35813-5. Disponível em: <[https://doi.org/10.1007/978-3-642-35813-5\\_4](https://doi.org/10.1007/978-3-642-35813-5_4)>. Citado 4 vezes na(s) página(s) 12, 21, 22 e 23.