

UNIVERSIDADE FEDERAL DO PAMPA

Juliana Mareco Medeiros

**Investigando o Estado da Arte na Evolução
de Bancos de Dados: Um Mapeamento
Sistemático e a Criação de um Guideline**

Alegrete
2022

Juliana Mareco Medeiros

Investigando o Estado da Arte na Evolução de Bancos de Dados: Um Mapeamento Sistemático e a Criação de um Guideline

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia de Software da Universidade Federal do Pampa como requisito parcial para a obtenção do título de Bacharel em Engenharia de Software.

Orientador: Prof. Dr. Fábio Paulo Basso

Coorientador: Prof. Dr. Elder de Macedo Rodrigues

Alegrete
2022

Juliana Mareco Medeiros

Investigando o Estado da Arte na Evolução de Bancos de Dados: Um Mapeamento Sistemático e a Criação de um Guideline

Trabalho de Conclusão de Curso apresentada ao Curso de Engenharia de Software da Universidade Federal do Pampa, como requisito parcial para obtenção do Título de Bacharel em Engenharia de Software.

Trabalho de Conclusão de Curso defendido e aprovado em 10 de março de 2022.

Banca examinadora:

Prof. Dr. Fábio Paulo Basso

Orientador

(UniPampa)

Prof. Dr. Elder de Macedo Rodrigues

(UniPampa)

Profa. Dra. Andrea Sadebra Bordin
(UFSC)

Prof. Dr. Maicon Bernardino da Silveira
(UniPampa)



Assinado eletronicamente por **Andréa Sabedra Bordin, Usuário Externo**, em 10/03/2022, às 18:30, conforme horário oficial de Brasília, de acordo com as normativas legais aplicáveis.



Assinado eletronicamente por **FABIO PAULO BASSO, PROFESSOR DO MAGISTERIO SUPERIOR**, em 14/03/2022, às 15:14, conforme horário oficial de Brasília, de acordo com as normativas legais aplicáveis.



Assinado eletronicamente por **ELDER DE MACEDO RODRIGUES, PROFESSOR DO MAGISTERIO SUPERIOR**, em 14/03/2022, às 15:20, conforme horário oficial de Brasília, de acordo com as normativas legais aplicáveis.



Assinado eletronicamente por **MAICON BERNARDINO DA SILVEIRA, PROFESSOR DO MAGISTERIO SUPERIOR**, em 14/03/2022, às 16:47, conforme horário oficial de Brasília, de acordo com as normativas legais aplicáveis.



A autenticidade deste documento pode ser conferida no site https://sei.unipampa.edu.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **0744719** e o código CRC **F4081DEB**.

Este trabalho é dedicado aos meus pais postiços, Suzana e Eduardo,
pois sem seu apoio, confiança e carinho nada seria possível.

AGRADECIMENTOS

Agradeço primeiramente a Deus, por me guiar em todo meu caminho. Também agradeço a meus familiares pelo amor, carinho e compreensão nos momentos mais difíceis. Em especial, aos meus pais por estarem sempre ao meu lado e por me amarem tanto e permitirem que eu voasse a fim de procurar um futuro melhor e aos meus pais postigos Suzana e Eduardo por tudo que fizeram para mim em minha vida, sem eles eu não seria o que sou hoje.

Agradeço aos meus professores pelo conhecimento compartilhado durante todo o curso, principalmente ao meu orientador Fábio Paulo Basso que embarcou nesse trabalho com muito entusiasmo mesmo sabendo de suas dificuldades.

“Se você não pode voar, então corra, se você não pode correr, ande, se você não pode andar, engatinhe, mas tudo que você tem que fazer é continuar seguindo em frente”
(Martin Luther King Jr.)

RESUMO

Dentro dos cenários atuais de desenvolvimento existe a necessidade de constante evolução de um *software*. Com o objetivo de facilitar esta atividade foram propostas soluções como integração contínua, implantação contínua, dentre outras. Decorrente dessa necessidade de se realizar a evolução de um *software*, torna-se necessário evoluir as tecnologias utilizadas em seu desenvolvimento. Dentre estas tecnologias uma crucial para manter seu funcionamento é o banco de dados. Os bancos de dados constituem hoje uma parte essencial de toda empresa, não só para armazenar tipos de informações que são comuns à maioria das organizações, mas também para informações específicas do seu negócio. Sua evolução é resultado de qualquer alteração no banco, que tenha o objetivo de resolver um problema encontrado ou adicionar dados de uma nova funcionalidade. Estas alterações são impostas por um banco de dados limitado ou pela evolução de um *software* ao longo das iterações de um processo de desenvolvimento. No entanto, a evolução de banco de dados é uma das atividades da Engenharia de Sistemas que não é claramente caracterizada na literatura, se tornando uma atividade complexa e desgastante. Sabendo dessa necessidade, foi realizado um mapeamento sistemático com o objetivo de encontrar as soluções propostas para esta área até o momento. Assim, tornou-se possível designar quais são as principais práticas, técnicas, tecnologias, ferramentas, *guidelines*, processos ou abordagens utilizados, juntamente com quais os domínios de aplicação e quais são as vantagens e desvantagens de cada técnica de evolução de um banco de dados. Isto foi realizado com o intuito de possibilitar a criação de *guidelines* voltados para todos os modelos de desenvolvimento, desde os tradicionais até os ágeis juntamente com o uso de integração contínua. Aumentando assim a eficácia e eficiência da evolução do banco de dados em diversos contextos.

Palavras-chave: Evolução de Banco de Dados, Evolução, Banco de Dados, Evolução de Esquema de Banco de Dados, Esquema de Banco de Dados, Reengenharia, Refatoração, Reengenharia de Banco de Dados, Refatoração de Banco de Dados, Reengenharia de Esquema de Banco de Dados, Refatoração de Esquema de Banco de Dados.

ABSTRACT

Within the current development scenarios, there is a need for the constant evolution of software. In order to facilitate this activity, solutions were proposed such as continuous integration, continuous deployment, among others. Due to this need to carry out the evolution of software, it becomes necessary to evolve the technologies used in its development. Among these technologies, one crucial to maintaining its functioning is the database. Databases are today an essential part of every business, not only to store types of information that are common to most organizations but also to specific information for your business. Its evolution is the result of any change in the bank that has the purpose of solving a problem encountered or adding data of new functionality. These changes are imposed by a limited database or the evolution of software over the iterations of a development process. However, the evolution of the database is one of the activities of Systems Engineering that is not clearly characterized in the literature, becoming a complex and exhausting activity. Knowing this need, a systematic mapping was carried out in order to find the solutions proposed for this area so far. Thus, it became possible to designate which are the main practices, techniques, technologies, tools, guidelines, processes or approaches used, together with what application domains and what are the advantages and disadvantages of each technique of evolution of a database. This was done with the aim of creating guidelines for all development models, from traditional to agile along with the use of continuous integration. Thus increasing the effectiveness and efficiency of the database evolution in various contexts.

Key-words: Database Evolution, Database, Evolution, Scheme, Database Schema Evolution, Reengineering, Refactoring, Database Reengineering, Database Refactoring, Database Schema Reengineering, Database Schema Refactoring.

LISTA DE FIGURAS

Figura 1 – Processo proposto por (PETERSEN et al., 2008) e adaptado pelo autor	33
Figura 2 – String de Busca	37
Figura 3 – Comparação de Tipos de Bancos de Dados por meio do Google Trends	38
Figura 4 – Diagrama em Camadas dos Estudos Seleccionados	44
Figura 5 – Quantidade de estudos por ano	48
Figura 6 – Métodos de realizar e abordar evolução de banco e suas categorias . . .	57
Figura 7 – Problema Motivante	63
Figura 8 – Sistema Implantado	64

LISTA DE TABELAS

Tabela 1 – Palavras-Chave e Sinônimos	36
Tabela 2 – Strings de Busca Utilizadas em Cada Base	41
Tabela 3 – Estudos Importados	43
Tabela 4 – Aplicação dos Critérios de Qualidade	45
Tabela 4 – Aplicação dos Critérios de Qualidade	46
Tabela 4 – Aplicação dos Critérios de Qualidade	47
Tabela 5 – Detalhes das realizações de evolução de banco de dados	50
Tabela 5 – Detalhes das realizações de evolução de banco de dados	51
Tabela 5 – Detalhes das realizações de evolução de banco de dados	52
Tabela 6 – Detalhes dos apoios à evolução de banco de dados	53
Tabela 6 – Detalhes dos apoios à evolução de banco de dados	54
Tabela 6 – Detalhes dos apoios à evolução de banco de dados	55
Tabela 6 – Detalhes dos apoios à evolução de banco de dados	56
Tabela 7 – Aplicação dos Critérios de Qualidade	89
Tabela 7 – Aplicação dos Critérios de Qualidade	90
Tabela 7 – Aplicação dos Critérios de Qualidade	91
Tabela 7 – Aplicação dos Critérios de Qualidade	92
Tabela 7 – Aplicação dos Critérios de Qualidade	93
Tabela 8 – Detalhes das realizações de evolução de banco de dados	95
Tabela 8 – Detalhes das realizações de evolução de banco de dados	96
Tabela 8 – Detalhes das realizações de evolução de banco de dados	97
Tabela 9 – Detalhes dos apoios à evolução de banco de dados	100
Tabela 9 – Detalhes dos apoios à evolução de banco de dados	101
Tabela 9 – Detalhes dos apoios à evolução de banco de dados	102

LISTA DE SIGLAS

- API** Interface de Programação de Aplicativos
- BDOO** Banco de Dados Orientado a Objeto
- CE** Critério de Exclusão
- CI** Critério de Inclusão
- CQ** Critério de Qualidade
- DC** *Deploy* Contínuo
- DSS** Sistema de Suporte a Decisão
- EC** Entrega Contínua
- ER** Entidade Relacionamento
- ETL** *Extract Transform Load*
- IC** Integração Contínua
- IDE** Ambiente Integral de Desenvolvimento
- MDD** *Model-Driven Engineering*
- NoSQL** Não Somente SQL
- QP** Questão de Pesquisa
- RP** Resolução de Problemas
- SGBD** Sistema de Gerenciamento de Banco de Dados
- SMO** *Schema Modification Operators*
- SMS** *Systematic Mapping Study*
- SPL** Linha de Produto de *Software*
- SQL** Linguagem de Consulta Estruturada
- TCC** Trabalho de Conclusão de Curso

SUMÁRIO

1	INTRODUÇÃO	23
1.1	Motivação	24
1.2	Objetivos	24
1.3	Metodologia	25
1.4	Organização	26
2	REFERENCIAL TEÓRICO	29
2.1	Banco de Dados	29
2.1.1	Banco de Dados Relacionais	30
2.1.2	Banco de Dados Não-Relacional	30
2.1.3	Banco de Dados Orientado a Objetos	31
2.1.4	Banco de Dados Distribuídos	31
2.2	Evolução no Contexto de <i>Software</i>	32
2.3	Estudo de Mapeamento Sistemático	33
3	REVISÃO DE LITERATURA	35
3.1	Planejamento	35
3.1.1	Protocolo	35
3.1.1.1	Escopo e Objetivos	35
3.1.1.2	Estrutura das Questões	35
3.1.1.3	Questões de Pesquisa	35
3.1.1.4	Estratégia de Pesquisa	36
3.1.1.5	Critérios de Inclusão e Exclusão	37
3.1.2	Critérios de Qualidade	37
3.1.2.1	Estratégia de Seleção	39
3.1.2.2	Formulário de Extração de Dados	40
3.1.2.3	Estratégia de Análise	40
3.2	Condução	41
3.2.1	Strings de Busca Específicas	41
3.2.2	Busca nas Bases de Dados	43
3.2.3	Seleção dos Estudos	43
3.2.4	Aplicação de Critérios de Qualidade	44
3.3	Análise	48
3.4	Discussão	60
3.4.1	Resultados	60
3.4.2	Ameaças à Validade	60
4	GUIDELINE	63
4.1	Problema Introduzido	63

4.2	<i>Guideline</i>	64
4.3	Resultado da Aplicação	65
5	CONSIDERAÇÕES FINAIS	67
5.1	Trabalhos Futuros	67
	REFERÊNCIAS	69
	 APÊNDICES	 83
	APÊNDICE A – GUIDELINE PARA EVOLUÇÃO DE BANCO DE DADOS	85
A.1	INTRODUÇÃO	85
A.2	Problema Motivante e Requisitos para o Melhor Uso do Guideline	86
A.3	GUIDELINE	86
A.3.1	Identificar o Tipo de SGBD Adotado	94
A.3.2	Identificar Problemas Específicos de Evolução de Banco de Dados	94
A.3.3	Adaptar O Processo Base para Problemas Específicos de Evolução de Banco de Dados	95
A.3.4	Identificar Possíveis Vantagens da Evolução de Banco de Dados	97
A.3.5	Identificar Possíveis Desvantagens na Evolução de Banco de Dados	98
A.3.6	Identificar Possíveis Inovações para Introduzir na Organização	99
A.4	CONSIDERAÇÕES FINAIS	103
A.4.1	Práticas Mínimas Esperadas para a Gerência de Configuração para um Marco de Cinco Semanas	103
A.4.2	Como Iniciar?	104
	Índice	107

1 INTRODUÇÃO

Com o avanço tecnológico, a utilização de banco de dados se tornou constante no dia a dia da sociedade, provocando grande impacto no desenvolvimento de *software* devido ao armazenamento e acesso de dados de forma rápida, permitindo a automatização de processos e tornando mais acessível os dados de forma virtualizada. Nossas vidas passaram a estar contidas em banco de dados e se tornaram informações valiosas para empresas, principalmente pela necessidade de destaque no mercado competitivo (IMPACTA, 2017). Existem vários sistemas cotidianos que precisam usar banco de dados para realizar suas funções, como é o caso dos que realizam operações bancárias, reservas em hotéis, compras de produtos em lojas físicas ou *onlines*, gerenciamento de estoques em supermercados, entre outros.

Os sistemas são feitos para refletir o comportamento do mundo real sendo necessário que o *software* acompanhe as mudanças impostas pelo meio em que está inserido (GALL et al., 1997). Se o sistema não sofre essas mudanças, pode ficar obsoleto e cair em desuso. Uma das maneiras de mitigar esse problema e modificar um *software* é por meio da evolução. A evolução de *software* compreende qualquer mudança que ocorre com um sistema a fim de deixá-lo completo e, se possível, livre de *bugs* tendo como base o *feedback* dos usuários. Consequentemente, com a evolução de um *software* há a necessidade de evoluir as tecnologias que são utilizadas pelo mesmo, como é o caso do banco de dados.

Sabe-se que a evolução de banco de dados consiste em qualquer alteração no banco, que tenha o objetivo de resolver algum problema imposto por um banco limitado ou pela evolução de um *software* ao longo das iterações de um processo de desenvolvimento. No entanto, a evolução de banco de dados é uma das atividades da Engenharia de Sistemas que não é claramente caracterizada na literatura. Primeiramente, falta uma boa caracterização para poder resultar em *guidelines* e/ou metodologias e ferramentas que levem a ótimos desempenhos dos times de desenvolvimento de *software* nas tarefas de evolução de bancos de dados.

Além disso, devido à necessidade da implantação da Entrega Contínua (EC) nas fábricas de *software* por meio de *DevOps*, os processos precisam ser cada vez mais automatizados. Isso impõe um novo desafio a equipe de desenvolvimento: a necessidade de automação no processo de desenvolvimento de atividades de evolução de banco de dados por meio de ferramentas para Integração Contínua (IC) e Implantação/*Deploy* Contínuo (DC) (LORENZ; HESSE; RUDOLPH, 2016). Tal necessidade motiva as equipes pela busca da perfeição nas tarefas de gerência de configuração. Estas, por sua vez, são afetadas diretamente por outras atividades de evolução do sistema, incluindo o banco de dados. Logo, analisando o contexto prático da indústria na automação dos processos de desenvolvimento (REDGATE, 2018), fica uma dúvida: o que é possível automatizar nas atividades de evolução de banco de dados de modo a agregar agilidade em processos de IC, DC e EC?

A fim de elencar quais são as principais práticas utilizadas tanto na área acadêmica quanto na área industrial, pretende-se aprofundar a literatura da área (LAUKKANEN; ITKONEN; LASSENIUS, 2017). Para isso, busca-se, em especial, os *guidelines*, metodologias e suporte ferramental para auxiliar na evolução de bancos de dados, bem como atributos de qualidade de propostas para evolução que direcionem a fatores de sucesso nas práticas de IC, DC e EC.

1.1 Motivação

Em cenários modernos de desenvolvimento de *software*, estabelecer um procedimento para a evolução de banco de dados é um objetivo frequente de times de desenvolvimento de *software*. Uma vez que o gerenciamento de mudanças em bancos de dados não têm um processo definido, fazendo com que seja preciso muito trabalho de análise manual (FIGUEIREDO; MARTINS, 2018), é preciso caracterizar estas atividades. Sabe-se que o sucesso na sincronização das atividades de desenvolvimento na indústria depende do conhecimento tácito do time em relação às ferramentas e boas práticas para a evolução do banco. Esta troca de informações é importante para as equipes de desenvolvimento escolherem uma estratégia de evolução. No entanto, a falta de sistematização pode levar à uma análise limitada, gerando incertezas quanto ao seu impacto na integração da evolução com a automação de processos em níveis mais altos da gerência de configuração.

Nesse sentido, a literatura carece de uma investigação aprofundada sobre o tema. A fim de verificar as estratégias utilizadas para evoluir um banco e seu impacto em processos automatizados, principalmente pela integração contínua, decidiu-se realizar um mapeamento sistemático dessas estratégias. O objetivo é designar quais são as principais práticas, técnicas, tecnologias, ferramentas, *guidelines*, processos ou abordagens utilizadas. Além disso, este estudo buscou elencar quais os domínios de aplicação que demandam da execução de técnicas de evolução de bancos de dados, as vantagens e desvantagem destas técnicas, e atributos de qualidade das propostas da área que resultem em processos otimizados para a entrega contínua.

1.2 Objetivos

O objetivo geral é "Investigar o tema de evolução de modelos físicos de banco de dados e propor um *guideline* para realizar a evolução de banco de dados. Assim, para o desenvolvimento completo dos trabalhos de conclusão de curso I e II, está prevista a execução de três atividades que caracterizam os objetivos específicos.

Objetivo Específico 1: Mapear as propostas da literatura da área. Assim, para o Trabalho de Conclusão de Curso (TCC) I, foi executada e relatada parcialmente a primeira atividade: Mapeamento sistemático sobre abordagens para evolução de banco

de dados em geral. Portanto, as outras duas são planejadas para execução ao longo do TCC II.

Objetivo Específico 2: Realizar a segunda parte do mapeamento sistemático. Visto que é um mapeamento de mais de quatro mil estudos, será necessário realizar a segunda parte durante TCC II.

Objetivo Específico 3: Criação de um *guideline* de evolução de banco de dados, ou seja, um passo a passo para orientar como deve ser realizada a evolução do banco de dados.

1.3 Metodologia

Este trabalho foca na realização de um mapeamento sistemático e criação de um *guideline* para evolução de banco de dados. O mapeamento é uma forma de identificar, avaliar e interpretar todas as pesquisas relevantes para alguma questão de pesquisa em particular. O mapeamento também busca evidenciar o que existe relacionado a um tema e, geralmente, é executado por mais de um revisor. Neste sentido, contou-se com a colaboração do aluno Yury Alencar. Já o *guideline* tem o objetivo de orientar a como realizar a evolução de banco de dados.

A seguir, descreve-se todos os passos deste trabalho de conclusão.

Atividade 1: Caracterizar o problema. Esta atividade foi realizada seguindo os seguintes passos: **Atividade 1.1)** Conversa com o professor da disciplina de Resolução de Problemas (RP) V de 2018, orientador desta pesquisa, para a definição do tema central de investigação; **Atividade 1.2)** Conversa informal com os colegas da disciplina para identificar dificuldades encontradas para a evolução de bancos de dados, de modo a caracterizar o problema de investigação; **Atividade 1.3)** Leitura de artigo sobre o estado da arte na entrega contínua com DevOps (LAUKKANEN; ITKONEN; LASSENIUS, 2017), em busca de metodologias, ferramentas e lições aprendidas sobre a evolução de banco de dados em pipelines de integração contínua, e; **Atividade 1.4)** Leitura de *survey* de 2018 realizado na indústria sobre práticas de entrega contínua (REDGATE, 2018).

Atividade 2: Início de revisão sistemática de literatura em abordagens para evolução de banco de dados em pipelines de Integração Contínua (IC). Uma vez que se identificou uma lacuna de pesquisa no tema, planejou-se um estudo de revisão de literatura bem focado em práticas de evolução de banco de dados aplicadas em processos automatizados por servidores de IC. Este estudo contou com os passos iniciais de uma revisão sistemática incluindo: **Atividade 2.1)** Definição das Questões de Pesquisa; **Atividade 2.2)** Triagem de Artigos a partir de Título e *Abstract*; **Atividade 2.3)** Triagem de Artigos por Critérios de Inclusão e Exclusão. Uma vez que se detectou um conjunto dez trabalhos selecionados, chegou-se à conclusão de que os critérios estavam muito restritivos, perdendo uma série de trabalhos que pareciam muito interessantes. Assim, para garantir a qualidade e quantidade dos estudos, suspendeu-se o protocolo atual de pesquisa

e optou-se por refazer um novo para um mapeamento sistemático.

Atividade 3: Mapeamento sistemático sobre abordagens para evolução de banco de dados em geral. **Atividade 3.1)** Planejamento do *Systematic Mapping Study* (SMS). Para a realização deste mapeamento é definido um protocolo que têm como objetivo especificar os métodos que serão usados para realizar o mapeamento. Tal protocolo baseia-se no protocolo de Petersen (PETERSEN et al., 2008) e foi adaptado para suprir as necessidades do mapeamento. Para descrever o objetivo e melhorar a maneira como irá ser elencada as questões de pesquisa, foi utilizado o PICO, uma variação do PICOC, que busca elencar qual a população, a intervenção, o contexto e resultado esperado do mapeamento (KITCHENHAM; CHARTERS, 2007). Além disso, foi utilizado o *Google Trends*¹ para elencar os principais bancos de dados que estão em alta para realizar a seleção dos estudos. Os resultados deste mapeamento pretendem especificar os problemas relacionados a evolução de *software* devido. Logo, pretende-se listar as principais práticas. **Atividade 3.2)** Execução do SMS, sendo necessária a realização dos seguintes passos da metodologia adotada: Definição das Questões de Pesquisa, Triagem de Artigos a partir de Título e *Abstract*, Triagem de Artigos por Critérios de Inclusão e Exclusão, Extração de Dados, e Análise dos Dados Extraídos. **Atividade 3.3)** Relatar o SMS, onde os estudos são tabulados em um *framework* de comparação e as questões de pesquisa são respondidas

Atividade 4: Escrita do TCC I, que além do relato do SMS também conta com as seções de metodologia, embasamento teórico e atividades futuras previstas em cronograma.

Atividade 5: Defesa do TCC I, em que foi necessário elaborar uma apresentação.

Atividade 6: Finalização do SMS, nova comparação dos estudos tabulados e das respostas obtidas nas questões de pesquisa.

Atividade 7: Criação de um *Guideline* de Validação de Banco de Dados e aplicação com os alunos da disciplina de Resolução de Problemas V.

Atividade 8: Defesa do TCC II.

1.4 Organização

A sequência deste trabalho está organizada e estruturada da seguinte maneira:

- O Capítulo 2 aborda a base teórica para este trabalho, apresentando os conceitos de banco de dados, evolução de *software*, protocolo de Petersen e *Guideline*.
- O Capítulo 3 trata de como foi realizado o planejamento, condução, análise e a discussão sobre o mapeamento sistemático.
- O Capítulo 4 apresenta o *guideline* criado bem como a aplicação na disciplina de Resolução de Problemas V e os problemas encontrados.

¹ Google Trends: <https://trends.google.com.br/trends/?geo=BR>

- O Capítulo 5 traz as considerações finais.

2 REFERENCIAL TEÓRICO

Neste capítulo são apresentados os principais conceitos relacionados ao tema deste trabalho, os quais serão utilizados como base para o desenvolvimento do estudo proposto: banco de dados, evolução de *software* e estudo de mapeamento sistemático.

2.1 Banco de Dados

Antigamente todas as informações das empresas eram armazenadas utilizando fichas de papel, que eram organizadas por meio de pastas. Extrair qualquer informação ou manter os arquivos organizados era uma tarefa muito trabalhosa, além do acesso à informação depender da localização geográfica. Decorrente da necessidade de facilitar o acesso e procura a esses dados, foram desenvolvidos os bancos de dados. Outro motivo foi para que estas informações fossem evoluídas para o meio digital. Com a revolução da internet, na década de 1990, aumentou-se o acesso direto aos bancos de dados. A utilização de banco de dados cresceu em todas as empresas, e as pessoas passaram a interagir indiretamente com os bancos de dados sem perceber. Então, o acesso a bases de dados através de formulários e páginas constituiu uma parte essencial na vida de todos (SILBERSCHATZ; SUNDARSHAN; KORTH, 2016).

Há muitas definições para bancos de dados. Cada autor descreve de uma maneira: Segundo (CERÍCOLA, 1995), o banco de dados é uma coleção de dados, organizados e integrados, que formam uma representação natural de dados, sem restrições ou modificações para todas as aplicações relevantes, sem haver redundância de dados. De acordo com (DATE, 2004), um banco de dados é uma coleção de dados persistentes utilizados pelos sistemas de aplicação de uma determinada empresa. E, conforme (SILBERSCHATZ; SUNDARSHAN; KORTH, 2016), um banco de dados é uma coleção de dados inter-relacionados, representando informações sobre um domínio específico. Em outras palavras, banco de dados são dados armazenados, ou seja, são dados que se relacionam e são organizados e armazenados de forma que possam ser manipulados facilmente, podendo ser inseridos, alterados, excluídos e consultados.

Os bancos de dados constituem hoje uma parte essencial de toda empresa, não só para armazenar tipos de informações que são comuns à maioria das empresas, mas também a informação que é específica para a categoria da empresa. São amplamente utilizados em aplicações, tais como, informações empresariais, bancos e finanças, universidades, companhias aéreas, telecomunicações, entre outras aplicações (SILBERSCHATZ; SUNDARSHAN; KORTH, 2016). Um exemplo cotidiano é os sistemas de vendas como, por exemplo, de lojas, onde se têm as informações sobre os clientes, os produtos e as compras realizadas.

As vantagens da utilização do banco de dados não se restringem a apenas a facilidade de armazenamentos e operações de dados. Segundo (DATE, 2004) seu uso pode ser vantajoso tanto para compartilhar dados de aplicações existentes, quanto para novas

aplicações sem ter de acrescentar novos dados e seu uso ajuda a reduzir a redundância que ocorre quando se têm duas aplicações que possuem os mesmos dados. Além disso, dá suporte a transações como, por exemplo, nas operações bancárias de transferência que é necessário tirar dinheiro de uma conta e passar para outra, se houver declaração que as atualizações fazem parte da mesma transferência, o sistema pode garantir que ambas sejam realizadas ou nenhuma será. Outra vantagem é o reforço da segurança dos dados devido a ter um acesso único por canais apropriados, entre outras vantagens.

Existem diversos tipos de bancos de dados, alguns exemplos são os bancos de dados relacionais, bancos de dados não relacionais, bancos de dados orientados a objetos, bancos de dados distribuídos, entre outros.

2.1.1 Banco de Dados Relacionais

O banco de dados relacional pode ser definido como sendo uma maneira de armazenar, recuperar e manipular dados estruturais na forma de tabelas (CERICOLA, 1991). Ele possui a capacidade de lidar com grandes quantidades de informações, eliminando a redundância dos dados. Neste tipo de banco existe a possibilidade de elaborar um relacionamento lógico entre as informações referentes a espécie e ao indivíduo, evitando a necessidade de repetir informações e facilitando as consultas feitas em duas fontes de dados (KAUFELD; LUDERMIR, 1996). Na prática, a utilização de bancos de dados relacionais podem oferecer benefícios. Segundo (DALCIN, 1994) tais benefícios são a simplicidade e uniformidade, independência dos dados físicos, interfaces de alto nível para usuários finais, visões múltiplas dos dados, melhoria na segurança dos dados, redução significativa do tempo gasto na manutenção da base de dados e possibilidade de expansão devido à flexibilidade do sistema.

2.1.2 Banco de Dados Não-Relacional

O banco de dados não relacional ou, como é popularmente conhecido, Não Somente SQL (NoSQL), é uma abordagem não estruturada que preza pelo desempenho, escalabilidade e disponibilidade. O NoSQL possui características importantes distintas dos bancos de dados relacionais. Dessa forma, atende os requisitos de gerência de volume de dados de grande porte, semi-estruturado ou não estruturado sem possuir uma linguagem nativa, como a Linguagem de Consulta Estruturada (SQL). Algumas características que distinguem os NoSQL de um banco de dados relacional são: escalabilidade horizontal, ausência de esquema ou esquema flexível, suporte nativo a replicação, Interface de Programação de Aplicativos (API) simples para acessar aos dados e eventual consistência (LÓSCIO; OLIVEIRA; PONTES, 2011).

Os bancos de dados NoSQL têm uma grande variedade de formas e funcionalidades, sendo a única característica comum é o fato de não serem relacionais (FILHO, 2015). Apesar disso, a indústria categoriza os bancos de dados NoSQL em pequenos conjuntos

de áreas funcionais (SHARP et al., 2013). Algumas dessas categorias são: banco de dados orientado a chave/valor, banco de dados orientado a documentos, banco de dados orientado a grafos e banco de dados orientado a família de colunas. Conforme (SHARP et al., 2013):

- Banco de dados orientado a chave/valor: É uma abordagem implementa mecanismo mais simples de armazenamento NoSQL. É considerada uma grande tabela *hash*, ou seja, uma estrutura de dados simples que pode suportar um conjunto de pares chave/valor.
- Banco de dados orientado a documentos: É similar, em seu conceito, ao orientado à chave/valor, com exceção de que os dados são armazenados em documentos. Um documento é uma coleção de campos nomeados e valores associados.
- Banco de dados orientado a grafos: Armazena entidades, mas seu foco principal é nos relacionamentos que essas entidades têm umas com as outras. Armazena dois tipos de informação nós, que são instâncias de entidades, e arestas, que especificam a relação entre os nós.
- Banco de dados orientada à família de colunas: Organiza os seus dados dentro de linhas e colunas e pode parecer muito similar a um banco de dados relacional, pelo menos de maneira conceitual.

2.1.3 Banco de Dados Orientado a Objetos

Os Banco de Dados Orientado a Objetos (BDOOs) possuem características distintas dos bancos de dados relacionais porém, servem para o mesmo propósito, eles persistem dados para manutenção do negócio para o qual são aplicados. Isso faz com que seja possível recuperar, comparar e tratar esses dados a fim de obter um resultado concreto. Os BDOOs possuem três pilares principais: herança, polimorfismo e encapsulamento. Eles podem criar objetos que sejam persistentes e compartilhados entre diferentes aplicativos (BOSCARIOLI et al., 2006). Os BDOOs possuem maior flexibilidade durante manipulação de conteúdo e manipula os dados de forma constante através de identificadores de objetos. Dessa maneira, eles são mais utilizados em mercados verticais, como telecomunicações, finanças e saúde. Além disso, seu uso como repositório de dados multimídia ou de outros tipos de dados complexos, tem potencial para elevar o nível de utilização dessa tecnologia (BOSCARIOLI et al., 2006).

2.1.4 Banco de Dados Distribuídos

Um banco de dado distribuído é considerado uma coleção de dados que pertencem logicamente ao mesmo sistema, mas são distribuídos sobre várias bandas de uma rede

de computadores (BOBROWSKI, 1996). Conforme Date (DATE, 2004), estes sistemas consistem em uma coleção de bandas que se conectam por meio de redes de computadores, de forma que cada banda é um sistema de banco de dados em seu próprio direito, mas essas bandas cooperam umas com as outras, de tal forma que um usuário em qualquer banda pode acessar qualquer dado na rede exatamente como se estivesse armazenado em sua própria banda. O banco de dados distribuído possui algumas funções como, por exemplo, o gerenciamento de dados distribuídos com níveis diferentes de transparência, melhoria da confiabilidade e disponibilidade, melhoria do desempenho e facilidade de expansão.

2.2 Evolução no Contexto de *Software*

Hoje em dia, a evolução de *software* tem se tornado inevitável devido ao processo incremental de desenvolvimento de sistemas complexos compostos de múltiplos ciclos de retroalimentação (LEHMAN, 1996). Nesse sentido, depois que os sistemas são implantados, eles devem mudar para permanecerem úteis a fim de satisfazer seus usuários. Mudanças nos negócios frequentemente geram novos requisitos para o *software* sendo necessário que haja modificações para corrigir erros descobertos, para adaptá-los a uma nova plataforma, ou, até mesmo, para aprimorar seu desempenho e outras características não funcionais (SOMMERVILLE, 2007). Essas necessidades fizeram com que a evolução se tornasse um campo de pesquisa muito ativo e respeitado em engenharia de *software* (MENS, 2008).

A evolução de *software* diz respeito ao que acontece com um sistema ao longo do tempo (GODFREY; GERMAN, 2008). Ela lida com o processo pelo qual os *softwares* são modificados e adaptados ao ambiente em mudança (HERRAIZ et al., 2013). De acordo com (LEITE, 2001), os processos de desenvolvimento de *software* estão cada vez mais baseados nos conceitos relacionados à evolução de *software*, ou seja, geralmente são realizadas modificações em produtos já existentes. Juntamente com as técnicas de evolução é possível observar como o *software* está se comportando com as modificações e assim tentar simplificar as futuras atualizações. Entretanto, o processo de evolução pode variar consideravelmente de acordo do *software* a ser mantido, dos processos de desenvolvimento utilizados e das pessoas envolvidas. Em algumas organizações, a evolução pode ser um processo informal no qual as solicitações de mudanças, na maioria das vezes provêm de conversas entre os usuários e desenvolvedores. Em outras empresas, é um processo formalizado, com documentos estruturados e produzidos em cada estágio do processo de desenvolvimento do *software* (SOMMERVILLE, 2007)

Os *softwares*, assim como as pessoas, envelhecem. O envelhecimento não é possível de ser interrompido, porém, é possível entender o que está o causando e evitar seus efeitos, revertendo os dados por um tempo (ARAÚJO; TRAVASSOS, 2004). Isso é algo essencial, visto que, atualmente, as organizações são muito dependentes dos sistemas nos quais investem, já que na maioria das vezes estes sistemas realizam todo o gerenciamento

da empresa. O que os torna importantes ativos de negócios, sendo preciso investir em mudanças para que os valores sejam mantidos, ou seja, para tornar cada vez mais longa a vida do *software* sem perder qualidade e suas funcionalidades (SOMMERVILLE, 2007). Além disso, a evolução de *software* pode acarretar outras vantagens como, por exemplo, a projeção de sistemas mais flexíveis, planejamento das manutenções e controle de uma melhor forma de desenvolvimento de *software* (SOMMERVILLE, 2007).

2.3 Estudo de Mapeamento Sistemático

O SMS é uma metodologia que é amplamente utilizada dentro da medicina, que pode ser adotada também dentro da Engenharia de *Software*. Tendo como finalidade a apresentação de todos os estudos publicados em forma de relatório, os categorizando de acordo com sua ocorrência e fornecendo um resumo visual, ou seja, um mapa de resultados (PETERSEN et al., 2008). Com o objetivo de encontrar estes dados existem processos que podem ser seguidos dentro da engenharia de *software* como o proposto por (PETERSEN et al., 2008).

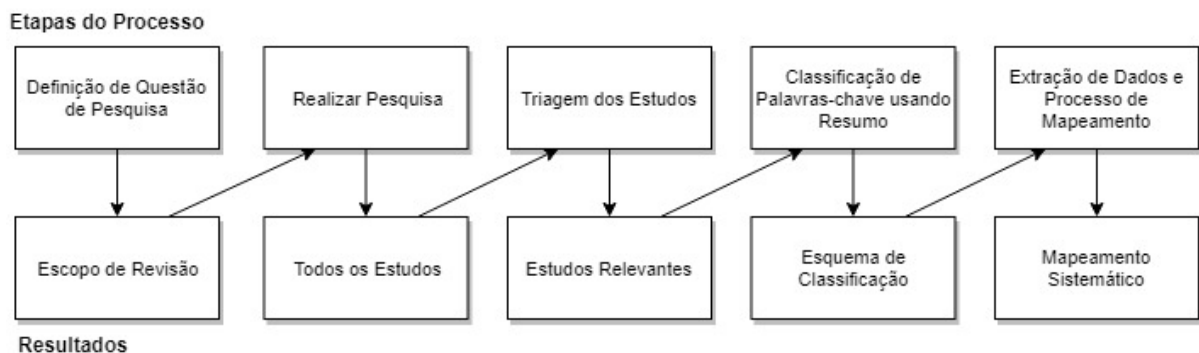


Figura 1 – Processo proposto por (PETERSEN et al., 2008) e adaptado pelo autor

O principal objetivo do SMS é fornecer uma visão geral da área, o processo proposto por Petersen apresenta cinco fases essenciais que são apresentadas na Figura 1. A primeira fase é a definição das questões de pesquisa, que tem como finalidade a exposição dos assuntos que se deseja abordar, assim, resultando em um escopo do mapeamento ou revisão. A segunda fase é realizar a pesquisa por estudos primários. Tais estudos são encontrados utilizando *strings* de pesquisa, através de bases de dados científicas. Uma maneira de criar a *string* de busca é estruturá-la em termos, ou palavras-chave relacionadas à população, intervenção, contexto e resultado (KITCHENHAM; CHARTERS, 2007). Como resultado desta etapa são encontrados diversos estudos com base nas palavras-chave empregadas para a criação da *string* de busca.

A terceira fase é a triagem dos estudos por meio da aplicação de critérios definidos anteriormente com o objetivo de incluir ou excluir estudos dentre os encontrados na fase anterior, pesquisados utilizando apenas palavras-chave. Esta etapa define quais estudos

são realmente relevantes para que se possa realizar uma extração de dados somente dos estudos que ajudem com o foco do mapeamento. A quarta etapa é relacionada a classificação de palavras-chave realizada por meio do resumo para que assim, se possa extrair as categorias dos artigos, estas categorias são atualizadas de acordo com leitura dos resumos. Assim, ao final desta fase se obtém uma lista com descrição e nome de todas as categorias existentes entre os estudos selecionados, denominada de esquema de classificação.

A última fase do mapeamento proposto por Petersen é a extração de dados e processo de mapeamento que tem como finalidade realizar a classificação dos estudos relevantes com base nas categorias encontradas na etapa anterior de acordo com a ocorrência. Além disto, toda a extração de dados também é realizada nesta etapa. A análise dos resultados do mapeamento tem como foco apresentar as frequências das publicações para cada categoria. Com isto é possível ver quais categorias foram enfatizadas por trabalhos anteriores e, assim, identificar lacunas e possibilidades de pesquisas futuras.

3 REVISÃO DE LITERATURA

3.1 Planejamento

Um mapeamento sistemático tem como objetivo conferir os estudos primários sobre um determinado tópico de pesquisa. Para realizar este mapeamento será utilizada como base a abordagem criada por (PETERSEN et al., 2008), explicada anteriormente na Subseção 2.3. No entanto, com o intuito de aumentar a qualidade do estudo, e encaixá-lo no objetivo geral do trabalho, algumas subetapas foram modificadas.

Na abordagem de Petersen, o mapeamento é separado em três fases: planejamento, condução e análise. Nesta seção será apresentado a primeira fase do mapeamento, o planejamento. O planejamento tem como objetivo a identificação da real necessidade do estudo, ou seja, a motivação para executar um mapeamento sistemático. Para que isso seja atingido, é necessário a definição de um protocolo.

3.1.1 Protocolo

O protocolo é um elemento importante do mapeamento. Ele especifica as questões de pesquisa, estratégias para condução, critérios de seleção de estudos e critérios de qualidade. As subseções a seguir apresentam detalhadamente as especificações do protocolo.

3.1.1.1 Escopo e Objetivos

Este mapeamento sistemático tem como objetivo analisar o estado da arte na área de Banco de Dados, de forma que seja possível caracterizar as diretrizes de evolução de banco de dados e elencar as principais práticas associadas.

3.1.1.2 Estrutura das Questões

As questões de pesquisa foram estruturadas de acordo com quatro critérios baseados em (KITCHENHAM; CHARTERS, 2007):

- **População:** Todo estado da arte que engloba tanto a academia quanto a indústria.
- **Intervenção:** Evolução de banco de dados
- **Contexto:** Todos os contextos do segmento acadêmico e industrial
- **Resultado:** Práticas/técnicas, tecnologias/ferramentas, *guidelines*/processos/abordagens

3.1.1.3 Questões de Pesquisa

As questões de pesquisa definem o foco para a identificação dos estudos, extração de dados e análise. Dessa forma, foram definidas as seguintes Questões de Pesquisa (QPs):

- QP1.** Quais são os domínios de aplicação dos projetos relatados como alvo de práticas de evolução de banco de dados?
- QP2.** Quais as práticas/técnicas, tecnologias/ferramentas, *guidelines*/processos/abordagens que são apresentadas para realizam a evolução de banco de dados?
- QP3.** Quais as práticas/técnicas, tecnologias/ferramentas, *guidelines*/processos/abordagens que apoiam a evolução de banco de dados?
- QP4.** Quais são as vantagens, desvantagens e tendências relatadas pelos estudos?

Na questão **QP2** buscamos estudos que realmente apresentam técnicas que realizam a evolução de banco de dados. Já na **QP3** buscamos técnicas que ajudam na evolução de banco de dados, seja através de adaptações, análise de mudanças. Essas serão apenas um suporte antes ou depois de ser realizada a evolução.

3.1.1.4 Estratégia de Pesquisa

Para realizar a pesquisa foram utilizadas bases de dados digitais que possuem mecanismos de buscas, que utilizam palavras-chave e que contêm estudos na área da computação. As bases de dados utilizadas foram: ACM Digital Library¹, Engineering Village (Compendex)², IEEE Xplore³, ScienceDirect⁴, SCOPUS⁵ e SpringerLink⁶. Com a finalidade de definir a *string* de busca, utilizamos palavras-chave e sinônimos, podemos vê-los na Tabela 1.

Tabela 1 – Palavras-Chave e Sinônimos

Palavra-Chave	Sinônimos
database	data base data-base datamodel data model schema
evolution	evolutionary evolutive reengineering refactoring

A fim de transformá-los em uma *string*, foram utilizados operadores booleanos, o OR para reunir os sinônimos e o AND associar as palavras-chave. O resultado pode ser

¹ ACM: <http://www.dl.acm.org>

² Compendex: <http://www.engineeringvillage>

³ IEEE Xplore: <http://www.ieeexplore.ieee.org>

⁴ ScienceDirect: <http://www.sciencedirect.com>

⁵ SCOPUS: <http://www.scopus.com>

⁶ SpringerLink: <http://www.link.springer.com>

observado na Figura 2. Entretanto, a *string* construída não pode ser usada em todas as bases de dados da mesma forma, pois cada base de dados possui uma maneira específica de lidar com as palavras-chave e sinônimos. Nesse sentido, foram adotados métodos específicos para construir *strings* de buscas equivalentes para cada base de dados. Tais especificações serão mais detalhadas na Subseção 3.2.1

(database OR "data base" OR "data-base" OR datamodel OR "data model" OR schema) AND (evolution OR evolutionary OR evolutive OR reengineering OR refactoring)

Figura 2 – String de Busca

3.1.1.5 Critérios de Inclusão e Exclusão

Uma das principais atividades durante o mapeamento é a definição dos Critérios de Inclusão (CIs) e dos Critérios de Exclusão (CEs). Esses critérios selecionam os estudos mais apropriados para que o número de artigos de retorno seja reduzido. Neste mapeamento definimos os seguintes CI e CE:

- CI1:** Estudos direcionados a evolução de banco de dados
- CE1:** Estudos que não são escritos em inglês
- CE2:** Estudos que não abordam evolução de banco de dados
- CE3:** Estudos que não abordam bancos de dados atuais
- CE4:** Estudos sobre evolução tecnológica
- CE5:** Estudos que não apresentam práticas/técnicas, tecnologias/ferramentas, *guidelines*/processos/abordagens para realizar ou apoiar a evolução de banco de dados

Para responder ao critério CE3, foi utilizada a ferramenta *Google Trends*⁷ para identificar quais os tipos de bancos de dados mais utilizados atualmente. O objetivo desse critério de exclusão é limitar os tipos de banco apenas aos que estão em alta. Para isso, foi realizada uma comparação entre Banco de Dados Relacional, NoSQL, Banco de Dados Orientado a Objeto e Banco de Dados Distribuído relacionado a quantidade de busca nos últimos cinco anos. Por meio dessa comparação foi possível observar que os tipos de banco de dados mais procurados são o Banco de Dados Relacional e o NoSQL. Também, que os Bancos de Dados Orientados a Objetos e Banco de Dados Distribuídos são muito menos utilizados do que os outros dois anteriormente discutidos. Isto pode ser observado na Figura 3. Dessa forma, os dois menos usados foram excluídos devido a seu baixo uso ser constante comparado aos demais.

3.1.2 Critérios de Qualidade

O propósito dos Critérios de Qualidade (CQs) é verificar quais estudos contribuem mais com os propósitos e motivação deste mapeamento. Dessa forma, as notas geradas

⁷ Google Trends: <https://trends.google.com.br/trends>

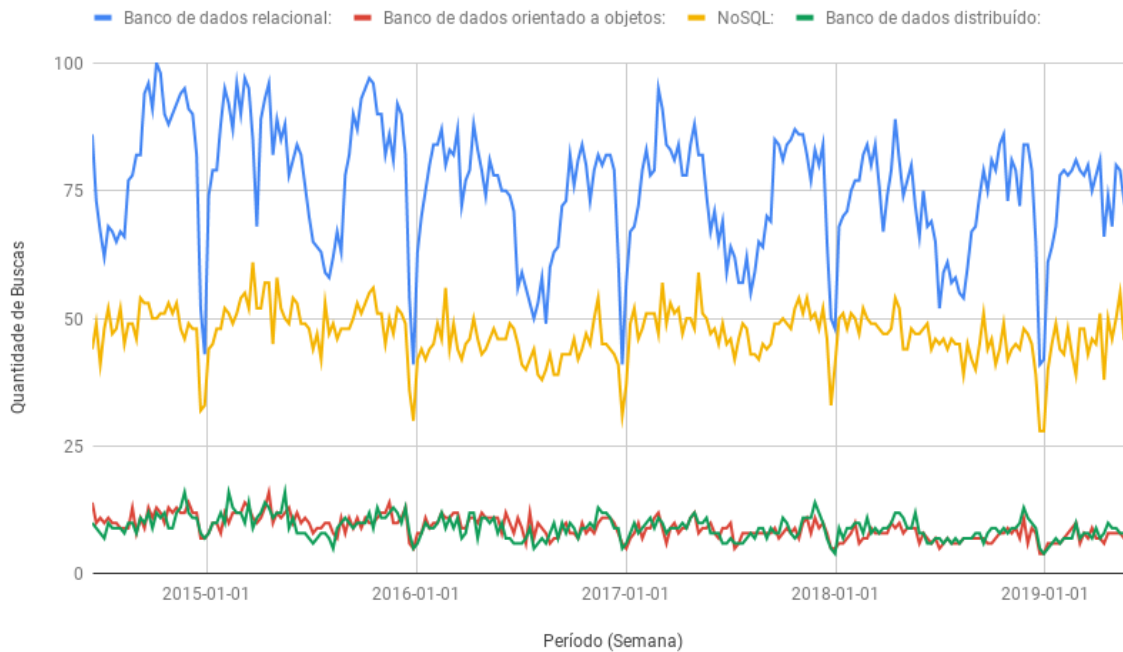


Figura 3 – Comparação de Tipos de Bancos de Dados por meio do Google Trends

pelos critérios não são usadas para eliminar estudos. Definimos os seguintes CQ:

CQ1. O estudo apresenta algum domínio de aplicação como alvo de práticas de evolução de banco de dados?

Respostas - S: O estudo apresenta totalmente o domínio alvo de práticas de evolução de banco de dados; P: O estudo apresenta parcialmente o domínio alvo de práticas de evolução de banco de dados; N: O estudo não apresenta o domínio alvo de práticas de evolução de banco de dados.

CQ2. O estudo apresenta alguma prática/técnica, tecnologia/ferramenta, *guideline*/ processo/abordagem que são apresentadas para **realizar** a evolução de banco de dados?

Respostas - S: O estudo apresenta alguma prática/técnica, tecnologia/ferramenta, *guideline*/processo/abordagem que são apresentadas para realizar a evolução de banco de dados; P: O estudo menciona alguma alguma prática/técnica, tecnologia/ferramenta, *guideline*/processo/abordagem que são apresentadas para realizar a evolução de banco de dados; N: O estudo não apresenta nenhuma alguma prática/técnica, tecnologia/ferramenta, *guideline*/processo/ abordagem que são apresentadas para realizar a evolução de banco de dados.

CQ3. O estudo apresenta prática/técnica, tecnologia/ferramenta, *guideline*/processo/ abordagem que **apoiam** a evolução de banco de dados?

Respostas - S: O estudo apresenta totalmente alguma prática/técnica, tecnologia/ferramenta, *guideline*/processo/abordagem que são apresentadas que apoiam a evolução de banco de dados; P: O estudo menciona alguma prática/técnica, tecnologia/ferramenta, *guideline*/processo/abordagem que são apresentadas que apoiam a evolução de banco de dados; N: O estudo não apresenta nenhuma alguma prática/técnica, tecnologia/ferramenta, *guideline*/processo/abordagem que são apresentadas que apoiam a evolução de banco de dados.

CQ4. O estudo apresenta quais são as vantagens, desvantagens e tendências na parte de evolução de banco de dados?

Respostas - S: O estudo apresenta totalmente vantagens, desvantagens e tendências; P: O estudo apresenta apenas vantagens, desvantagens ou tendência; N: O estudo não apresenta vantagens, desvantagens e tendências.

CQ5. O estudo apresenta uma evolução de banco de dados utilizando integração contínua?

Resposta - S: O artigo apresenta uma evolução de banco utilizando IC; P: O estudo faz referência a uma evolução de banco de dados utilizando IC; N: O estudo não apresenta uma evolução de *software* utilizando IC.

3.1.2.1 Estratégia de Seleção

A estratégia de seleção foi dividida em sete etapas, onde parte dela foi realizada por dois pesquisadores. As etapas da estratégia, assim como os pesquisadores envolvidos nelas, são descritos a seguir:

Etapa 1: Base de dados e seleção inicial: as *strings* foram geradas por meio das palavras-chave e sinônimos. A partir delas foram encontrados os estudos em cada base.

Etapa 2: Eliminando duplicados: nesta etapa, dois pesquisadores trabalharam juntos em uma pré-análise dos artigos a fim de eliminar artigos duplicados.

Etapa 3: Triagem de Título e *Abstract*: nesta etapa, dois pesquisadores realizaram a leitura de título e *abstract* de cada estudo e aplicaram o critério de inclusão CI1 e os critérios de exclusão CE1 e CE2.

Etapa 4: Triagem Inclusão e Exclusão: nesta etapa, um pesquisador realizou a leitura completa dos estudos aplicando os critérios de inclusão e exclusão, dando ênfase aos critérios de exclusão CE3, CE4 e CE5.

Etapa 5: Aplicação Critérios de Qualidade: nesta etapa, um pesquisador realizou a leitura completa dos estudos aplicando os critérios de qualidade a fim de *rankear* os estudos.

Etapa 6: Extração dos Dados: nesta etapa é realizada a extração de dados baseada nas questões de pesquisa utilizando um formulário de extração, apresentado na Subseção 3.1.2.2.

Etapa 7: Análise dos Dados: nesta etapa é realizada a análise dos dados coletados através da extração de dados.

3.1.2.2 Formulário de Extração de Dados

O formulário de extração é essencial para capturar os dados relevantes dos artigos selecionados, sendo elaborado de modo a ajudar a responder às questões de pesquisa. Dessa forma, os seguintes dados foram extraídos de cada estudo:

- Título
- Autor(es)
- Ano
- Qual conferência/anal o artigo foi publicado?
- O objetivo está claro? Sim, Não
- Este estudo aborda: Evolução de Banco de Dados, Evolução de Esquema de Banco de Dados, Evolução de Modelo de Dados, Reengenharia de Banco de Dados, Reengenharia de Esquema de Banco de Dados, Reengenharia de Modelo de Dados, Refatoração de Banco de Dados, Refatoração de Esquema de Banco de Dados, Refatoração de Modelo de Dados
- Qual o domínio de aplicação alvo da práticas de evolução de banco de dados?
- Quais as práticas/técnicas, tecnologias/ferramentas, *guidelines*/processos/abordagens que são apresentadas para realizam a evolução de banco de dados?
- Quais as práticas/técnicas, tecnologias/ferramentas, *guidelines*/processos/abordagens que apoiam a evolução de banco de dados?
- Quais são as vantagens das propostas de evolução?
- Quais são as desvantagens das propostas de evolução?
- Quais são as tendências das propostas de evolução?

3.1.2.3 Estratégia de Análise

Com o intuito de identificar formas de realizar ou apoiar a evolução de banco de dados, os dados foram tabulados como abordagem, ferramentas, modelos, linguagens, extensões dentre outros. Além disso, com o intuito de verificar a quantidade de estudos por categoria que foram utilizados para responder às questões de pesquisa, foi elaborado um mapa por meio do gráfico de bolha. Com isto, tornou-se possível a identificação de problemas e lacunas na área de evolução de banco de dados.

3.2 Condução

A condução foi realizada de acordo com as etapas mencionadas na Subseção 3.1.2.1. Nessa seção demonstraremos mais detalhadamente cada etapa realizada na condução deste mapeamento sistemático⁸.

3.2.1 Strings de Busca Específicas

Como havia sido apresentado na Subseção 3.1.1.4, cada base de dados necessita de uma *string* específica. Nesse sentido, antes de realizarmos a busca pelos estudos em cada base de dados, elaborou-se *strings* específicas para cada. Para que as *strings* ficassem similares e adaptadas de acordo com as especificações e limitações de cada base de dados, e com o propósito de se obter o melhor resultado em cada base de dados, utilizou-se como base a *string* apresentada na Figura 2. O resultado das *strings* específicas para cada base de dados pode ser conferido na Tabela 2.

Tabela 2 – Strings de Busca Utilizadas em Cada Base

Base de Dados	String de Busca
ACM DL	(database "data basedata-base"datamodel "data model"schema) AND (evolution evolutionary evolutive reengineering refactoring)
Compendex	("database evolution"OR "data base evolution"OR "data-base evolution"OR "data model evolution"OR "datamodel evolution"OR "schema evolution"OR "database evolutionary"OR "data base evolutionary"OR "data-base evolutionary"OR "data model evolutionary"OR "datamodel evolutionary"OR "schema evolutionary"OR "database evolutive"OR "data base evolutive"OR "data-base evolutive"OR "data model evolutive"OR "datamodel evolutive"OR "schema evolutive"OR "database reengineering"OR "data base reengineering"OR "data-base reengineering"OR "data model reengineering"OR "datamodel reengineering"OR "schema reengineering"OR "database refactoring"OR "data base refactoring"OR "data-base refactoring"OR "data model refactoring"OR "datamodel refactoring"OR "schema refactoring")
IEEE Xplore	(("database"OR "data base"OR "data-base"OR "data model"OR "datamodel"OR "schema") NEAR/2 ("evolution"OR "evolutionary"OR "evolutive"OR "reengineering"OR "refactoring"))

⁸ SMS - github.com/julianamareco/evolution-db/wiki/Estudo-de-Mapeamento-Sistem%C3%A1tico

Tabela 2 continuação da página anterior

Science Direct	("database evolution"OR "data base evolution"OR "data-base evolution"OR "data model evolution"OR "datamodel evolution"OR "schema evolution"OR "database evolutionary"OR "data base evolutionary"OR "data-base evolutionary"OR "data model evolutionary"OR "datamodel evolutionary"OR "schema evolutionary"OR "database evolutive"OR "data base evolutive"OR "data-base evolutive"OR "data model evolutive"OR "datamodel evolutive"OR "schema evolutive"OR "database reengineering"OR "data base reengineering"OR "data-base reengineering"OR "data model reengineering"OR "datamodel reengineering"OR "schema reengineering"OR "database refactoring"OR "data base refactoring"OR "data-base refactoring"OR "data model refactoring"OR "datamodel refactoring"OR "schema refactoring")
SCOPUS	TITLE-ABS-KEY (("database" OR "data base" OR "data-base" OR "data model" OR "datamodel" OR "schema.") W/2 ("evolution" OR "evolutionary" OR "evolutive" OR "reengineering" OR "refactoring")) AND (LIMIT-TO (SUBJAREA,"COMP")) AND (LIMIT-TO (LANGUAGE,"English"))
Springer Link	("database evolution"OR "data base evolution"OR "data-base evolution"OR "data model evolution"OR "datamodel evolution"OR "schema evolution"OR "database evolutionary"OR "data base evolutionary"OR "data-base evolutionary"OR "data model evolutionary"OR "datamodel evolutionary"OR "schema evolutionary"OR "database evolutive"OR "data base evolutive"OR "data-base evolutive"OR "data model evolutive"OR "datamodel evolutive"OR "schema evolutive"OR "database reengineering"OR "data base reengineering"OR "data-base reengineering"OR "data model reengineering"OR "datamodel reengineering"OR "schema reengineering"OR "database refactoring"OR "data base refactoring"OR "data-base refactoring"OR "data model refactoring"OR "datamodel refactoring"OR "schema refactoring")

Além dos operadores booleanos, utilizamos operadores de proximidade como "NEAR" presente na *string* da IEEE Xplore e "W" presente na *string* da SCOPUS que são utilizados para buscar termos próximos. Dessa forma é possível restringir a distância em que os termos podem estar uns dos outros quando apresentados no texto. Tais operadores foram utilizados para que retornassem mais estudos específicos da evolução de banco de dados, e não estudos que possuem os termos evolução e banco de dados, mas que não estão rela-

cionados à área de banco de dados. Outras especificações usadas na *string* da SCOPUS foram o "LIMIT-TO", que serve para limitar os estudos que retornam a uma categoria ou linguagem, como é o caso do que foi utilizado neste mapeamento, e "TITLE-ABS-KEY" que serve para reduzir a busca dos termos em título, resumo e palavras-chave dos estudos.

Para as bases como Compendex, ScienceDirect e SpringerLink, a *string* passou a conter todos os termos possíveis unidos. Isso ocorreu devido ao retorno de grandes quantidades de estudos que não pertenciam à área de banco de dados e por não ter operadores de aproximação.

3.2.2 Busca nas Bases de Dados

Após a elaboração das *strings* de buscas específicas, foram realizadas buscas nas suas respectivas bases. Em algumas bases, o retorno dessas buscas passaram por filtragem, como é o caso da Compendex que foi filtrada pela linguagem e a SpringerLink. Esta base foi filtrada por linguagem e categoria, removendo-se os estudos cujos artigos ofereciam apenas as suas prévias. Além disso, na ScienceDirect foi possível realizar a busca apenas por título, resumo e palavra-chave. A quantidade de estudos retornados após busca e filtragem pode ser visualizado na Tabela 3.

Tabela 3 – Estudos Importados

Base de Dados	Total
ACM DL	1175
Compendex	696
IEEE Xplore	286
Science Direct	575
SCOPUS	993
Springer Link	957
Total Importados	4682

3.2.3 Seleção dos Estudos

A seleção dos estudos seguiu a estratégia de seleção, presente na Subseção 3.1.2.1, aonde a primeira etapa foi detalhada na Seção 3.2.2. Após esta primeira etapa, foram selecionados previamente 4682 estudos. Desse resultado aplicou-se a remoção de duplicados, segunda etapa da seleção, com o objetivo de retirar todos os artigos que estavam repetidos independente de base de dados. A remoção resultou em 3486 estudos para aplicar os critérios de inclusão e exclusão.

Na terceira etapa, a partir da leitura do título e *abstract* de cada um dos estudos, foi aplicado os critérios de inclusão e exclusão. Essa etapa foi realizada por dois pesquisadores e no final da aplicação dos critérios foram aceitos 206 estudos. Os artigos aceitos passaram para a quarta etapa, nessa etapa foi realizamda a aplicação dos critérios de inclusão e

exclusão por meio da leitura completa dos estudos. A finalização desta etapa resultou em 76 artigos aceitos.

O resultado dessas etapas pode ser visualizado na Figura 4. A quinta e sexta etapa são detalhadas nas seções a seguir. Tais etapas não resultam na remoção de estudos.

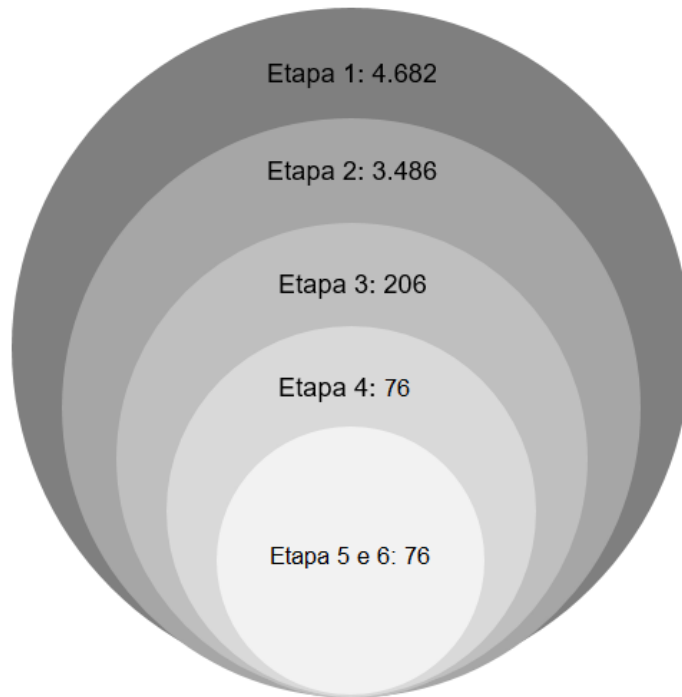


Figura 4 – Diagrama em Camadas dos Estudos Seleccionados

3.2.4 Aplicação de Critérios de Qualidade

A aplicação dos critérios de qualidade tem como objetivo de verificar os estudos que mais contribuem para propósito deste trabalho. A Tabela 7 fornece as informações sobre as pontuações neste quesito. Cada estudo possui um identificador (ID) e sua respectiva referência. As colunas CQ1, CQ2, CQ3, CQ4 e CQ5 mostram a pontuação baseada do estudo em cada critério de qualidade. A coluna "Score Geral" descreve se o estudo é Pouco Relevante, Relevante ou Muito Relevante para nosso objetivo. A coluna "Score" mostra a pontuação final de cada estudo. E, por fim, a coluna Status demonstra se já foi realizada a aplicação dos critérios de qualidade para o estudo. Foram avaliados separadamente 76 estudos, de acordo com os cinco critérios de qualidade apresentados na Subseção 3.1.2. Utilizamos os critérios de qualidade como uma forma de verificar a relevância desse estudo para nossos objetivos, desta forma, nenhum estudo foi excluído nesta etapa.

Tabela 4 – Aplicação dos Critérios de Qualidade

ID	Referência	CQ1	CQ2	CQ3	CQ4	CQ5	Score Geral	Score
A01	(CLEVE; BROGNEAUX; HAINAUT, 2010)	P	S	S	P	N	Relevante	3.0
A02	(MINSKY; ROZENSHTEIN; CHOMICKI, 1986)	S	S	P	P	N	Relevante	3.0
A03	(HSU, 1986)	P	N	S	P	N	Relevante	2.0
A04	(YOON; KERSCHBERG, 1993)	P	P	S	P	N	Relevante	2.5
A05	(CHANG et al., 2007)	P	S	S	P	N	Relevante	3.0
A06	(LERNER, 2000)	P	P	S	S	N	Relevante	3.0
A07	(SCALAS; CAPPELLI; CASTRO, 1993)	P	N	S	P	N	Relevante	2.0
A08	(XUE et al., 2012)	P	S	P	S	N	Relevante	3.0
A09	(GROLINGER; CAPRETZ, 2011)	P	S	S	S	N	Relevante	3.5
A10	(DOMINGUES et al., 2014)	P	P	S	P	N	Relevante	2.5
A11	(PAPASTEFANATOS; VASSILIADIS; VASSILIOU, 2006)	P	N	S	S	N	Relevante	2.5
A12	(GARCIA; DIAZ; CABOT, 2014)	P	S	S	S	P	Muito Relevante	4.0
A13	(NGUEZ; LLORET; ZAPATA, 2003)	P	N	S	P	N	Relevante	2.0
A14	(LIU; CHANG; CHRYSANTHIS,)	S	S	S	P	N	Relevante	3.5
A15	(MACEK; RICHTA, 2014)	P	S	N	S	N	Relevante	2.5
A16	(TERWILLIGER; BERNSTEIN; UNNITHAN, 2010a)	S	S	N	P	N	Relevante	2.5
A17	(SONG et al., 2012)	P	N	S	S	N	Relevante	2.5
A18	(CURINO; MOON; ZANIOLO, 2009)	P	S	N	P	N	Relevante	2.0
A19	(CURINO et al., 2013)	S	S	S	S	N	Muito Relevante	4.0
A20	(MORAES; SALGADO; TEDESCO, 2009)	S	N	S	P	N	Relevante	2.5
A21	(COOX, 2003)	S	N	S	P	N	Relevante	2.5
A22	(YANNAKOUDAKIS; DIAMANTIS, 2008)	S	N	S	P	N	Relevante	2.5
A23	(SCHERZINGER; KLETTKE; STORL, 2015)	S	N	S	N	N	Relevante	2.0
A24	(HERRMANN et al., 2015b)	P	S	N	P	N	Relevante	2.0
A25	(LIU et al., 2010)	S	N	S	P	N	Relevante	2.5
A26	(JONG; DEURSEN, 2015)	S	S	S	P	S	Muito Relevante	4.5
A27	(SCHERZINGER; CERQUEUS; ALMEIDA,)	S	S	S	P	N	Relevante	3.5
A28	(MEURICE; CLEVE, 2014)	S	N	S	P	N	Relevante	2.5
A29	(PROPER, 1997)	P	N	S	P	N	Relevante	2.0
A30	(HERRMANN et al., 2015a)	S	S	N	P	N	Relevante	2.5
A31	(HAINAUT et al., 1994)	S	N	S	P	N	Relevante	2.5
A32	(LIU; CHRYSANTHIS; CHANG, 1994)	N	S	S	P	N	Relevante	2.5
A33	(LIU; CHANG; CHRYSANTHIS, 1994)	S	N	S	P	N	Relevante	2.5
A34	(AND; AND,)	S	N	S	P	N	Relevante	2.5

Tabela 4 – Aplicação dos Critérios de Qualidade

ID	Referência	CQ1	CQ2	CQ3	CQ4	CQ5	Score Geral	Score
A35	(MEURICE; NAGY; CLEVE, 2016)	S	S	S	S	N	Muito Relevante	4.0
A36	(CHEN; MCLEOD; O'LEARY, 1995)	S	S	N	P	N	Relevante	2.5
A37	(GUEDES; BAIOCO; MORAES, 2016)	P	N	S	P	N	Relevante	2.0
A38	(BOMMEL, 1994)	P	N	S	S	N	Relevante	2.5
A39	(BELLAHSENE, 1997)	S	N	S	P	N	Relevante	2.5
A40	(CURINO; MOON; ZANIOLO, 2008a)	P	N	S	P	N	Relevante	2.0
A41	(PAPASTEFANATOS et al., 2008a)	S	N	S	P	N	Relevante	2.5
A42	(PAPASTEFANATOS et al., 2010)	S	N	S	P	N	Relevante	2.5
A43	(PAPASTEFANATOS et al., 2008b)	S	S	S	P	N	Relevante	3.5
A44	(JENSEN; BOHLEN, 2004a)	P	S	S	P	N	Relevante	3.0
A45	(BRUMMERMANN; KEUNECKE; SCHMID, 2013)	S	N	S	P	N	Relevante	2.5
A46	(CURINO; MOON; ZANIOLO, 2008b)	S	N	S	P	N	Relevante	2.5
A47	(DOMÍNGUEZ et al., 2008)	N	P	P	P	N	Não Relevante	1.5
A48	(WANG; SHEN; CHEN, 2009)	S	S	S	S	N	Muito Relevante	4.0
A49	(DOMINGUEZ et al., 2008)	P	N	S	P	N	Relevante	2.0
A50	(BANERJEE; DAVIS, 2009)	S	N	S	P	N	Relevante	2.5
A51	(GOTZ; KUHN, 2015)	S	S	S	S	S	Muito Relevante	5.0
A52	(NEJI et al., 2006)	P	P	S	S	N	Relevante	3.0
A53	(JENSEN; BOHLEN, 2004b)	N	N	S	P	N	Relevante	1.5
A54	(KLETTKE et al., 2016)	S	N	S	S	S	Muito Relevante	4.0
A55	(MORO; MALAIKA; LIM, 2007)	S	N	S	S	N	Relevante	3.0
A56	(MOON et al., 2009)	S	S	N	P	N	Relevante	2.5
A57	(HERRMANN et al., 2018)	S	S	N	P	N	Relevante	2.5
A58	(BELLAHSENE, 2002)	P	N	S	S	N	Relevante	2.5
A59	(VASSILIADIS, 2017)	N	N	S	P	N	Não Relevante	1.5
A60	(CHEN; MCLEOD; O'LEARY, 1994)	N	N	S	P	N	Não Relevante	1.5
A61	(FAN; POULOVASSILIS, 2004)	P	S	N	P	N	Relevante	2.0
A62	(WU; ZHANG; KONG, 2012)	S	N	S	S	N	Relevante	3.0
A63	(BOUNIF; POTTINGER, 2006)	S	N	S	P	N	Relevante	2.5
A64	(ASSOUDI; LOUNIS, 2011)	N	N	S	S	N	Relevante	2.0
A65	(RA, 2004)	S	S	N	P	N	Relevante	2.5
A66	(HICK; HAINAUT, 2003)	S	S	N	P	P	Relevante	2.5
A67	(LAMMARI; AKOKA; COMYN-WATTIAU, 2003)	N	N	S	P	N	Não Relevante	1.5
A68	(MEURICE; CLEVE, 2017)	S	S	S	P	N	Relevante	3.5

Tabela 4 – Aplicação dos Critérios de Qualidade

ID	Referência	CQ1	CQ2	CQ3	CQ4	CQ5	Score Geral	Score
A69	(CURINO et al., 2009)	P	N	S	P	N	Relevante	2.0
A70	(DRAHEIM; HORN; SCHULZ,)	P	N	S	P	N	Relevante	2.0
A71	(SCHULER; KESSELMAN, 2018)	P	N	S	S	N	Relevante	2.5
A72	(BAEKGAARD, 1997)	S	S	S	P	N	Relevante	3.5
A73	(CURINO et al., 2010)	S	S	S	P	N	Relevante	3.5
A74	(ATHINAIIOU; KONDYLAKIS, 2019)	S	N	S	P	N	Relevante	2.5
A75	(TERWILLIGER; BERNSTEIN; UNNITHAN, 2010b)	P	S	S	P	N	Relevante	3.0
A76	(JONG; DEURSEN; CLEVE, 2017)	S	S	S	S	S	Muito Relevante	5.0

3.3 Análise

Após a aplicação dos critérios de qualidade, foi realizada a extração de dados dos estudos aceitos utilizando o formulário de extração já apresentado na Subseção 3.1.2.2⁹. A análise dos estudos foi realizada em 76 estudos. Portanto, nesta seção será apresentado os resultados relacionados as respostas das questões de pesquisa especificadas na Subseção 3.1.1.3.

Com a extração dos dados dos artigos foi possível montar um gráfico de estudos por ano parcial. Neste gráfico, apresentado na Figura 5, podemos ver que, apesar de em alguns altos e baixos, os estudos relacionados a evolução de banco de dados e suas derivações se mostra um tópico atual de pesquisa.

A seguir, apresenta-se a análise pontual das questões de pesquisa.

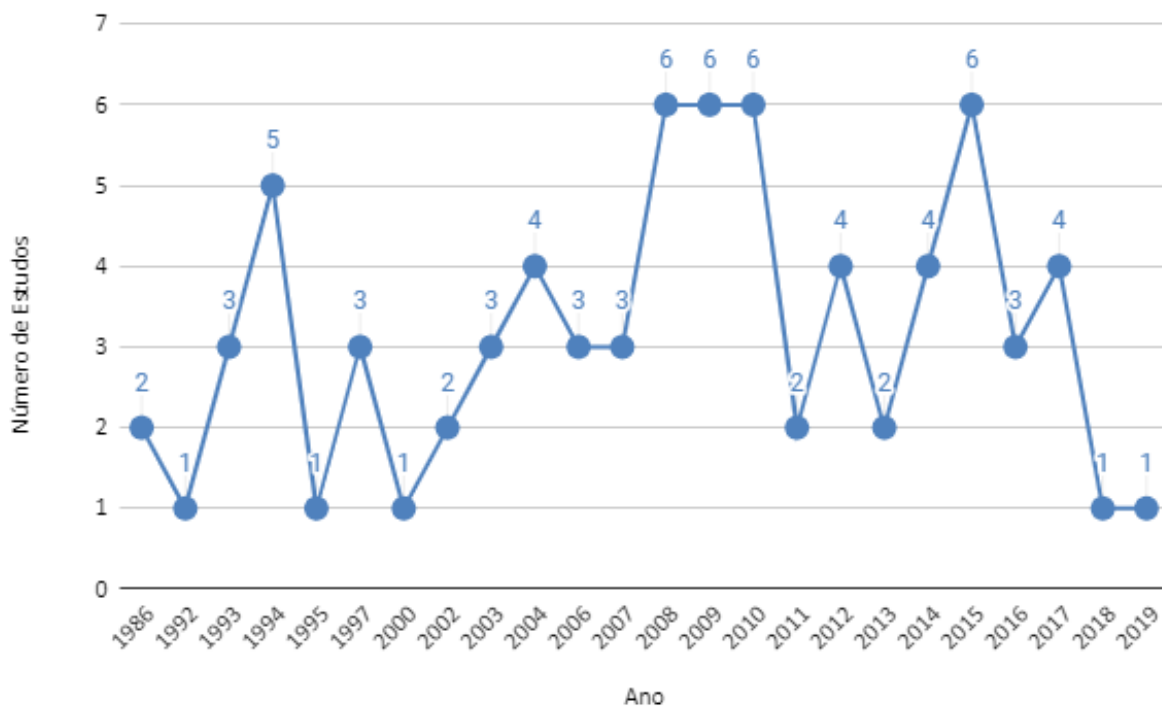


Figura 5 – Quantidade de estudos por ano

QP1. *Quais são os domínios de aplicação dos projetos relatados como alvo de práticas de evolução de banco de dados?*

Após a análise dos estudos selecionados, é possível observar que a maioria dos estudos não aborda um domínio específico de aplicação e sim um tipo de banco de dados específicos. Sabendo disto, foram identificados, sem a especificação, 19 estudos focados em bancos de dados relacionais (A04, A06, A10, A12, A13, A24, A25, A33, A38, A42, A46, A48, A49, A50, A57, A65, A66, A69, A71), um estudo focado em banco de dados relacional e SQL (A11) e um estudo focado em banco de dados relacional e ORMs (A16)

⁹ Extração: github.com/julianamareco/evolution-db/wiki/Extra%C3%A7%C3%A3o-de-Dados

nove estudos focados somente em banco de dados que utilizam a linguagem SQL para realizar seu gerenciamento (A15, A26, A28, A41, A43, A51, A70, A73, A75). Outros bancos relacionais que também foram o foco de alguns estudos, como o PostgreSQL (A76), Oracle (A58), Banco de Dados Temporal (A07) e Banco de dados de Larga Escala (A03) com um estudo apresentado para cada tipo de banco mencionado. E XML com três estudos selecionados (A21, A55, A56).

Seguindo a análise desta questão de pesquisa, foram encontrados: Um estudo que apresenta o uso do banco NoSQL MongoDB e práticas ágeis, para realizar o DC (A54); Três estudos que também apresentam foco em NoSQL, entretanto, sem a especificação de um banco de dados (A23, A27, A68); Outros estudos focaram no processo de evolução em um nível de abstração mais alto, como em modelos Entidade Relacionamento (ER), como tratado no estudo A14, além disto este mesmo estudo também tem como foco um modelo relacional apresentado como *objetivado*.

Alguns estudos utilizam outros tipos de banco de dados, como em A20 cujo foco é *multi tenancy*, seguido do A52 e A61 que têm como objetivo o *Data Warehouse*, o A52 apresenta também um modo de apoiar a evolução em *Data Marts*. Entretanto, foram encontrados três estudos que são independentes do domínio (A01, A63, A72). Houveram estudos que explicitaram os domínios de Linha de Produto de *Software* (SPL) (A30) e *Software* como um serviço (A62). Existem estudos (A02, A22) que tem o objetivo de auxiliar um Sistema de Banco de Dados e um Sistema de Banco de Dados Federados a realizar a evolução, respectivamente.

Até o momento, os demais estudos não apresentam explicitamente o tipo ou domínio do banco de dados, mas oferecem um exemplo de uso. Sabendo-se disto, foi possível categorizá-los por Adaptação de Aplicativos (A17, A34), Modularização de Banco de Dados (A37), Banco de dados MySQL (A08), Sistemas de informação da Web (A18, A40, A45), Banco de Dados atuais (A44), Linguagem Java e MySQL (A47), JPA e Hibernate (A35)

QP2. *Quais as práticas/técnicas, tecnologias/ferramentas, guidelines/processos/abordagens que são apresentadas para realizar a evolução de banco de dados?*

Metade dos estudos apresentam uma maneira de realizar a evolução do banco de dados. Entre toda as apresentadas a que mais se destaca são as abordagens contendo 17 estudos. Em seguida temos três estudos que utilizam ou criam uma ferramenta para realizar a evolução de banco de dados (A30, A66, A69). Outro modo apresentado seria através do uso de modelos propostos, onde são expressados ambas versões do banco e a relação entre si (A15, A51). Três estudos tinham proposta de novas linguagens para evoluir o banco (A24, A57, A72). Também foram encontrados três artigos que apresentavam *frameworks* com o mesmo intuito (A05, A18, A27). Extensões de Ambientes Integrados de Desenvolvimento (IDEs) e de Linguagens já utilizadas no cenário de desenvolvimento,

como SQL, também são utilizadas para realizar de maneira (semi-)automatizada a evolução do banco de dados, como os estudos A43, A65 e A75.

Dois estudos que apresentam sistemas, um utilizando uma linguagem de evolução (A74) e outro utilizando operadores de modificação de esquema (A56). Em decorrência de não apresentar uma contribuição substancial, mas sim discutir como o engenheiro de software deve proceder para realizar a evolução, alguns estudos foram classificados como outros (A04). Dentro desta categoria também foram encontrados um ambiente para evolução, um kit de ferramentas (A31), um conjunto de transformações (A61) e uma técnica relacionada a mapeamento de modelos (A16). Além desses, também foram encontrados um *Schema Modification Operators* (SMO), responsável por definir uma série de modificações de esquema por meio de operações (A19), os SMO são consideravelmente utilizados em outras propostas para evolução. Na Tabela 8, é possível encontrar mais detalhes relacionados a cada modo mencionado.

Tabela 5 – Detalhes das realizações de evolução de banco de dados

ID	Tipo	Descrição
A01	Abordagem	Abordagem baseada em representação genérica de esquemas e em uma definição formal de transformação e mapeamentos de esquemas.
A02	Abordagem	Abordagem baseada na linguagem Prolog com o objetivo de adicionar restrições no banco de dados através da criação de um modelo.
A04	Outros	Conciliam ou harmonizam novas regras do banco de dados com as antigas, a fim de manter a consistência.
A05	<i>Framework</i>	<i>Framework</i> para refatoração de banco de dados, baseado em lógica epistêmica, usa um agente para descobrir e resolver inconsistências e analisa o impacto das alterações nas consultas.
A08	Abordagem	Abordagem para a evolução de esquemas, que suporta a compatibilidade com as versões anteriores para isso é utilizado as <i>views</i> .
A09	Abordagem	Abordagem suportada por teste unitário para o código de aplicativo que acessa bancos de dados com o objetivo de avaliar proativamente o código em relação ao banco de dados alterado.
A12	Abordagem	Abordagem para evolução de esquema de banco de dados
A14	Abordagem	Abordagem para a evolução do esquema através de alterações no esquema Entidade-Relacionamento (ER) de um banco de dados
A15	Modelo	Modelo formal capaz de migrar o banco de dados partindo das refatorações que foram criadas no código de aplicação.
A16	Técnica	A técnica é relacionada a mapear o modelo físico para um conceitual, e evoluir ambos em conjunto.
A18	<i>Framework</i>	<i>Framework</i> que automatiza totalmente o gerenciamento da migração de dados, consulta e visualização das adaptações mediante alterações estruturais do esquema.
A19	Outros	Conjunto de Operadores de Modificação de Esquema (SMOs) baseados em SQL.

Tabela 5 – Detalhes das realizações de evolução de banco de dados

ID	Tipo	Descrição
A24	Linguagem	Linguagem de referência para implementações produtivas de evolução de banco de dados em Sistemas de Gerenciamento de Banco de Dados (SGBDs) denominada CoDEL.
A26	Abordagem	Abordagem para evolução de banco de dados, no contexto de Integração Contínua, utilizando versões do mesmo banco.
A27	<i>Framework</i>	<i>Framework</i> para evolução controlada de esquemas no desenvolvimento de aplicativos em relação a armazenamentos de dados NoSQL.
A30	Ferramenta	Kit de ferramentas de evolução de banco de dados para o SPL.
A31	Outro	Ambiente CASE que possui funções para engenharia direta e reversa de banco de dados e assistente no processamento de esquemas.
A32	Abordagem	Abordagem de derivação de esquemas para a evolução através de mudanças no esquema Entidade-Relacionamento.
A35	Outros	Pega os códigos de um versionamento e computa as diferenças entre as anotações JPA dos fontes Java.
A36	Abordagem	Abordagem baseada no conhecimento para apoiar a evolução do esquema de banco de dados contábil dinâmico em um contexto de modelagem de dados baseado em objeto.
A43	Extensão	Extensão viável e poderosa para a linguagem SQL adaptada para o gerenciamento da evolução.
A44	Outros	Esquema concluído estendido que registra atributos juntamente com suas incompatibilidades.
A48	Abordagem	Abordagem baseada em modelos para a gerência e realização da evolução.
A51	Modelo	Modelo de tempo de execução que realiza a evolução do esquema de banco de dados e analisa o código fonte facilitando a configuração.
A52	Abordagem	Abordagem que adapta o MS e é baseada em um conjunto de operadores algébricos.
A56	Sistema	Utiliza o PRIMA um sistema que tem como base o uso de SMOs (Operadores de modificação de esquema).
A57	Linguagem	Linguagem de evolução de banco de dados relacional CODEL.
A61	Outros	Conjunto simples de transformações primitivas de esquema aumentadas com uma linguagem de consulta funcional, ambas aplicáveis a vários modelos de dados.
A65	Extensão	Extensão SQL capaz de lidar com a evolução de esquemas em sistemas de bancos de dados relacionais.
A68	Abordagem	Abordagem automática para inferir o esquema de um banco de dados NoSQL sem esquema e analisar sua evolução ao longo do tempo.
A69	Ferramenta	Ferramenta que suporta evolução de esquema relacional.
A72	Linguagem	Linguagem de duas camadas ortogonal às linguagens de especificação baseadas em estado, podendo ser um complemento para tais.
A73	Outros	<i>Schema Modification Operators (SMO)</i> , denominado PRISM ++.

Tabela 5 – Detalhes das realizações de evolução de banco de dados

ID	Tipo	Descrição
A74	Sistema	Sistema apresentado pelo artigo utiliza o BiDEL que é uma linguagem de evolução de banco de dados que tem o objetivo de evoluir o banco de forma automatizada depois de especificadas suas alterações.
A75	Extensão	Extensão do Visual Studio que suporta a evolução automática de artefatos de mapeamento objeto-relacional no <i>Microsoft Entity Framework</i> .
A76	Abordagem	Abordagem para implantar vários esquemas ativos simultaneamente e migrar dados entre esses esquemas.

QP3. *Quais as práticas/técnicas, tecnologias/ferramentas, guidelines/processos/abordagens que apoiam a evolução de banco de dados ?*

A maioria dos estudos foram encontrados com o objetivo de apoiar a evolução de banco de dados. Com 10 estudos, o uso de ferramentas caracteriza a maioria dos tipos de apoio a evolução encontrado, entre eles estão os estudos: A01, A20, A26, A27, A28, A38, A40, A42, A50 e A75. Outros dois métodos que se destacaram foram as abordagens e modelos, contendo respectivamente oito (A17, A44, A47, A48, A52, A58, A68, A76) e sete (A07, A11, A21, A22, A32, A51, A53) estudos.

Frameworks (A23, A43, A46, A70, A71), *Processos* (A09, A10, A37, A64), *Sistemas* (A03, A06, A29, A74) e *Tecnologias* (A12, A14, A63). Foram encontrados também estudos que usam arquiteturas (A13, A49, A62). Mecanismos para apoiar alguma etapa como, por exemplo, a migração de dados entre um esquema antigo para um esquema evoluído, também tiveram destaque (A34, A39), bem como as linguagens (A02 e A33).

Dentre os estudos analisados até o momento foi encontrada uma lógica epistêmica, que é uma lógica complementar a lógica clássica que trata dos conhecimentos, também foi utilizada (A05). Seguindo da lógica epistêmica, também apareceram outros modos como metodologia para realizar o gerenciamento (A67), estratégias de como realizar a evolução (A54), uma plataforma de apoio (A25). Os SMOs (A73) também foram utilizados como apoio à evolução juntamente com *Guidelines* (A31) de como esta atividade pode ser realizada. Outro modo de apoio à evolução é através do uso de técnicas de banco de dados como, por exemplo, consultar em banco de dados para descobrir regras (A04). Estes estudos por possuírem poucos em suas categorias foram separados no tipo Outros. Na tabela 9, é possível verificar mais detalhes relacionados a cada estudo que apoia a evolução de banco de dados.

Tabela 6 – Detalhes dos apoios à evolução de banco de dados

ID	Tipo	Descrição
A01	Ferramenta	DB-MAIN , uma ferramenta de modelagem genérica dedicada à engenharia de aplicações de bancos de dados e, em particular, ao design de banco de dados, engenharia reversa, reengenharia, integração, manutenção e evolução.
A02	Linguagem	Linguagem Prolog.
A03	Sistema	Sistema de Suporte a Decisão (DSS) para sistemas de banco de dados (DBS) baseado na filosofia e metodologia de controle adaptativo e confiabilidade. Além disso, arquitetura de quatro esquemas que promete a independência do modelo de dados.
A04	Outros	Técnicas de banco de dados para descobrir novas regras. Usam uma consulta de banco de dados para descobrir novas regras, usando não apenas exemplos positivos (a resposta a uma consulta) mas também exemplos negativos (que não são aceitas e que podem fazer sentido de ter).
A05	Outros	Lógica epistêmica é a lógica do conhecimentos. Trata dos mecanismos de raciocínio do conhecimento e do processo de revisão de uma base de crenças.
A06	Sistema	Sistema chamado Tess que usa modelos para reconhecer mudanças de tipo comparando esquemas e então produz um transformador que pode atualizar dados em um banco de dados para corresponder a uma versão mais nova do esquema.
A07	Modelo	Modelo para registrar a evolução do esquema e um mecanismo para gerenciar catálogos e conjuntos de dados quando ocorrem mudanças
A09	Processo	Testes unitários e estratégias de teste.
A10	Processo	Processo, com passo a passo e o que fazer para realizar a evolução do banco de dados utilizando refatoração.
A11	Modelo	Modelo baseado em gráficos que captura, de maneira uniforme, relações, visualizações, restrições e consultas.
A12	Tecnologia	Instruções MOFScript com instruções SQL para realizar as transformações.
A13	Arquitetura	Arquitetura para gerenciar a evolução do banco de dados com uma estratégia avançada.
A14	Tecnologia	Esquema Entidade-Relacionamento.
A17	Abordagem	Abordagem (AppCatchupDB) para adaptar automaticamente aplicativos de <i>software</i> à evolução de seus bancos de dados.
A19	Outros	Migração de Dados
A20	Ferramenta	Ferramenta, a AutonomousDB, que utiliza conceitos de Computação Autônoma aplicada ao SGBDs, para auxiliar o DBA na tarefa de evolução do esquema e atualização de propagação
A21	Modelo	Modelo axiomático que fornece uma solução geral para o problema de suportar a evolução de um esquema de banco de dados XML.

Tabela 6 – Detalhes dos apoios à evolução de banco de dados

ID	Tipo	Descrição
A22	Modelo	Modelo de dados capaz de lidar com alterações nos esquemas subjacentes (conceituais e lógicos).
A23	<i>Framework</i>	Estrutura para gerenciar avidamente a evolução de esquema em armazenamentos de documentos NoSQL sem esquema.
A25	Outros	Plataforma que manipula a evolução de dados em bancos de dados orientados por colunas de maneira eficiente (CODS).
A26	Ferramenta	Ferramenta para realizar a evolução de bando de dados o QuantumDB, em desenvolvimento.
A27	Ferramenta	IDE a qual é integrada.
A28	Ferramenta	Ferramenta de visualização que permite analisar o histórico de evolução de um banco de dados ao longo de sua vida útil.
A29	Sistema	Sistema de gerenciamento de versão para evolução de esquema.
A31	Outros	Quatro estratégias importantes para lidar com evolução, ou seja, manutenção antecipada, manutenção reversa, engenharia reversa e antecipação do projeto.
A32	Modelo	Modelo entidade relacionamento (ER)
A33	Linguagem	Linguagem de especificação gráfica para suportar a evolução do esquema com base na abordagem de Entidade Relacionamento (ER) para modelagem de dados.
A34	Mecanismo	Mecanismo de apoio de ponta a ponta à evolução de esquema de banco de dados.
A35	Outros	Apresentam um método denominado que consegue recomendar mudanças.
A37	Processo	Processo para conduzir a evolução do esquema de banco de dados, estendendo a modularização do banco de dados para funcionar de maneira evolucionária.
A38	Ferramenta	Ferramenta de projeto de banco de dados chamada de EDO (<i>Evolutionary Database Optimizer</i>) que tem o objetivo de gerar um conjunto inicial de representações internas preliminares para um dado modelo de dados conceitual.
A39	Mecanismo	Mecanismo de visualização aprimorado com recursos de importação/exportação para suportar a evolução do esquema.
A40	Ferramenta	PRISM, uma ferramenta que suporta a atividade temporal e suscetível a erros da evolução de esquema.
A43	<i>Framework</i>	<i>Framework</i> coerente baseado em gráficos para capturar o efeito de possíveis mudanças no <i>software</i> de banco de dados de um Sistema de Informações.
A44	Abordagem	Abordagem paramétrica para resolver inconsistências de acordo com as necessidades do aplicativo.
A45	Outros	Guideline de suporte para a evolução.
A46	<i>Framework</i>	Framework projetado para fornecer ferramentas.

Tabela 6 – Detalhes dos apoios à evolução de banco de dados

ID	Tipo	Descrição
A47	Abordagem	Abordagem de explicação dirigida.
A48	Abordagem	Abordagem para medir possíveis impactos nas modificações e utiliza tecnologias como eclipse, QVT, <i>model transformations</i> e EMF.
A49	Arquitetura	MeDEA, uma arquitetura de evolução de banco de dados com propriedades de rastreabilidade.
A50	Ferramenta	Ferramenta de <i>software</i> que permite a visualização do impacto da evolução do esquema através do uso de gatilhos e procedimentos armazenados.
A51	Modelo	Modelo para retirar as informações de mapeamento do estado de tempo de execução do aplicativo.
A52	Abordagem	Abordagem de duas fases a primeira compara o novo requisito com o MS (o que caracteriza um data mart) para detectar os novos elementos de requisito
A53	Modelo	Modelos conceituais para mudanças de esquema condicionais com muitas propriedades desejáveis que são essenciais para relações que são tuplas com diferentes esquemas.
A54	Outros	Estratégias para evolução de esquema lento de longo prazo escalável em armazenamentos de dados NoSQL.
A55	Outros	Taxonomia para auxiliar a evolução de esquema e o impacto dessas alterações na validação, formulação de consultas e resultados.
A58	Abordagem	Abordagem para adaptar dinamicamente as visualizações de acordo com as mudanças de esquema.
A59	Outros	Padrões para auxiliar na evolução de banco de dados.
A60	Outros	Conjunto de operadores básicas de esquemas de evolução.
A62	Arquitetura	Arquitetura de metadados <i>Multi-version</i> .
A63	Tecnologia	Repositório de esquemas que é utilizado juntamente com uma Abordagem Preditiva para Evolução do Esquema do Banco de Dados.
A64	Processo	Processo para apoiar a evolução do banco de dados propondo uma solução para reduzir a intervenção humana necessária para manter os processos de troca de dados após a evolução do esquema.
A67	Outros	Metodologia para gerenciar evoluções de banco de dados, explorando duas ontologias.
A70	<i>Framework</i>	<i>Framework</i> que suporta a evolução simultânea de modelos de dados orientados a objetos e esquemas relacionais com respeito a um mapeamento relacional de objetos suportado por ferramentas.
A71	<i>Framework</i>	Estrutura de evolução de esquemas baseada em uma abordagem algébrica (CHiSEL) que introduz operadores relacionais compostos estendidos e de nível superior adaptados à tarefa de evolução de esquemas.
A73	Outros	PRISM ++, um <i>Schema Modification Operators</i> (SMO), oferece suporte de ponta a ponta para a adaptação de consultas e atualizações através de mudanças estruturais de esquema e evolução de restrições de integridade.

Tabela 6 – Detalhes dos apoios à evolução de banco de dados

ID	Tipo	Descrição
A74	Sistema	Sistema que permite a exploração de múltiplas versões de esquema (VE-SEL) usando consultas de proveniência.
A75	Ferramenta	Visual Studio apoia já que o MoDEF depende do mesmo para funcionar.
A76	Abordagem	Para realizar a criação das tabelas fantasma e criação das views a abordagem utiliza da ferramenta QuantumDB.

Uma vez que muitos artigos foram analisados e classificados para as questões de pesquisa, foi possível criar um gráfico de bolha, como ilustra a Figura 6. Neste gráfico, apresenta-se os atributos de qualidade associados com atividades que visam evoluir ou apoiar a evolução de bancos de dados. Nesse gráfico pode ser observado a qual categoria de evolução cada um dos estudos aborda, com seus respectivos identificadores.

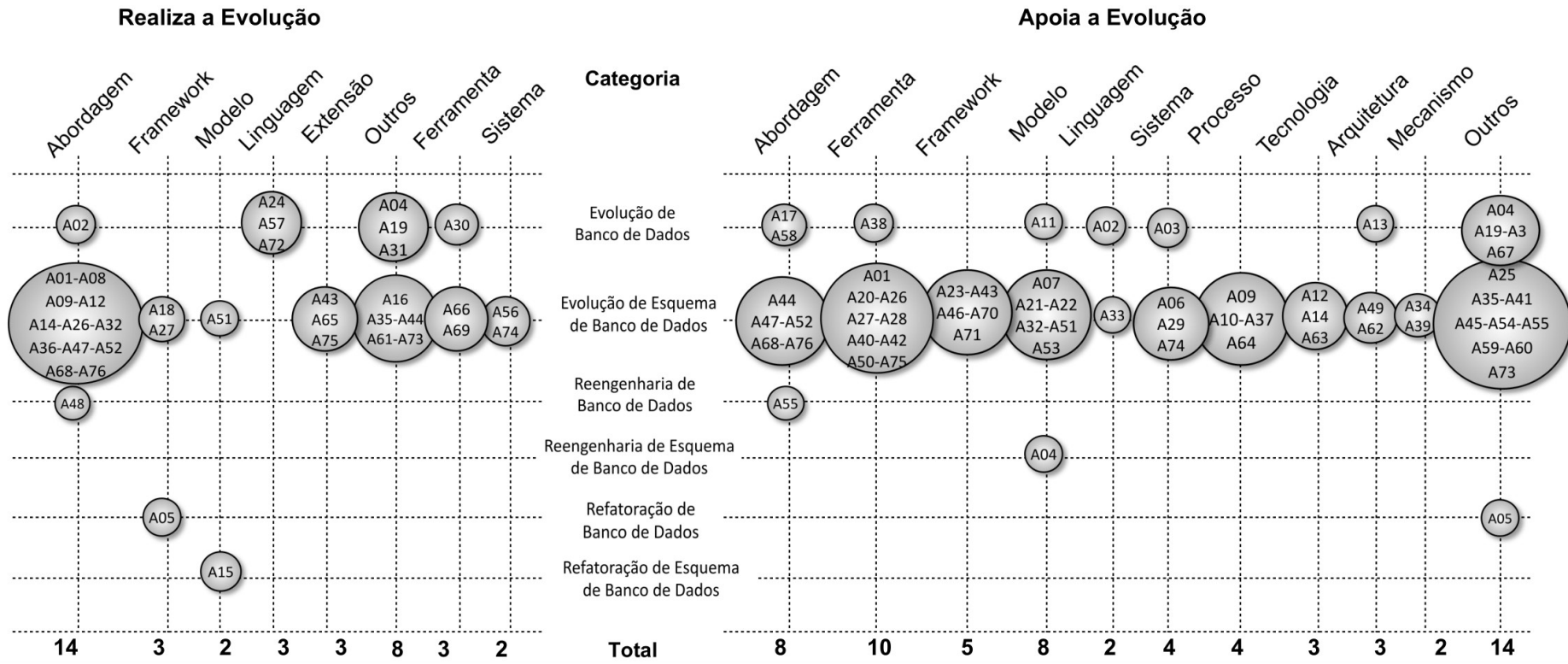


Figura 6 – Métodos de realizar e abordar evolução de banco e suas categorias

QP4. *Quais são as vantagens, desvantagens e tendências relatadas pelos estudos?*

De acordo com a análise, é possível salientar algumas vantagens apresentadas pela adoção da evolução de banco de dados, como: 1) uma melhor **rastreabilidade** de evoluções e dados entre as tabelas; 2) **praticidade** na realização da evolução, já que algumas soluções não precisam especificar todos os dados relacionados às mudanças; 3) **simplicidade**, uma vez que algumas arquiteturas propostas tendem a simplificar toda a evolução do banco de dados; 4) **adaptabilidade** é um ponto forte apresentado foram as migrações de dados, onde torna-se possível migrar os dados entre versões dos bancos, fazendo com que se tornem bancos flexíveis e ágeis, o que sem estas tecnologias torna-se uma tarefa muito complicada; 5) **corretude**, já que atualmente existem soluções que podem sugerir correções a serem realizadas em cima do esquema do banco de dados e notificar automaticamente possíveis riscos e incompatibilidades, e; 6) **manutenibilidade**, já que a utilização de *views* permite isolar o problema da evolução com uma camada entre o sistema e o banco, assim quando o banco for evoluído o sistema não é totalmente afetado.

Outras soluções podem ser encontradas no apoio deste processo como o controle das versões do banco, com representação visual dessas versões no formato de grafo dirigido. Os seguintes atributos de qualidade são observados: 1) Reduções de esforço, como no caso de Armazém de dados, existem propostas que tem o intuito de propagar alterações realizadas entre todos os seus subconjuntos, minimizando assim o esforço necessário para a evolução; 2) Coevolução de dados e esquemas SQL, já que devido a possibilidade de se conseguir realizar uma fácil relação entre as versões de um esquema de banco de dados, propostas de modelos para especificar transformações de modelos foram muito citadas; 3) Automações do processo, como encontrado em relatos de soluções apresentadas por estudos mais atuais, estão focadas no aumento de desempenho, *deploy* contínuo e evolução do banco durante a manutenção do sistema, e 4) Alterações em *runtime*, já que estas operações ocorrem sem que banco de dados fique fora do ar, ou seu usuário perceba e alta escalabilidade, fornecendo também outros atributos como liberdade e flexibilidade para realizar alterações.

Mesmo com todas as vantagens citados, ainda existem problemas dentro do cenário de evolução de banco de dados, como: 1) **Imaturidade** da tecnologia, já que observa-se uma falta de garantia que as tecnologias propostas funcionam em todos os casos, pois algumas não possuem avaliação ou até mesmo estão em fase de desenvolvimento; 2) **Desempenho**, uma vez que parte das abordagens não se preocupa com este fator, o que pode gerar gargalos e impactar diretamente no *software* que utiliza o banco de dados evoluído, além de viabilizar possíveis problemas nas consultas; 3) **Processo manual** para o controle da evolução de banco de dados. Algumas abordagens tentam reduzir a quantidade de escolhas cruciais que o mantenedor precisa fazer, entretanto este profissional ainda é responsável por verificar todo o risco de utilizar alguma das abordagens propostas ou evoluções indicadas; 4) **Limitações de escopo**, pois as soluções apresentadas muitas

vezes estão vinculadas a uma tecnologia ou ferramenta específica o pode acabar limitando o seu uso; 5) **Inconsistências e perda de dados**, pois mesmo que algumas soluções realizem a migração de dados, nem todas as necessidades estão mapeadas, podendo acarretar possíveis perdas de dados em alguns casos de coevolução de modelos e dados, e 6) **Incompatibilidade** entre versões de sistemas que assistem a evolução, o que pode gerar um alto custo para manter a evolução do banco.

Sabendo dessas dificuldades no cenário de evolução, os estudos apresentam algumas tendências: 1) **Transformações bidirecionais**, ajudando na tradução de um modelo para o outro, gerando de alguma forma um rastro utilizável por sistemas de gerenciamento das atualizações; 2) **Modularização**, que para o contexto de banco de dados com módulos configura uma tendência de pesquisa. Pode-se destacar aqui a criação de alguma linguagem de definição de módulos para que assim possa se gerar ou extrair dados para a evolução do banco de acordo com os módulos desenvolvidos, além de ferramenta assistivas para aumentar o nível de abstração das entidades do sistema; 3) **Isolamento do problema**, uma vez que o uso de *views* se apresentou como prática promissora para gerenciar alterações e isolar os problemas decorrentes da execução de novas iterações do processo de desenvolvimento. Entretanto, faltam estudos que explorem diferentes técnicas de utilização de *views*; 4) **Suporte ferramental** como *plugins*, que caracteriza outra tendência para a inserção de tecnologias de evolução de banco nos SGBDs atuais; 5) **Guidelines**, projetadas com o intuito de ajudar o mantenedor a escolher qual a melhor estratégia de evolução. Uma possível proposta apresentada pelos estudos seria a criação de modelos que capturem as compensações de acordo com a escolha; 6) **Visualização de dados**, já que foram apresentadas também como promissor estudos tratando do aumento nas técnicas de visualização das evoluções e sua interdependência com artefatos do processo de software, e 7) O **suporte ferramental** de evolução para diversos SGBDs, que pode tornar possível a realização de mais testes na indústria, juntamente com configurações dinâmicas.

Como tecnologias a serem utilizadas foram indicados duas boas perspectivas: o uso de ontologias para os esquemas de banco de dados, e o uso de práticas de *Model-Driven Engineering* (MDD) para aumentar o nível de abstração e aumentar o dinamismo. Outra possibilidade é quanto ao fornecimento destas tecnologias centradas em SGBDs também nas IDEs, pois, juntamente com a identificação de padrões e sequências de evolução, tornaria possível ativar evoluções através de gatilhos, além de um modo de medir o impacto das decisões tomadas pelos desenvolvedores. Devido a possibilidade de se manter o histórico, para o mapeamento entre os esquemas, também foram apresentadas tendências de integrar da evolução com o aprendizado de máquina.

3.4 Discussão

3.4.1 Resultados

De acordo com os estudos analisados é possível inferir que os termos não são totalmente concretos dentro da literatura, partindo algumas vezes do conhecimento tácito dos desenvolvedores. Um exemplo disto é a diferenciação de evolução de banco de dados e esquema de banco de dados, este é um conceito não muito definido o que acarreta em estudos com o mesmo objetivo, referenciando de diferentes modos. Os limites entre os termos também não são explícitos de forma clara, como por exemplo, quando uma refatoração se torna uma evolução dentro do esquema de banco de dados.

Algumas áreas como evolução de modelos de dados, reengenharia de esquema de banco de dados, reengenharia de modelos de dados, refatoração de modelos de dados até o momento não apresentaram nenhum estudo classificado o que pode ser um possível *gap* de pesquisa. Alguns outros estudos como reengenharia de banco de dados, refatoração de banco de dados e refatoração de esquema de banco de dados apresentaram poucos estudos com o foco na área, o que pode acarretar falta de apoio para estas maneiras de se realizar a evolução.

O resultado da classificação dos estudos aponta uma grande quantidade de estudos voltados para a área de evolução de banco de dados e principalmente de seus esquemas, apresentando abordagens, ferramentas e *frameworks*. Com isto, pode-se inferir que o problema é real e antigo dentro do cenário de desenvolvimento, e que não apresenta ainda uma solução totalmente satisfatória. Esta afirmação pode ser apoiada por meio do gráfico presente na Figura 5, em que estudos antigos são apresentados, juntamente com estudos atuais (do ano de 2019 quando realizando primeira etapa do mapeamento). Isto ocorre devido ao surgimento de novas tecnologias que necessitam de uma entrega mais rápida do produto de forma contínua como foi apresentado em alguns artigos no formato de tendências de estudos.

A IC e o DC, apresentou a necessidade no cenário atual de formas automatizadas que possibilitem a evolução do banco de dados, ou apoiar alguma etapa muito complexa desta atividade. Esta é considerada a principal tendência atual, alguns estudos já começaram a trabalhar nesta parte, entretanto se mostraram limitados às tecnologias ou banco de dados específicos.

3.4.2 Ameaças à Validade

Um ponto importante a ser destacado, é decorrente à falta de especificação da diferença entre as categorias apresentadas na Figura 6. Sabendo disto, é possível que os estudos possam pertencer a mais de uma categoria ou use algum termo errado o que impactaria na sua classificação. Outro ponto importante é que possível que as análises e categorização dos estudos sejam afetadas de acordo com os nossos interesses, para mitigar

isto foram utilizados os critérios de inclusão e exclusão durante a etapa de seleção o que resultou em estudos conflitantes entre os envolvidos. Outra ameaça é que os termos foram definidos com base em nossa experiência e investigação dentro do cenário de evolução de banco de dados o que não podemos afirmar que todos os sinônimos foram utilizados. Um fator que também pode impactar no resultado são as *strings* de busca, decorrente a algumas bases apresentarem limitações ou falta de mecanismos para otimizar a busca.

Uma ameaça à validade interna seria o viés de publicação referente às formas de automatização, processos ou demais formas que apoiassem o uso de evolução de banco de dados na IC e EC de *software*. Outra ameaça seria a limitação de banco de dados atuais, eliminando estudos que não utilizam banco de dados geralmente utilizados, com isto pode ter algum modo de auxílio de evolução não investigado. Entretanto, estes estudos deveriam ser adaptados para o contexto de outros bancos de dados, todos os estudos que apresentavam a possibilidade de evolução de bancos de dados geralmente utilizados e bancos mais antigos foram aceitos.

Como ameaça a validade de construção é possível salientar a exclusão de algum estudo que apresentassem formas de realizar ou apoiar a evolução, para isto foram utilizadas regras de busca e fases bem definidas de exclusão e inclusão com diversos filtros. Além de ter realizado todo o processo com mais envolvidos. Já como ameaça externa a validade é possível apresentar que não foram levados em consideração possíveis mapeamentos na área de evolução de *software*, tendo em vista que tem objetivos diferentes. Isto acontece já que nenhum mapeamento ou estudo comparativo como revisões foram encontradas para este contexto específico.

Além das ameaças anteriores a demora para finalização deste mapeamento bem como seus resultados faz com que os resultados não estejam atualizados. Dessa forma para uma análise atualizada será necessário realizar uma nova pesquisa.

4 GUIDELINE

4.1 Problema Introduzido

Para serem inseridos na temática de evolução de banco de dados foi apresentado um problema para os alunos da disciplina de Resolução de Problemas V no semestre 2021/1. Tal problema foi inspirado em (TITE, 2014).

Na ilustração do problema apresenta-se um banco de dados que possui duas tabelas, a tabela Usuário e a tabela Empregado. A ligação das tabelas se dá por meio de um relacionamento um para um (1:1). Assumindo que o sistema já está implantado e em execução, parte-se da necessidade de modificação tanto no esquema do banco de dados como nos dados. Para tal, ilustra-se uma estratégia de evolução que é recomendada pela literatura.

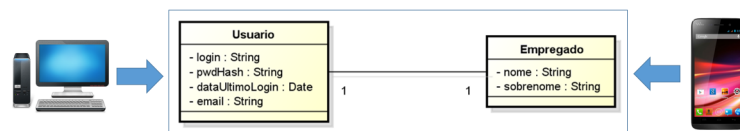


Figura 7 – Problema Motivante

No exemplo, assuma que o time de desenvolvimento percebeu a necessidade de que apenas alguns empregados tenham dados de usuário. Isso poderia ser resolvido com uma informação nula para a chave estrangeira que consta num ou noutro lado da relação entre as entidades. No entanto, é uma regra da empresa contratante que todo empregado deve ter um e-mail. Neste caso, é preciso adotar uma estratégia para evolução tanto do esquema como dos dados para mover o e-mail da tabela Usuário para a Empregado.

Uma possível estratégia é realizar os seguintes passos:

- 1: Planejar um *refactoring* do banco de dados para controlar a mudança.
- 2: Adicionar a coluna email na tabela Empregado, desnormalizando o banco, já que o atributo email constará nas duas tabelas.
- 3: Criar uma *trigger* para replicar o dado de email da tabela Usuário para a tabela Empregado.
- 4: Depois de seis meses, por exemplo, quando o banco estiver estável, remover a *trigger* e a coluna email da tabela Usuário normalizando o banco.

Este tipo de problema é bastante interessante para professores guiarem os alunos no exercício de situações reais de evolução de banco de dados, independentemente do domínio de aplicação que venha à ser utilizado em disciplinas como Resolução de Problemas V. Para tal, deve ser considerado pelos alunos as estratégias associadas com as mudanças que serão aplicadas diretamente no banco de dados implantado. Dessa forma, os alunos podem ser desafiados para encontrar qual seria a solução para aplicar essa alteração.

Apesar do problema motivante apresentado por (TITE, 2014) ser bastante interessante, a sua apresentação é limitada para justificar a utilização de três bancos de dados

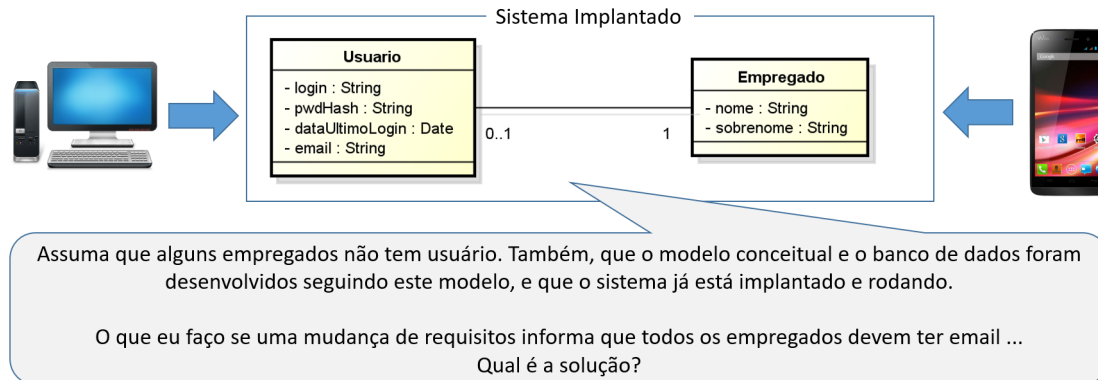


Figura 8 – Sistema Implantado

em sincronia com processos de entrega contínua. Logo, percebe-se a necessidade de um *guideline* para ser utilizado por alunos e professores para a melhor problematização e traçamento de estratégias de evolução de banco de dados que considere um universo maior de alternativas, como as encontradas pela revisão sistemática da literatura.

4.2 *Guideline*

Para que ocorra com segurança alteração no banco de dados foi proposta a utilização do *Guideline* desenvolvido a partir deste trabalho, incluído no Apêndice A. O restante do material de apoio consta todo no Moodle da disciplina¹ no **Marco 3 - Processos Assistidos por Ferramental**. Para que fosse realizada a evolução de banco de dados, os alunos deveriam seguir o passo a passo:

Passo 1: Identificar qual o tipo de SGBDs que iriam adotar.

Passo 2: Identificar quais os problemas de evolução de banco de dados que iriam enfrentar ao utilizar tal SGBDs.

Passo 3: Adaptar o processo base para problemas específicos de evolução de banco de dados a fim de identificar atividades e suporte ferramental.

Passo 4: Identificar os possíveis beneficiários ao realizar a evolução de banco de dados.

Passo 5: Identificar os possíveis malefícios na evolução de banco de dados

Passo 6: Identificar possíveis inovações para introduzir na organização

Além do passo a passo foi disponibilizado para os alunos video explicativo sobre evolução de banco, sua importância e os critérios de qualidade. Bem como os resultados das tabelas 8, 9 e 7.

¹ RP V - <<https://moodle.unipampa.edu.br/moodle/course/view.php?id=14916>>

4.3 Resultado da Aplicação

O *guideline* foi colocado à prova durante a disciplina de Resolução de Problemas 5 no semestre 2021/1. Devido ao problema de evolução de BD e ao *guideline* serem apresentados aos alunos no último marco, ou seja, no fim do semestre, e também de sua devolutiva quanto ao preenchimento de formulários elaborados no *Google Forms* não ser obrigatória, uma avaliação rigorosa não foi viável. Estes entraves acarretaram em poucos resultados quantitativos, o que atrapalha o desenho de conclusões acerca de benefícios do *guideline* para os alunos.

No lugar, com o objetivo de identificar elementos de evolução que foram adotados, foram analisados os relatórios dos quatro grupos de RP V. Porém, o retorno também não foi suficientemente relevante para ser apresentado neste trabalho de modo quantitativo.

Entretanto, houveram *feedbacks* positivos dos alunos em relação a ideia de problemática e ao próprio *guideline*. Os alunos levantaram que tal prática é interessante mas deve ser inserida no início da disciplina para que com o passar do semestre possam desenvolver a evolução do banco com tempo para aprender e praticar. Isso também foi corroborado na apresentação dos grupos no último marco da disciplina.

Uma observação interessante do professor ministrante da disciplina é que muitos alunos não acompanharam o material escrito para realizar algumas atividades associadas com a evolução de BD. Em vez disso, optaram por outras fontes de informação, como vídeos no Youtube ou dicas de colegas que já cursaram a disciplina. Isso é importante, pois demonstra que, além do *guideline*, material didático em formato de vídeo poderia transmitir conhecimento sobre evolução de BDs.

No entanto, as práticas relatadas pelos grupos que não adotaram o *guideline* são restritas para a evolução de esquema de dados. Ou seja, limitaram-se ao uso de APIs para o versionamento dos *scripts* SQL. Como observado na revisão de literatura, existem vários outros problemas além da evolução do esquema de banco que deveriam ser exploradas. Assim, recomenda-se para as próximas execuções que os alunos exercitem alternativas de *frameworks* e estratégias de evolução de dados, bem como ferramentas que lhes permitam realizar tais evoluções de modo automático.

5 CONSIDERAÇÕES FINAIS

Com base nos artigos analisados é possível concluir que ainda existem alguns *gaps* de pesquisa em diversos campos dentro da evolução de banco de dados. Como por exemplo o limite entre os termos a serem utilizados, quando é caracterizado como refatoração, reengenharia ou até mesmo evolução de banco de dados ou de esquema. Além disso outros problemas foram detectados como a falta de estudos direcionados a algumas áreas visto que a maioria dos estudos está direcionado apenas para a evolução de esquema em si.

Entretanto, ainda é uma área de pesquisa que possui diversas abordagens e ferramentas para realização e apoio à evolução. Também se observa que, na maioria dos casos, isso torna o uso das ferramentas difícil, que pode ser decorrente da falta de avaliações em ambientes de larga escala. Um ponto encontrado também é relacionado à falta de estudos que apresentem alguma abordagem, ferramenta ou *guidelines* generalizáveis. Ou seja, que não sejam vinculadas à alguma tecnologia de SGBD e de linguagem de programação. Portanto, isso limita o uso das propostas existentes e pode acarretar em desuso devido à depreciação da tecnologia dependente.

Diante da aplicação do *guideline* na disciplina de Resolução de Problemas V e que seu resultado não foi o esperado é necessário que seja abordada uma nova maneira de realizar essa aplicação visto que alguns alunos não consultavam o material teórico mas sim materiais da internet, minitutoriais ou até mesmo a ajuda de colegas. A explicação de novos métodos para serem utilizadas estão na Seção 5.1.

Mesmo a evolução de software sendo algo cotidiano nas empresas, ela ainda é um assunto pouco abordado. Esse assunto sempre está presente em conversas e teorias, mas na prática ainda não se aplica. Ainda hoje, empresas multinacionais sofrem para realizar alterações em banco. Tais alterações acabam demandando muitos processos e pessoas para que sejam realizadas acarretando em uma perda de produtividade. Dessa forma tanto para a área acadêmica quanto para a área da indústria é de grande importância que tenham *guidelines* e processos que facilitem essas evoluções.

5.1 Trabalhos Futuros

Como trabalho futuro deverá ser realizada a aplicação da ideia do *guideline*, mas de uma nova maneira na próxima edição da disciplina de Resolução de Problemas V. Inicialmente pode ser realizado a criação de mini tutoriais para inserir a parte de evolução de banco já ao início dos marcos da disciplina. Com o passar do tempo, pode-se ir acompanhando como acontece as alterações do banco a medida que o software evolui para acompanhar novos requisitos. Também, acompanhar a normalização dos dados após a aplicação de técnicas de *Extract Transform Loads* (ETLs) e decorrente alteração nos esquemas e dados conforme avanço do software para melhorar o software desenvolvido. Neste contexto de problematização, seria explorado na disciplina mais o que acontece

realmente na indústria, visto que existem milhares de software que já tem seus bancos a décadas e partir de suas novas funcionalidades ou alterações acaba sendo necessário que seus bancos evoluam linearmente.

Outra atividade para trabalho futuro seria a criação de um meta-processo, visto que com esse estudo pode ser direcionado para várias frentes de utilização, sejam elas em relação ao tipo de banco utilizado e a metodologia que será utilizada para lidar com situações de estratégias de evolução de esquema e dados. Dessa forma, este trabalho pode passar a ter uma continuidade até com demais alunos que se interessem pela área e queiram investigá-la após a realização da disciplina de RP V.

REFERÊNCIAS

- AGARWAL, R.; SARANGI, A.; DAS, S. Reengineering of database intensive application. **SIGSOFT Softw. Eng. Notes**, v. 28, p. 1–1, 2003. Citado na página 93.
- AND; AND. Design and implementation of database schema evolution for service continuity of web-based internet applications. In: **14th Asia-Pacific Network Operations and Management Symposium (APNOMS)**. 2012: [s.n.]. p. 1–4. Citado 2 vezes nas páginas 45 e 89.
- ANDANY, J.; L'EONARD, M.; PALISSER, C. Management of schema evolution in databases. In: **Proceedings of the 17th International Conference on Very Large Data Bases**. [S.l.]: VLDB '91, 1991. p. 161–170. ISBN 1-55860-150-3. Citado na página 93.
- ARAÚJO, M. A. P.; TRAVASSOS, G. H. Em busca de um framework para estudos experimentais em evolução de software. **COPPE/UFRJ**, 2004. Citado na página 32.
- ARIAV, G. Temporally oriented data definitions: Managing schema evolution in temporally oriented databases. **Data \ Knowledge Engineering**, v. 6, p. 451 – 467, 1991. Citado na página 90.
- ASSOUDI, H.; LOUNIS, H. Self-healing data exchange process under evolving schemas: A new mapping adaptation approach based on self-optimization. In: . [S.l.: s.n.], 2011. p. 188 – 190. Citado 2 vezes nas páginas 46 e 93.
- ATHINAIIOU, C.; KONDYLAKIS, H. Vesel: Visual exploration of schema evolution using provenance queries. In: . [S.l.: s.n.], 2019. v. 2322. Citado 2 vezes nas páginas 47 e 92.
- AULBACH, S. et al. Extensibility and data sharing in evolving multi-tenant databases. In: . [S.l.: s.n.], 2011. p. 99–110. Citado na página 93.
- BANERJEE, S.; DAVIS, K. C. Modeling data warehouse schema evolution over extended hierarchy semantics. In: . [S.l.: s.n.], 2009. v. 5530 LNCS, p. 72 – 96. Citado 2 vezes nas páginas 46 e 92.
- BAQASAH, A.; PARDEDE, E. Managing schema evolution in hybrid xml-relational database systems. In: . [S.l.: s.n.], 2010. p. 455 – 460. Citado na página 92.
- BAEKGAARD, L. Transaction-based specification of database evolution. In: **Conceptual Modeling — ER '97**. [S.l.: s.n.], 1997. v. 1331, p. 127–140. ISBN 978-3-540-63699-1 978-3-540-69630-8. Citado 2 vezes nas páginas 47 e 89.
- BELLAHSENE, Z. Extending a view mechanism to support schema evolution in federated database systems. In: **Database and Expert Systems Applications**. [S.l.: s.n.], 1997. v. 1308, p. 573–582. ISBN 978-3-540-63478-2 978-3-540-69580-6. Citado 2 vezes nas páginas 46 e 89.
- BELLAHSENE, Z. Schema evolution in data warehouses. **Knowledge and Information Systems**, v. 4, p. 283–304, 2002. Citado 2 vezes nas páginas 46 e 89.
- BLASCHKA, M.; SAPIA, C.; HOFLING, G. On schema evolution in multidimensional databases. In: . [S.l.: s.n.], 1999. v. 1676, p. 153 – 164. Citado na página 91.

BOBROWSKI, S. **Oracle 7 Server Concepts**. [S.l.]: Oracle Corporation, 1996. Citado na página 32.

BOMMEL, P. van. Experiences with edo: An evolutionary database optimizer. **Data and Knowledge Engineering**, v. 13, p. 243–263, 1994. Citado 2 vezes nas páginas 46 e 93.

BOSCARIOLI, C. et al. Uma reflexão sobre banco de dados orientados a objetos. In: **Congresso de Tecnologias para Gestão de Dados e Metadados do Cone Sul, Paraná, Brasil**. [S.l.: s.n.], 2006. Citado na página 31.

BOSSUNG, S. et al. Open and dynamic schema evolution in content-intensive web applications. In: . [S.l.: s.n.], 2006. IT/WIA, p. 109 – 116. Citado na página 91.

BOUNIF, H.; POTTINGER, R. Schema repository for database schema evolution. In: . [S.l.: s.n.], 2006. p. 647 – 651. Citado 2 vezes nas páginas 46 e 92.

BOUNIF, H.; SPACCAPIETRA, S.; POTTINGER, R. Requirements ontology and multi-representation strategy for database schema evolution. In: . [S.l.: s.n.], 2007. v. 4623 LNCS, p. 68 – 84. Citado na página 91.

BRUMMERMANN, H.; KEUNECKE, M.; SCHMID, K. Managing the evolution and customization of database schemas in information system ecosystems. **Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)**, v. 7908 LNCS, p. 417–432, 2013. Citado 2 vezes nas páginas 46 e 93.

CARUCCIO, L.; POLESE, G.; TORTORA, G. Synchronization of queries and views upon schema evolutions: A survey. **ACM Transactions on Database Systems**, v. 41, 2016. Citado na página 90.

CARUCCIO, L.; POLESE, G.; TORTORA, G. Dependency-based query/view synchronization upon schema evolutions. In: . [S.l.: s.n.], 2018. v. 11158 LNCS, p. 91 – 105. Citado na página 91.

CASTELLI, D. Three levels of reuse for supporting the evolution of database schemas. **Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)**, v. 1728, p. 1–16, 1999. Citado na página 93.

CERICOLA, O. V. **Banco de dados relacional e distribuído**. [S.l.]: Livros Tecnicos e Cientificos Editora, 1991. Citado na página 30.

CERÍCOLA, V. O. Oracle: Banco de dados relacional e distribuído. **Ferramentas para desenvolvimento**. São Paulo: Makron–McGraw-Hill, 1995. Citado na página 29.

CHANG, S.-K. et al. A logic framework to support database refactoring. In: . [S.l.: s.n.], 2007. v. 4653 LNCS, p. 509 – 518. Citado 2 vezes nas páginas 45 e 92.

CHEN, J. L.; MCLEOD, D.; O’LEARY, D. Schema evolution for object-based accounting database systems. In: **Object-Oriented Methodologies and Systems**. [S.l.: s.n.], 1994. v. 858, p. 40–52. ISBN 978-3-540-58451-3 978-3-540-48804-0. Citado 2 vezes nas páginas 46 e 89.

- CHEN, J.-L.; MCLEOD, D.; O'LEARY, D. Domain-knowledge-guided schema evolution for accounting database systems. **Expert Systems with Applications**, v. 9, p. 491 – 501, 1995. Citado 2 vezes nas páginas 46 e 90.
- CHIAO, C. M.; KUNZLE, V.; REICHERT, M. To wards schema evolution in object-aware process management systems. In: . [S.l.: s.n.], 2014. P-234, p. 101 – 115. Citado na página 91.
- CLAMEN, S. M. Schema evolution and integration. **Distributed and Parallel Databases**, v. 2, p. 101–126, 1994. Citado na página 89.
- CLAYPOOL, K. T.; NATARAJAN, C.; RUNDENSTEINER, E. A. Optimizing performance of schema evolution sequences. In: . [S.l.: s.n.], 2001. v. 1944, p. 114 – 127. Citado na página 91.
- CLEVE, A.; BROGNEAUX, A.-F.; HAINAUT, J.-L. A conceptual approach to database applications evolution. **Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)**, v. 6412 LNCS, p. 132–145, 2010. Citado 2 vezes nas páginas 45 e 93.
- CLEVE, A. et al. Understanding database schema evolution: A case study. **Science of Computer Programming**, v. 97, p. 113 – 121, 2015. Citado na página 90.
- CLEVE, A.; HAINAUT, J.-L. Co-transformations in database applications evolution. In: . [S.l.: s.n.], 2006. v. 4143 LNCS, p. 409 – 421. Citado na página 90.
- CONNOR, R. et al. Using persistence technology to control schema evolution. In: . [S.l.: s.n.], 1994. Part F129433, p. 441 – 446. Citado na página 91.
- COOX, S. Axiomatization of the evolution of xml database schema. **Programming and Computer Software**, v. 29, p. 140 – 146, 2003. Citado 2 vezes nas páginas 45 e 92.
- CURINO, C. et al. Automating the database schema evolution process. **The VLDB Journal**, v. 22, p. 73–98, 2013. Citado 2 vezes nas páginas 45 e 89.
- CURINO, C.; MOON, H. J.; ZANIOLO, C. Automating database schema evolution in information system upgrades. In: . [S.l.: s.n.], 2009. p. ACM Special Interest Group on Programming Languages, SIGPLAN –. Citado 2 vezes nas páginas 45 e 92.
- CURINO, C. A. et al. Update rewriting and integrity constraint maintenance in a schema evolution support system: Prism++. **Proceedings of the VLDB Endowment**, v. 4, p. 117 – 128, 2010. Citado 2 vezes nas páginas 47 e 92.
- CURINO, C. A. et al. The prism workbench: Database schema evolution without tears. In: . [S.l.: s.n.], 2009. p. 1523 – 1526. Citado 2 vezes nas páginas 47 e 92.
- CURINO, C. A. et al. Schema evolution in wikipedia - toward a web information system benchmark. In: . [S.l.: s.n.], 2008. DISI, p. 323 – 332. Citado na página 92.
- CURINO, C. A.; MOON, H. J.; ZANIOLO, C. Graceful database schema evolution: The prism workbench. **Proceedings of the VLDB Endowment**, v. 1, p. 761 – 772, 2008. Citado 2 vezes nas páginas 46 e 91.

- CURINO, C. A.; MOON, H. J.; ZANIOLO, C. Managing the history of metadata in support for db archiving and schema evolution. In: . [S.l.: s.n.], 2008. v. 5232 LNCS, p. 78 – 88. Citado 2 vezes nas páginas 46 e 91.
- DALCIN, E. A informática no inventário e monitoramento da arborização urbana. In: **Congresso Brasileiro Sobre Arborização Urbana**. [S.l.: s.n.], 1994. v. 2, p. 201–206. Citado na página 30.
- DATE, C. J. **Introdução a sistemas de bancos de dados**. [S.l.]: Elsevier Brasil, 2004. Citado 2 vezes nas páginas 29 e 32.
- DELGADO, C.; SAMOS, J.; TORRES, M. Primitive operations for schema evolution in odmng databases. In: . [S.l.: s.n.], 2003. v. 2817, p. 226 – 237. Citado na página 91.
- DELPLANQUE, J. et al. Relational database schema evolution: An industrial case study. In: . [S.l.: s.n.], 2018. p. 635 – 644. Citado na página 92.
- DOMÍNGUEZ, E. et al. Medea: A database evolution architecture with traceability. **Data \ Knowledge Engineering**, v. 65, p. 419 – 441, 2008. Citado 2 vezes nas páginas 46 e 90.
- DOMINGUES, H. H.; KON, F.; FERREIRA, J. E. Asynchronous replication for evolutionary database development: A design for the experimental assessment of a novel approach. In: . [S.l.: s.n.], 2011. v. 7045 LNCS, p. 818 – 825. Citado na página 92.
- DOMINGUES, M. B. P. et al. A workflow for database refactoring. **International Journal of Innovative Computing, Information and Control**, v. 10, p. 2209 – 2220, 2014. Citado 2 vezes nas páginas 45 e 92.
- DOMINGUES, M. B. P. et al. Refactoring database to improve queries performance. In: . [S.l.: s.n.], 2013. p. 159 – 166. Citado na página 90.
- DOMINGUEZ, E. et al. Model-driven, view-based evolution of relational databases. In: . [S.l.: s.n.], 2008. v. 5181 LNCS, p. 822 – 836. Citado 2 vezes nas páginas 46 e 92.
- DRAHEIM, D.; HORN, M.; SCHULZ, I. The schema evolution and data migration framework of the environmental mass database imis. In: **Proceedings. 16th International Conference on Scientific and Statistical Database Management, 2004**. 2004: [s.n.]. p. 341–344. Citado 2 vezes nas páginas 47 e 89.
- EWALD, C. A.; ORLOWSKA, M. E. A procedural approach to schema evolution. In: **Active Flow and Combustion Control 2018**. [S.l.: s.n.], 1993. v. 141, p. 22–38. ISBN 978-3-319-98176-5 978-3-319-98177-2. Citado na página 89.
- FAISAL, S. et al. Temporal and evolving data warehouse design. **Scientific Programming**, v. 2017, 2017. Citado na página 93.
- FAN, H.; POULOVASSILIS, A. Schema evolution in data warehousing environments a schema transformation-based approach. **Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)**, v. 3288, p. 639–653, 2004. Citado 2 vezes nas páginas 46 e 93.

- FAVRE, C.; BENTAYEB, F.; BOUSSAID, O. Evolution of data warehouses' optimization: A workload perspective. In: . [S.l.: s.n.], 2007. v. 4654 LNCS, p. 13 – 22. Citado na página 93.
- FIADEIRO, J.; SERNADAS, A. Specification and verification of database dynamics. **Acta Informatica**, v. 25, p. 625–661, 1988. Citado na página 93.
- FIGUEIREDO, I. C. de A.; MARTINS, D. M. S. Gerenciamento de mudanças em banco de dados: Um estudo de caso utilizando flyway. **Caderno de Estudos em Sistemas de Informação**, v. 5, n. 1, 2018. Citado na página 24.
- FILATOV, V.; SEMENETS, V. Methods for synthesis of relational data model in information systems reengineering problems. In: **International Scientific-Practical Conference Problems of Infocommunications. Science and Technology (PIC S T)**. [S.l.: s.n.], 2018. p. 247–251. Citado na página 89.
- FILHO, M. A. P. M. Sql x nosql: Análise de desempenho do uso do mongodb em relação ao uso do postgresql. 2015. Citado na página 30.
- FOSTER, J. C. The evolution of an integrated data base. In: **Proceedings of the 12th Design Automation Conference**. [S.l.]: DAC '75, 1975. p. 394–398. Citado na página 93.
- GALL, H. et al. Software evolution observations based on product release history. In: IEEE. **Proceedings International Conference on Software Maintenance**. [S.l.], 1997. p. 160–166. Citado 2 vezes nas páginas 23 e 85.
- GARCIA, J.; DIAZ, O.; CABOT, J. An adapter-based approach to co-evolve generated sql in model-to-text transformations. In: . [S.l.: s.n.], 2014. v. 8484 LNCS, p. 518 – 532. Citado na página 45.
- GOBERT, M. et al. Understanding schema evolution as a basis for database reengineering. In: . [S.l.: s.n.], 2013. p. 472 – 475. Citado na página 92.
- GODFREY, M. W.; GERMAN, D. M. The past, present, and future of software evolution. In: IEEE. **Frontiers of Software Maintenance**. [S.l.], 2008. p. 129–138. Citado na página 32.
- GORALWALLA, I. A. et al. Managing schema evolution using a temporal object model. In: **Conceptual Modeling — ER '97**. [S.l.: s.n.], 1997. v. 1331, p. 71–84. ISBN 978-3-540-63699-1 978-3-540-69630-8. Citado na página 89.
- GOSAIN, A.; SAROHA, K. Storage structure for handling schema versions in temporal data warehouses. In: . [S.l.: s.n.], 2018. v. 518, p. 501 – 511. Citado na página 91.
- GOTZ, S.; KUHN, T. Models@run.time for object-relational mapping supporting schema evolution. In: . [S.l.: s.n.], 2015. v. 1474, p. 41 – 50. Citado 2 vezes nas páginas 46 e 91.
- GROLINGER, K.; CAPRETZ, M. A. A unit test approach for database schema evolution. **Information and Software Technology**, v. 53, p. 159 – 170, 2011. Citado 2 vezes nas páginas 45 e 90.

GUEDES, G. B.; BAIOCO, G. B.; MORAES, R. L. de O. Evolutionary database design: Enhancing data abstraction through database modularization to achieve graceful schema evolution. In: . [S.l.: s.n.], 2016. v. 9827 LNCS, p. 355 – 369. Citado 2 vezes nas páginas 46 e 91.

HAINAUT, J. L. et al. Database evolution: the db-main approach. In: **Entity-Relationship Approach — ER '94 Business Modelling and Re-Engineering**. [S.l.: s.n.], 1994. v. 881, p. 112–131. ISBN 978-3-540-58786-6 978-3-540-49100-2. Citado 2 vezes nas páginas 45 e 89.

HAMAJI, K.; NAKAMOTO, Y. Toward a database refactoring support tool. In: . [S.l.: s.n.], 2016. p. 443 – 446. Citado na página 92.

HENRARD, J. et al. Large-scale data reengineering: Return from experience. In: **15th Working Conference on Reverse Engineering**. [S.l.: s.n.], 2008. p. 305–308. Citado na página 89.

HERRAIZ, I. et al. The evolution of the laws of software evolution: A discussion based on a systematic literature review. **ACM Computing Surveys (CSUR)**, ACM, v. 46, n. 2, p. 28, 2013. Citado na página 32.

HERRMANN, K. et al. Database evolution for software product lines. In: . [S.l.: s.n.], 2015. p. 125 – 133. Citado 2 vezes nas páginas 45 e 90.

HERRMANN, K. et al. Codel a relationally complete language for database evolution. In: . [S.l.: s.n.], 2015. v. 9282, p. 63 – 76. Citado 2 vezes nas páginas 45 e 90.

HERRMANN, K. et al. Living in parallel realities - co-existing schema versions with a bidirectional database evolution language. In: . [S.l.: s.n.], 2017. Part F127746, p. 1101 – 1116. Citado na página 90.

HERRMANN, K. et al. Robust and simple database evolution. **Information Systems Frontiers**, v. 20, p. 45 – 61, 2018. Citado 2 vezes nas páginas 46 e 90.

HICK, J.-M.; HAINAUT, J.-L. Strategy for database application evolution: The db-main approach. In: . [S.l.: s.n.], 2003. v. 2813, p. 291 – 306. Citado 2 vezes nas páginas 46 e 90.

HSU, C. A decision support system for database evolution using data model independent architecture. **Computers \ Operations Research**, v. 13, p. 427 – 436, 1986. Citado 2 vezes nas páginas 45 e 90.

HUO, W.; TSOTRAS, V. J. Querying transaction-time databases under branched schema evolution. In: . [S.l.: s.n.], 2012. v. 7446 LNCS, p. 265 – 280. Citado na página 92.

IMPACTA, R. **Entenda a importância de um banco de dados em uma organização**. 2017. Citado 2 vezes nas páginas 23 e 85.

JENSEN, O. G.; BOHLEN, M. Current, legacy, and invalid tuples in conditionally evolving databases. In: . [S.l.: s.n.], 2002. v. 2457, p. 65 – 82. Citado na página 90.

JENSEN, O. G.; BOHLEN, M. H. Lossless conditional schema evolution. In: . [S.l.: s.n.], 2004. v. 3288, p. 610 – 623. Citado 2 vezes nas páginas 46 e 91.

- JENSEN, O. G.; BOHLEN, M. H. Multitemporal conditional schema evolution. In: . [S.l.: s.n.], 2004. v. 3289, p. 441 – 454. Citado 2 vezes nas páginas 46 e 91.
- JIANFENG, Y.; BO, Z. Support multi-version applications in saas via progressive schema evolution. In: . [S.l.: s.n.], 2009. p. 1717 – 1724. Citado na página 92.
- JONG, M. D.; DEURSEN, A. V. Continuous deployment and schema evolution in sql databases. In: . [S.l.: s.n.], 2015. p. 16 – 19. Citado 2 vezes nas páginas 45 e 92.
- JONG, M. D.; DEURSEN, A. V.; CLEVE, A. Zero-downtime sql database schema evolution for continuous deployment. In: . [S.l.: s.n.], 2017. p. 143 – 152. Citado 2 vezes nas páginas 47 e 92.
- KAAS, C. E.; PEDERSEN, T. B.; RASMUSSEN, B. D. Schema evolution for stars and snowflakes. In: . [S.l.: s.n.], 2004. p. 425 – 433. Citado na página 91.
- KAUFELD, J.; LUDERMIR, J. **Access 95 para Windows para leigos**. [S.l.]: Berkeley, 1996. Citado na página 30.
- KITCHENHAM, B.; CHARTERS, S. Guidelines for performing systematic literature reviews in software engineering. Citeseer, 2007. Citado 3 vezes nas páginas 26, 33 e 35.
- KLETTKE, M. et al. Nosql schema evolution and big data migration at scale. In: **IEEE International Conference on Big Data (Big Data)**. [S.l.: s.n.], 2016. p. 2764–2774. Citado 2 vezes nas páginas 46 e 89.
- LAMMARI, N.; AKOKA, J.; COMYN-WATTIAU, I. Supporting database evolution: Using ontologies matching. In: . [S.l.: s.n.], 2003. v. 2817, p. 284 – 288. Citado 2 vezes nas páginas 46 e 91.
- LAUKKANEN, E.; ITKONEN, J.; LASSENIUS, C. Problems, causes and solutions when adopting continuous delivery—a systematic literature review. **Information and Software Technology**, Elsevier, v. 82, p. 55–79, 2017. Citado 3 vezes nas páginas 24, 25 e 85.
- LEHMAN, M. M. Laws of software evolution revisited. In: SPRINGER. **European Workshop on Software Process Technology**. [S.l.], 1996. p. 108–124. Citado na página 32.
- LEITE, J. Gerenciando a qualidade de software com base em requisitos. **Qualidade de software: teoria e prática**. São Paulo: Prentice Hall, p. 238–246, 2001. Citado na página 32.
- LERNER, B. S. A model for compound type changes encountered in schema evolution. **ACM Transactions on Database Systems**, v. 25, p. 83 – 127, 2000. Citado 2 vezes nas páginas 45 e 92.
- LIM, E.-P.; CHIANG, R. Accommodating instance heterogeneities in database integration. **Decision Support Systems**, v. 38, p. 213–231, 2004. Citado na página 93.
- LIN, D.-Y.; NEAMTIU, I. Collateral evolution of applications and databases. In: . [S.l.: s.n.], 2009. p. 31 – 40. Citado na página 90.

- LIU, C.; CHANG, S.; CHRYSANTHIS, P. Database schema evolution using ever diagrams. In: . [S.l.: s.n.], 1994. p. 123 – 123. Citado 2 vezes nas páginas 45 e 92.
- LIU, C. T.; CHANG, S. K.; CHRYSANTHIS, P. K. An entity-relationship approach to schema evolution. In: **Proceedings of ICCI'93: 5th International Conference on Computing and Information**. 1993: [s.n.]. p. 575–578. Citado 2 vezes nas páginas 45 e 89.
- LIU, C.-T.; CHRYSANTHIS, P. K.; CHANG, S.-K. Database schema evolution through the specification and maintenance of changes on entities and relationships. In: **Entity-Relationship Approach — ER '94 Business Modelling and Re-Engineering**. [S.l.: s.n.], 1994. v. 881, p. 132–151. ISBN 978-3-540-58786-6 978-3-540-49100-2. Citado 2 vezes nas páginas 45 e 89.
- LIU, Z. et al. Cods: Evolving data efficiently and scalably in column oriented databases. **Proceedings of the VLDB Endowment**, v. 3, p. 1521 – 1524, 2010. Citado 2 vezes nas páginas 45 e 90.
- LORENZ, M.; HESSE, G.; RUDOLPH, J.-P. Object-relational mapping revised-a guideline review and consolidation. In: **ICSOFTEA**. [S.l.: s.n.], 2016. p. 157–168. Citado 2 vezes nas páginas 23 e 103.
- LÓSCIO, B. F.; OLIVEIRA, H. d.; PONTES, J. d. S. Nosql no desenvolvimento de aplicações web colaborativas. **VIII Simpósio Brasileiro de Sistemas Colaborativos**, v. 10, n. 1, p. 11, 2011. Citado na página 30.
- LU, J. Reengineering of database applications to ejb based architecture. **Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)**, v. 2348, p. 361–376, 2002. Citado na página 93.
- LUCIV, D. Deductive approach to semistructured schema evolution. In: . [S.l.: s.n.], 2005. v. 154. Citado na página 91.
- MAATUK, A.; ALI, M.; ROSSITER, N. Re-engineering relational databases: The way forward. In: . [S.l.: s.n.], 2011. Citado na página 93.
- MACEK, O.; RICHTA, K. Application and relational database co-refactoring. **Computer Science and Information Systems**, v. 11, p. 503–524, 2014. Citado 2 vezes nas páginas 45 e 93.
- MANNISTO, T.; SULONEN, R. Evolution of schema and individuals of configurable products. In: . [S.l.: s.n.], 1999. v. 1727, p. 12 – 23. Citado na página 90.
- MANOUSIS, P. et al. Schema evolution for databases and data warehouses. In: . [S.l.: s.n.], 2016. v. 253, p. 1 – 31. Citado na página 92.
- MARCINIAK, J.; PANKOWSKI, T. Tracking changes in database schemas. **Advances in Intelligent Systems and Computing**, v. 183 AISC, p. 155 – 165, 2013. Citado na página 91.
- MAROTTA, A.; MOTZ, R.; RUGGIA, R. Managing source schema evolution in web warehouses. **Journal of the Brazilian Computer Society**, v. 8, p. 20 – 31, 2002. Citado na página 92.

- MASOOD, N.; ANDLEEB, N. A taxonomy of changes in schema evolution. In: **International Conference on Computational Science and Technology (ICCSST)**. [S.l.: s.n.], 2014. p. 1–6. Citado na página 89.
- MCKENZIE, E.; SNODGRASS, R. Schema evolution and the relational algebra. **Information Systems**, v. 15, p. 207 – 232, 1990. Citado na página 90.
- MEIMARIS, M.; PAPASTEFANATOS, G. The evogen benchmark suite for evolving rdf data. In: . [S.l.: s.n.], 2016. v. 1585, p. 20–35. Citado na página 93.
- MENS, T. Introduction and roadmap: History and challenges of software evolution. In: **Software evolution**. [S.l.]: Springer, 2008. p. 1–11. Citado na página 32.
- MEURICE, L.; CLEVE, A. Dahlia: A visual analyzer of database schema evolution. In: **Software Evolution Week - IEEE Conference on Software Maintenance, Reengineering, and Reverse Engineering (CSMR-WCRE)**. [S.l.: s.n.], 2014. p. 464–468. Citado 2 vezes nas páginas 45 e 89.
- MEURICE, L.; CLEVE, A. Supporting schema evolution in schema-less nosql data stores. In: **IEEE 24th International Conference on Software Analysis, Evolution and Reengineering (SANER)**. [S.l.: s.n.], 2017. p. 457–461. Citado 2 vezes nas páginas 46 e 89.
- MEURICE, L.; NAGY, C.; CLEVE, A. Detecting and preventing program inconsistencies under database schema evolution. In: **IEEE International Conference on Software Quality, Reliability and Security (QRS)**. [S.l.: s.n.], 2016. p. 262–273. Citado 2 vezes nas páginas 46 e 89.
- MINSKY, N.; ROZENSHTAIN, D.; CHOMICKI, J. A controllable prolog database system. In: **IEEE Second International Conference on Data Engineering**. [S.l.: s.n.], 1986. p. 618–628. Citado 2 vezes nas páginas 45 e 89.
- MOEHRKE, D. The evolution of data base management at a. o. smith. In: . [S.l.: s.n.], 1971. p. 299–323. Citado na página 93.
- MOON, H.; CURINO, C.; ZANIOLO, C. Scalable architecture and query optimization for transaction-time dbs with evolving schemas. In: . [S.l.: s.n.], 2010. p. 207–218. Citado na página 93.
- MOON, H. J. et al. Managing and querying transaction-time databases under schema evolution. **Proc. VLDB Endow.**, v. 1, p. 882–895, 2008. Citado na página 93.
- MOON, H. J. et al. Prima: Archiving and querying historical data with evolving schemas. In: . [S.l.: s.n.], 2009. p. 1019 – 1021. Citado 2 vezes nas páginas 46 e 91.
- MORAES, A. C.; SALGADO, A. C.; TEDESCO, P. A. Autonomouddb: A tool for autonomic propagation of schema updates in heterogeneous multi-database environments. In: . [S.l.: s.n.], 2009. p. 251 – 256. Citado 2 vezes nas páginas 45 e 90.
- MORO, M. M.; MALAIKA, S.; LIM, L. Preserving xml queries during schema evolution. In: . [S.l.: s.n.], 2007. p. 1341 – 1342. Citado 2 vezes nas páginas 46 e 91.
- MORZY, T.; WREMBEL, R. Modeling a multiversion data warehouse: A formal approach. In: . [S.l.: s.n.], 2003. v. 1, p. 120 – 127. Citado na página 90.

- MOTZ, R. Dynamic maintenance of an integrated schema. **Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)**, v. 3379 LNCS, p. 21 – 30, 2005. Citado na página 90.
- NEJI, M. et al. Multidimensional schema evolution: Integrating new olap requirements. In: . [S.l.: s.n.], 2006. DISI, p. 331 – 334. Citado 2 vezes nas páginas 46 e 92.
- NGUEZ, E. D. I.; LLORET, J.; ZAPATA, M. A. An architecture for managing database evolution. In: . [S.l.: s.n.], 2003. v. 2784, p. 63 – 74. Citado 2 vezes nas páginas 45 e 90.
- OVEREEM, M.; SPOOR, M.; JANSEN, S. The dark side of event sourcing: Managing data conversion. In: **IEEE 24th International Conference on Software Analysis, Evolution and Reengineering (SANER)**. [S.l.: s.n.], 2017. p. 193–204. Citado na página 89.
- PAPASTEFANATOS, G. et al. Hecataeus: A what-if analysis tool for database schema evolution. In: . [S.l.: s.n.], 2008. p. 326 – 328. Citado 2 vezes nas páginas 46 e 92.
- PAPASTEFANATOS, G. et al. Language extensions for the automation of database schema evolution. In: . [S.l.: s.n.], 2008. DISI, p. 74 – 81. Citado 2 vezes nas páginas 46 e 92.
- PAPASTEFANATOS, G. et al. Hecataeus: Regulating schema evolution. In: **IEEE 26th International Conference on Data Engineering (ICDE 2010)**. [S.l.: s.n.], 2010. p. 1181–1184. Citado 2 vezes nas páginas 46 e 89.
- PAPASTEFANATOS, G.; VASSILIADIS, P.; VASSILIOU, Y. Adaptive query formulation to handle database evolution. In: . [S.l.: s.n.], 2006. v. 231. Citado 2 vezes nas páginas 45 e 90.
- PETERSEN, K. et al. Systematic mapping studies in software engineering. In: **Ease**. [S.l.: s.n.], 2008. v. 8, p. 68–77. Citado 4 vezes nas páginas 15, 26, 33 e 35.
- POLESE, G.; VACCA, M. The envision project: Towards a visual tool to support schema evolution in distributed databases. In: . [S.l.: s.n.], 2009. p. 174 – 179. Citado na página 91.
- PROPER, H. Data schema design as a schema evolution process. **Data \ Knowledge Engineering**, v. 22, p. 159 – 189, 1997. Citado 2 vezes nas páginas 45 e 90.
- QIU, D.; LI, B.; SU, Z. An empirical analysis of the co-evolution of schema and code in database applications. In: . [S.l.: s.n.], 2013. p. 125 – 135. Citado na página 92.
- RA, Y.-G. Relational schema evolution for program independency. **Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)**, v. 3356, p. 273–281, 2004. Citado 2 vezes nas páginas 46 e 93.
- REDGATE. **Survey on DevOps adoption rates among SQL Server Professionals**. 2018. The 2018 State of Database DevOps. Citado 3 vezes nas páginas 23, 25 e 103.
- REITER, R. Formalizing database evolution in the situation calculus. In: . [S.l.: s.n.], 1992. p. 600 – 600. Citado na página 91.

- REITER, R. On specifying database updates. **The Journal of Logic Programming**, v. 25, p. 53 – 91, 1995. Citado na página 90.
- RISTIĆ, S. et al. Database reverse engineering based on meta-models. **Open Computer Science**, v. 4, 2014. Citado na página 89.
- RODDICK, J. Schema evolution in database systems: An annotated bibliography. **ACM SIGMOD Record**, v. 21, p. 35–40, 1992. Citado na página 93.
- RODDICK, J. Sql/se: A query language extension for databases supporting schema evolution. **ACM SIGMOD Record**, v. 21, p. 10–16, 1992. Citado na página 93.
- RODDICK, J.; VRIES, D. D. Reduce, reuse, recycle: Practical approaches to schema integration, evolution and versioning. **Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)**, v. 4231 LNCS, p. 209–216, 2006. Citado na página 93.
- SCALAS, M. R.; CAPPELLI, A.; CASTRO, C. D. A model for schema evolution in temporal relational databases. In: **CompEuro Proceedings Computers in Design, Manufacturing, and Production**. [S.l.: s.n.], 1993. p. 223–231. Citado 2 vezes nas páginas 45 e 89.
- SCHERZINGER, S.; CERQUEUS, T.; ALMEIDA, E. C. d. Controvol: A framework for controlled schema evolution in nosql application development. In: **IEEE 31st International Conference on Data Engineering**. 2015: [s.n.]. p. 1464–1467. Citado 2 vezes nas páginas 45 e 89.
- SCHERZINGER, S.; KLETTKE, M.; STORL, U. Cleager: Eager schema evolution in nosql document stores. In: . [S.l.: s.n.], 2015. v. 241, p. 659 – 662. Citado 2 vezes nas páginas 45 e 91.
- SCHERZINGER, S. et al. Datalution: A tool for continuous schema evolution in nosql-backed web applications. In: . [S.l.: s.n.], 2016. p. 38 – 39. Citado na página 91.
- SCHINK, H. sql-schema-comparer: Support of multi-language refactoring with relational databases. In: **IEEE 13th International Working Conference on Source Code Analysis and Manipulation (SCAM)**. [S.l.: s.n.], 2013. p. 173–178. Citado na página 89.
- SCHULER, R. E.; KESSELMAN, C. Towards an efficient and effective framework for the evolution of scientific databases. In: . [S.l.: s.n.], 2018. p. Alpin; EOS Solutions; Systems; Wurth Phoenix –. Citado 2 vezes nas páginas 47 e 90.
- SHARP, J. et al. Data access for highly-scalable solutions: Using sql, nosql, and polyglot persistence. **Microsoft patterns & practices**, 2013. Citado na página 31.
- SILBERSCHATZ, A.; SUNDARSHAN, S.; KORTH, H. F. **Sistema de banco de dados**. [S.l.]: Elsevier Brasil, 2016. Citado na página 29.
- SIMANOVSKY, A. A. Data schema evolution support in xml-relational database systems. **Programming and Computer Software**, v. 34, p. 16–26, 2008. Citado na página 89.

- SJØBERG, D. Quantifying schema evolution. **Information and Software Technology**, v. 35, p. 35 – 44, 1993. Citado na página 90.
- SKOULIS, I.; VASSILIADIS, P.; ZARRAS, A. Open-source databases: Within, outside, or beyond lehman’s laws of software evolution?. In: . [S.l.: s.n.], 2014. v. 8484 LNCS, p. 379 – 393. Citado na página 90.
- SKOULIS, I.; VASSILIADIS, P.; ZARRAS, A. V. Growing up with stability: How open-source relational databases evolve. **Information Systems**, v. 53, p. 363 – 385, 2015. Citado na página 89.
- SOLODOVNIKOVA, D. Metadata to support data warehouse evolution. In: . [S.l.: s.n.], 2009. p. 627 – 635. Citado na página 90.
- SOMMERVILLE, I. Engenharia de software, 8 edição. **Pearson, Addison Wesley**, v. 8, n. 9, p. 10, 2007. Citado 2 vezes nas páginas 32 e 33.
- SONG, Y. et al. Automatic adaptation of software applications to database evolution by graph differencing and aop-based dynamic patching. In: . [S.l.: s.n.], 2012. p. 111 – 118. Citado 2 vezes nas páginas 45 e 90.
- SUBOTIC, D. Data warehouse schema evolution perspectives. In: . [S.l.: s.n.], 2015. v. 312, p. 333 – 338. Citado na página 92.
- SVENSSON, P. The evolution of vertical database architectures - a historical review. **Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)**, v. 5069 LNCS, p. 3–5, 2008. Citado na página 93.
- TAKAHASHI, J. Hybrid relations for database schema evolution. In: **Proceedings., Fourteenth Annual International Computer Software and Applications Conference**. [S.l.: s.n.], 1990. p. 465–470. Citado na página 89.
- TERWILLIGER, J. et al. Support for schema evolution in data stream management systems. **Journal of Universal Computer Science**, v. 16, p. 3073–3101, 2010. Citado na página 93.
- TERWILLIGER, J. F. Bidirectional by necessity: Data persistence and adaptability for evolving application development. In: . [S.l.: s.n.], 2013. v. 7680 LNCS, p. 219 – 270. Citado na página 91.
- TERWILLIGER, J. F.; BERNSTEIN, P. A.; UNNITHAN, A. Automated co-evolution of conceptual models, physical databases, and mappings. In: . [S.l.: s.n.], 2010. v. 6412 LNCS, p. 146 – 159. Citado 2 vezes nas páginas 45 e 91.
- TERWILLIGER, J. F.; BERNSTEIN, P. A.; UNNITHAN, A. Worry-free database upgrades: Automated model-driven evolution of schemas and complex mappings. In: . [S.l.: s.n.], 2010. p. 1191 – 1194. Citado 2 vezes nas páginas 47 e 91.
- TITE, C. **Evolutionary Database Design**. 2014. Disponível em: <https://www.youtube.com/watch?v=QuZnVn_6cBY>. Citado 2 vezes nas páginas 63 e 86.
- TURKER, C. Schema evolution in sql-99 and commercial (object-)relational dbms. In: . [S.l.: s.n.], 2001. v. 2065, p. 1 – 32. Citado na página 92.

- VASSILIADIS, P. Schema evolution and gravitation to rigidity: A tale of calmness in the lives of structured data. In: . [S.l.: s.n.], 2017. v. 10563 LNCS, p. 18 – 23. Citado 2 vezes nas páginas 46 e 91.
- VASSILIADIS, P. et al. Schema evolution and foreign keys: Birth, eviction, change and absence. In: . [S.l.: s.n.], 2017. v. 10650 LNCS, p. 106 – 119. Citado na página 91.
- VASSILIADIS, P. et al. Schema evolution and foreign keys: a study on usage, heartbeat of change and relationship of foreign keys to table activity. **Computing**, 2019. Citado na página 89.
- VASSILIADIS, P.; ZARRAS, A. V. Survival in schema evolution: Putting the lives of survivor and dead tables in counterpoint. In: . [S.l.: s.n.], 2017. v. 10253 LNCS, p. 333 – 347. Citado na página 91.
- VASSILIADIS, P.; ZARRAS, A. V.; SKOULIS, I. Gravitating to rigidity: Patterns of schema evolution – and its absence – in the lives of tables. **Information Systems**, v. 63, p. 24 – 46, 2017. Citado na página 90.
- VIANU, V. Dynamic constraints and database evolution. In: **Proceedings of the 2Nd ACM SIGACT-SIGMOD Symposium on Principles of Database Systems**. [S.l.]: PODS '83, 1983. p. 389–399. ISBN 0-89791-097-4. Citado na página 93.
- VIANU, V. Dynamic functional dependencies and database aging. **Journal of the ACM (JACM)**, v. 34, p. 28–59, 1987. Citado na página 93.
- WALL, B.; ANGRYK, R. Storing long-lived concurrent schema and data versions in relational databases. In: . [S.l.: s.n.], 2015. v. 312, p. 97 – 108. Citado na página 90.
- WANG, H.; SHEN, B.; CHEN, C. Model-driven reengineering of database. In: . [S.l.: s.n.], 2009. v. 3, p. 113 – 117. Citado 2 vezes nas páginas 46 e 90.
- WEIGAND, H.; KREMERS, B. An object-oriented active data dictionary to support database evolution. In: **Integrated Management of Technical Documentation**. [S.l.: s.n.], 1992. p. 150–167. ISBN 978-3-540-55880-4 978-3-642-84840-7. Citado na página 89.
- WISNESKY, R. et al. Using category theory to facilitate multiple manufacturing service database integration. **Journal of Computing and Information Science in Engineering**, v. 17, 2017. Citado na página 93.
- WU, S.; NEAMTIU, I. Schema evolution analysis for embedded databases. In: **IEEE 27th International Conference on Data Engineering Workshops**. [S.l.: s.n.], 2011. p. 151–156. Citado na página 89.
- WU, S.; ZHANG, S.; KONG, L. Schema evolution via multi-version metadata in saas. In: . [S.l.: s.n.], 2012. v. 29, p. 1107 – 1112. Citado 2 vezes nas páginas 46 e 92.
- WU, X. et al. Managing schema evolution in a federated spatial database system. In: _____. [S.l.: s.n.], 2012. p. 56 – 77. Citado na página 92.
- XUE, J. et al. A transparent approach for database schema evolution using view mechanism. In: . [S.l.: s.n.], 2012. v. 7418 LNCS, p. 405 – 418. Citado 2 vezes nas páginas 45 e 92.

- YANNAKOUDAKIS, E.; DIAMANTIS, I. Beyond database schema evolution. In: . [S.l.: s.n.], 2008. v. 1, p. 245 – 250. Citado 2 vezes nas páginas 45 e 91.
- YOON, J. P.; KERSCHBERG, L. A framework for knowledge discovery and evolution in databases. **IEEE Transactions on Knowledge and Data Engineering**, v. 5, p. 973–979, 1993. Citado 2 vezes nas páginas 45 e 89.
- YU, C.; POPA, L. Semantic adaptation of schema mappings when schemas evolve. In: . [S.l.: s.n.], 2005. v. 3, p. 1006 – 1017. Citado na página 93.
- ZASCHKE, T.; NORRIE, M. C. Revisiting schema evolution in object databases in support of agile development. In: . [S.l.: s.n.], 2010. v. 6348 LNCS, p. 10 – 24. Citado na página 91.
- ZGHAL, S. Change operators for managing schema evolution in geographical databases. In: . [S.l.: s.n.], 2013. v. 3, p. 187 – 193. Citado na página 91.
- ZHANG, K.; LI, Q.; SHI, Y. Data privacy preservation during schema evolution for multi-tenancy applications in cloud computing. In: . [S.l.: s.n.], 2011. v. 6987 LNCS, p. 376 – 383. Citado na página 92.

Apêndices

APÊNDICE A – *GUIDELINE* PARA EVOLUÇÃO DE BANCO DE DADOS

A.1 INTRODUÇÃO

Com o avanço tecnológico, a utilização de banco de dados se tornou constante no dia a dia da sociedade, provocando grande impacto no desenvolvimento de *software* devido ao armazenamento e acesso de dados de forma rápida, permitindo a automatização de processos e tornando mais acessível os dados de forma virtualizada. Nossas vidas passaram a estar contidas em banco de dados e se tornaram informações valiosas para empresas, principalmente pela necessidade de destaque no mercado competitivo (IMPACTA, 2017). Existem vários sistemas cotidianos que precisam usar banco de dados para realizar suas funções, como é o caso dos que realizam operações bancárias, reservas em hotéis, compras de produtos em lojas físicas ou *onlines*, gerenciamento de estoques em supermercados, entre outros.

Os sistemas são feitos para refletir o comportamento do mundo real sendo necessário que o *software* acompanhe as mudanças impostas pelo meio em que está inserido (GALL et al., 1997). Se o sistema não sofre essas mudanças, pode ficar obsoleto e cair em desuso. Uma das maneiras de mitigar esse problema e modificar um *software* é através da evolução. A evolução de *software* compreende qualquer mudança que ocorre com um sistema a fim de deixá-lo completo e, se possível, livre de *bugs* tendo como base o *feedback* dos usuários. Consequentemente, com a evolução de um *software* há a necessidade de evoluir as tecnologias que são utilizadas pelo mesmo, como é o caso do banco de dados.

Sabe-se que a evolução de banco de dados consiste em qualquer alteração no banco, que tenha o objetivo de resolver algum problema imposto por um banco limitado ou pela evolução de um *software* ao longo das iterações de um processo de desenvolvimento. No entanto, a evolução de banco de dados é uma das atividades da Engenharia de Sistemas que não é claramente caracterizada na literatura. Primeiramente, falta uma boa caracterização para poder resultar em *guidelines* e/ou metodologias e ferramentas que levem a desempenhos ótimos do times de desenvolvimento de *software* nas tarefas de evolução de bancos de dados.

A fim de elencar quais são as principais práticas utilizadas tanto na área acadêmica quanto na área industrial, foi aprofundada a literatura da área (LAUKKANEN; ITKONEN; LASSENIUS, 2017) e condensada em um *guideline* de práticas de evolução para problemas específicos. Para isso, buscou-se por meio de uma extensiva revisão de literatura, em especial, as metodologias e suporte ferramental disponíveis para auxiliar na evolução de bancos de dados, bem como seus atributos de qualidade de propostas para evolução que direcionam a fatores de sucesso nas práticas de de evolução de banco de dados.

Este documento apresenta as boas práticas em um *guideline* que tem como pro-

pósito dirigir a escolha de materiais específicos conforme problemas de evolução de banco de dados. O restante do documento é organizado como segue: A Seção A.2 apresenta os requisitos para o melhor uso do *guideline*; A Seção A.3 apresenta o *guideline* derivado de uma extensiva revisão de literatura no tema; Por fim, a Seção A.4 apresenta as considerações finais quanto ao *guideline* e problema apresentados.

A.2 Problema Motivante e Requisitos para o Melhor Uso do Guideline

O problema de evolução de banco de dados será apresentado em aula. Com a finalidade didática, recomenda-se elaborar o problema conforme a palestra em inglês de (TITE, 2014).

Requisitos para os alunos iniciarem na temática de evolução de banco de dados:

- Conhecimentos básicos em gerenciamento de configuração;
- Conhecimentos básicos em gerenciamento de *releases*;
- Conhecimentos básicos em modelagem entidade-relacionamento;
- Conhecimentos básicos em sistemas gerenciadores de banco de dados;
- Conhecimentos básicos em desenvolvimento de sistemas de informação;
- Ter participado do desenvolvimento de algum software que inclui a persistência de dados em SGBDs, relacionais ou não;
- Ter interesse em se aprofundar em leituras em material específico do tema.

Requisitos para os professores introduzirem o problema na temática de evolução de banco de dados:

- Certificar-se de que os alunos possuem um Repositório de Código configurado;
- Certificar-se de que os alunos possuem um SGBD e um banco de dados operacional;
- Certificar-se que os problemas de software tenham sido previamente conduzidos.
- Certificar-se de que os alunos possuem os conhecimentos mínimos para o problema.

A.3 GUIDELINE

O presente guideline oferece um conjunto de referências bibliográficas no tema tratando da solução de determinados problemas de evolução. Tais estudos foram extraídos de um estudo de mapeamento sistemática de literatura. A seguir, apresentamos os critérios de qualidade medidos para propósito de esclarecimento. O propósito dos critérios de

qualidade é verificar quais estudos contribuem mais com os propósitos e motivação deste mapeamento. Dessa forma, as notas geradas pelos critérios não são usadas para eliminar estudos, e são organizadas como segue:

CQ1. O estudo apresenta algum domínio de aplicação como alvo de práticas de evolução de banco de dados?

Respostas - S: O estudo apresenta totalmente o domínio alvo de práticas de evolução de banco de dados; P: O estudo apresenta parcialmente o domínio alvo de práticas de evolução de banco de dados; N: O estudo não apresenta o domínio alvo de práticas de evolução de banco de dados.

CQ2. O estudo apresenta alguma prática/técnica, tecnologia/ferramenta, *guideline*/ processo/abordagem que são apresentadas para **realizar** a evolução de banco de dados?

Respostas - S: O estudo apresenta totalmente alguma prática/técnica, tecnologia/ferramenta, *guideline*/processo/abordagem que são apresentadas para realizar a evolução de banco de dados; P: O estudo apresenta parcialmente alguma alguma prática/técnica, tecnologia/ferramenta, *guideline*/processo/abordagem que são apresentadas para realizar a evolução de banco de dados; N: O estudo não apresenta nenhuma alguma prática/técnica, tecnologia/ferramenta, *guideline*/processo/ abordagem que são apresentadas para realizar a evolução de banco de dados.

CQ3. O estudo apresenta prática/técnica, tecnologia/ferramenta, *guideline*/ processo/abordagem que **apoiam** a evolução de banco de dados?

Respostas - S: O estudo apresenta totalmente alguma prática/técnica, tecnologia/ferramenta, *guideline*/processo/abordagem que são apresentadas que apoiam a evolução de banco de dados; P: O estudo apresenta parcialmente alguma alguma prática/técnica, tecnologia/ferramenta, *guideline*/processo/abordagem que são apresentadas que apoiam a evolução de banco de dados; N: O estudo não apresenta nenhuma alguma prática/técnica, tecnologia/ferramenta, *guideline*/processo/abordagem que são apresentadas que apoiam a evolução de banco de dados.

CQ4. O estudo apresenta quais são os vantagens, desvantagens e tendências na parte de evolução de banco de dados?

Respostas - S: O estudo apresenta totalmente vantagens, desvantagens e tendências; P: O estudo apresenta parcialmente vantagens, desvantagens e tendência; N: O estudo não apresenta vantagens, desvantagens e tendências.

CQ5. O estudo apresenta uma evolução de banco de dados utilizando integração contínua?

Resposta - S: O artigo apresenta uma evolução de banco utilizando IC; P: O estudo apresenta parcialmente uma evolução de banco de dados utilizando IC; N: O estudo não apresenta uma evolução de *software* utilizando IC.

A aplicação dos critérios de qualidade tem como objetivo de verificar os estudos que mais contribuem para propósito da formulação deste *guideline*. A Tabela 7 fornece as informações sobre as pontuações neste quesito. Cada estudo possui um identificador (ID) e sua respectiva referência. As colunas CQ1, CQ2, CQ3, CQ4 e CQ5 mostram a pontuação baseada do estudo em cada critério de qualidade. A coluna "Score Geral" descreve se o estudo é Pouco Relevante, Relevante ou Muito Relevante para nosso objetivo. A coluna "Score" mostra a pontuação final de cada estudo. E, por fim, a coluna Status demonstra se já foi realizada a aplicação dos critérios de qualidade para o estudo. Foram avaliados separadamente 55 estudos, de acordo com os cinco critérios de qualidade. Utilizamos os critérios de qualidade como uma forma de verificar a relevância desse estudo para nossos objetivos, desta forma, nenhum estudo foi excluído nesta etapa.

Tabela 7 – Aplicação dos Critérios de Qualidade

ID	Referência	CQ1	CQ2	CQ3	CQ4	CQ5	Score Geral	Score	Status
E1	(EWALD; ORLOWSKA, 1993)							0.0	Não Classificado
E2	(WEIGAND; KREMERS, 1992)							0.0	Não Classificado
E3	(CURINO et al., 2013)							0.0	Não Classificado
E4	(SIMANOVSKY, 2008)							0.0	Não Classificado
E5	(HAINAUT et al., 1994)	S	N	S	P	N	Relevante	2.5	Classificado
E6	(RISTIĆ et al., 2014)							0.0	Não Classificado
E7	(LIU; CHRYSANTHIS; CHANG, 1994)	N	S	S	P	N	Relevante	2.5	Classificado
E8	(BELLAHSENE, 1997)	S	N	S	P	N	Relevante	2.5	Classificado
E9	(GORALWALLA et al., 1997)							0.0	Não Classificado
E10	(VASSILIADIS et al., 2019)							0.0	Não Classificado
E11	(CLAMEN, 1994)							0.0	Não Classificado
E12	(CHEN; MCLEOD; O’LEARY, 1994)							0.0	Não Classificado
E13	(BELLAHSENE, 2002)	P	N	S	S	N	Relevante	2.5	Classificado
E14	(BAEKGAARD, 1997)	S	S	S	P	N	Relevante	3.5	Classificado
E15	(MEURICE; CLEVE, 2017)	S	S	S	P	N	Relevante	3.5	Classificado
E16	(MASOOD; ANDLEEB, 2014)							0.0	Não Classificado
E17	(OVEREEM; SPOOR; JANSEN, 2017)							0.0	Não Classificado
E18	(MEURICE; CLEVE, 2014)							0.0	Não Classificado
E19	(DRAHEIM; HORN; SCHULZ,)	P	N	S	P	N	Relevante	2.0	Classificado
E20	(MEURICE; NAGY; CLEVE, 2016)							0.0	Não Classificado
E21	(SCHERZINGER; CERQUEUS; ALMEIDA,)	S	S	S	P	N	Relevante	3.5	Classificado
E22	(FILATOV; SEMENETS, 2018)							0.0	Não Classificado
E23	(LIU; CHANG; CHRYSANTHIS,)	S	S	S	P	N	Relevante	3.5	Classificado
E24	(AND; AND,)	S	N	S	P	N	Relevante	0.0	Classificado
E25	(KLETTKE et al., 2016)	S	N	S	S	S	Muito Relevante	4.0	Classificado
E26	(SCHINK, 2013)							0.0	Não Classificado
E27	(PAPASTEFANATOS et al., 2010)							0.0	Não Classificado
E28	(HENRARD et al., 2008)							0.0	Não Classificado
E29	(TAKAHASHI, 1990)							0.0	Não Classificado
E30	(WU; NEAMTIU, 2011)							0.0	Não Classificado
E31	(SCALAS; CAPPELLI; CASTRO, 1993)	P	N	S	P	N	Relevante	2.0	Classificado
E32	(YOON; KERSCHBERG, 1993)	P	P	S	P	N	Relevante	2.5	Classificado
E33	(MINSKY; ROZENSHTAIN; CHOMICKI, 1986)	S	S	P	P	N	Relevante	3.0	Classificado
E34	(SKOULIS; VASSILIADIS; ZARRAS, 2015)							0.0	Não Classificado

Tabela 7 – Aplicação dos Critérios de Qualidade

ID	Referência	CQ1	CQ2	CQ3	CQ4	CQ5	Score Geral	Score	Status
E35	(SJØBERG, 1993)							0.0	Não Classificado
E36	(PROPER, 1997)	P	N	S	P	N	Relevante	2.0	Classificado
E37	(VASSILIADIS; ZARRAS; SKOULIS, 2017)							0.0	Não Classificado
E38	(ARIAV, 1991)							0.0	Não Classificado
E39	(GROLINGER; CAPRETZ, 2011)	P	S	S	S	N	Relevante	3.5	Classificado
E40	(CHEN; MCLEOD; O'LEARY, 1995)	S	S	N	P	N	Relevante	2.5	Classificado
E41	(MCKENZIE; SNODGRASS, 1990)							0.0	Não Classificado
E42	(HSU, 1986)	P	N	S	P	N	Relevante	2.0	Classificado
E43	(DOMÍNGUEZ et al., 2008)							0.0	Não Classificado
E44	(CLEVE et al., 2015)							0.0	Não Classificado
E45	(REITER, 1995)							0.0	Não Classificado
E46	(MOTZ, 2005)							0.0	Não Classificado
E47	(MORAES; SALGADO; TEDESCO, 2009)	S	N	S	P	N	Relevante	2.5	Classificado
E48	(CARUCCIO; POLESE; TORTORA, 2016)							0.0	Não Classificado
E49	(JENSEN; BOHLEN, 2002)							0.0	Não Classificado
E50	(MANNISTO; SULONEN, 1999)	P	S	S	S	P	Muito Relevante	4.0	Classificado
E51	(LIN; NEAMTIU, 2009)							0.0	Não Classificado
E52	(SKOULIS; VASSILIADIS; ZARRAS, 2014)							0.0	Não Classificado
E53	(WALL; ANGRYK, 2015)							0.0	Não Classificado
E54	(SOLODOVNIKOVA, 2009)							0.0	Não Classificado
E55	(MORZY; WREMBEL, 2003)							0.0	Não Classificado
E56	(DOMINGUES et al., 2013)							0.0	Não Classificado
E57	(HICK; HAINAUT, 2003)							0.0	Não Classificado
E58	(CLEVE; HAINAUT, 2006)							0.0	Não Classificado
E59	(WANG; SHEN; CHEN, 2009)	S	S	S	S	N	Muito Relevante	4.0	Classificado
E60	(HERRMANN et al., 2015b)	P	S	N	P	N	Relevante	2.0	Classificado
E61	(SONG et al., 2012)	P	N	S	S	N	Relevante	2.5	Classificado
E62	(LIU et al., 2010)	S	N	S	P	N	Relevante	2.5	Classificado
E63	(HERRMANN et al., 2018)							0.0	Não Classificado
E64	(HERRMANN et al., 2015a)	S	S	N	P	N	Relevante	2.5	Classificado
E65	(PAPASTEFANATOS; VASSILIADIS; VASSILIOU, 2006)	P	N	S	S	N	Relevante	2.5	Classificado
E66	(HERRMANN et al., 2017)							0.0	Não Classificado
E67	(NGUEZ; LLORET; ZAPATA, 2003)	P	N	S	P	N	Relevante	2.0	Classificado
E68	(SCHULER; KESSELMAN, 2018)	P	N	S	S	N	Relevante	2.5	Classificado

Tabela 7 – Aplicação dos Critérios de Qualidade

ID	Referência	CQ1	CQ2	CQ3	CQ4	CQ5	Score Geral	Score	Status
E69	(REITER, 1992)							0.0	Não Classificado
E70	(LAMMARI; AKOKA; COMYN-WATTIAU, 2003)	N	N	S	P	N	Relevante	1.5	Classificado
E71	(MARCINIAK; PANKOWSKI, 2013)							0.0	Não Classificado
E72	(TERWILLIGER; BERNSTEIN; UNNITHAN, 2010a)							0.0	Não Classificado
E73	(GOSAIN; SAROHA, 2018)							0.0	Não Classificado
E74	(TERWILLIGER; BERNSTEIN; UNNITHAN, 2010b)	P	S	S	P	N	Relevante	3.0	Classificado
E75	(MOON et al., 2009)							0.0	Não Classificado
E76	(CARUCCIO; POLESE; TORTORA, 2018)							0.0	Não Classificado
E77	(TERWILLIGER, 2013)							0.0	Não Classificado
E78	(LUCIV, 2005)							0.0	Não Classificado
E79	(CLAYPOOL; NATARAJAN; RUNDENSTEINER, 2001)							0.0	Não Classificado
E80	(GOTZ; KUHN, 2015)	S	S	S	S	S	Muito Relevante	5.0	Classificado
E81	(BOUNIF; SPACCAPIETRA; POTTINGER, 2007)							0.0	Não Classificado
E82	(ZASCHKE; NORRIE, 2010)							0.0	Não Classificado
E83	(CURINO; MOON; ZANIOLO, 2008a)	P	N	S	P	N	Relevante	2.0	Classificado
E84	(CURINO; MOON; ZANIOLO, 2008b)							0.0	Não Classificado
E85	(MORO; MALAIKA; LIM, 2007)							0.0	Não Classificado
E86	(VASSILIADIS, 2017)							0.0	Não Classificado
E87	(VASSILIADIS et al., 2017)							0.0	Não Classificado
E88	(POLESE; VACCA, 2009)							0.0	Não Classificado
E89	(SCHERZINGER et al., 2016)							0.0	Não Classificado
E90	(BLASCHKA; SAPIA; HOFLING, 1999)							0.0	Não Classificado
E91	(DELGADO; SAMOS; TORRES, 2003)							0.0	Não Classificado
E92	(SCHERZINGER; KLETTKE; STORL, 2015)	S	N	S	N	N	Relevante	2.0	Classificado
E93	(GUEDES; BAIOCO; MORAES, 2016)	P	N	S	P	N	Relevante	2.0	Classificado
E94	(JENSEN; BOHLEN, 2004a)	P	S	S	P	N	Relevante	3.0	Classificado
E95	(JENSEN; BOHLEN, 2004b)	N	N	S	P	N	Relevante	1.5	Classificado
E96	(YANNAKOUDAKIS; DIAMANTIS, 2008)							0.0	Não Classificado
E97	(BOSSUNG et al., 2006)							0.0	Não Classificado
E98	(CONNOR et al., 1994)							0.0	Não Classificado
E99	(VASSILIADIS; ZARRAS, 2017)							0.0	Não Classificado
E100	(CHIAO; KUNZLE; REICHERT, 2014)							0.0	Não Classificado
E101	(ZGHAL, 2013)							0.0	Não Classificado
E102	(KAAS; PEDERSEN; RASMUSSEN, 2004)							0.0	Não Classificado

Tabela 7 – Aplicação dos Critérios de Qualidade

ID	Referência	CQ1	CQ2	CQ3	CQ4	CQ5	Score Geral	Score	Status
E103	(NEJI et al., 2006)	P	P	S	S	N	Relevante	3.0	Classificado
E104	(JIANFENG; BO, 2009)							0.0	Não Classificado
E105	(MAROTTA; MOTZ; RUGGIA, 2002)							0.0	Não Classificado
E106	(LERNER, 2000)	P	P	S	S	N	Relevante	3.0	Classificado
E107	(JONG; DEURSEN, 2015)	S	S	S	P	S	Muito Relevante	4.5	Classificado
E108	(PAPASTEFANATOS et al., 2008b)	S	S	S	P	N	Relevante	3.5	Classificado
E109	(DELPLANQUE et al., 2018)							0.0	Não Classificado
E110	(JONG; DEURSEN; CLEVE, 2017)	S	S	S	S	S	Muito Relevante	5.0	Classificado
E111	(TURKER, 2001)							0.0	Não Classificado
E112	(PAPASTEFANATOS et al., 2008a)							0.0	Não Classificado
E113	(CURINO et al., 2010)	S	S	S	P	N	Relevante	3.5	Classificado
E114	(LIU; CHANG; CHRYSANTHIS, 1994)							0.0	Não Classificado
E115	(GOBERT et al., 2013)							0.0	Não Classificado
E116	(CHANG et al., 2007)	P	S	S	P	N	Relevante	3.0	Classificado
E117	(HAMAJI; NAKAMOTO, 2016)							0.0	Não Classificado
E118	(DOMINGUES et al., 2014)	P	P	S	P	N	Relevante	2.5	Classificado
E119	(ATHINAIYOU; KONDYLAKIS, 2019)	S	N	S	P	N	Relevante	2.5	Classificado
E120	(MANOUSIS et al., 2016)							0.0	Não Classificado
E121	(BOUNIF; POTTINGER, 2006)	S	N	S	P	N	Relevante	2.5	Classificado
E122	(CURINO et al., 2008)							0.0	Não Classificado
E123	(BANERJEE; DAVIS, 2009)	S	N	S	P	N	Relevante	2.5	Classificado
E124	(XUE et al., 2012)	P	S	P	S	N	Relevante	3.0	Classificado
E125	(CURINO; MOON; ZANIOLO, 2009)							0.0	Não Classificado
E126	(HUO; TSOTRAS, 2012)							0.0	Não Classificado
E127	(CURINO et al., 2009)							0.0	Não Classificado
E128	(ZHANG; LI; SHI, 2011)							0.0	Não Classificado
E129	(WU; ZHANG; KONG, 2012)	S	N	S	S	N	Relevante	3.0	Classificado
E130	(WU et al., 2012)							0.0	Não Classificado
E131	(SUBOTIC, 2015)							0.0	Não Classificado
E132	(BAQASAH; PARDEDE, 2010)							0.0	Não Classificado
E133	(DOMINGUEZ et al., 2008)	P	N	S	P	N	Relevante	2.0	Classificado
E134	(QIU; LI; SU, 2013)							0.0	Não Classificado
E135	(DOMINGUES; KON; FERREIRA, 2011)							0.0	Não Classificado
E136	(COOX, 2003)	S	N	S	P	N	Relevante	2.5	Classificado

Tabela 7 – Aplicação dos Critérios de Qualidade

ID	Referência	CQ1	CQ2	CQ3	CQ4	CQ5	Score Geral	Score	Status
E137	(YU; POPA, 2005)							0.0	Não Classificado
E138	(ASSOUDI; LOUNIS, 2011)	N	N	S	S	N	Relevante	2.0	Classificado
E139	(FAVRE; BENTAYEB; BOUSSAID, 2007)							0.0	Não Classificado
E140	(WISNESKY et al., 2017)							0.0	Não Classificado
E141	(FAISAL et al., 2017)							0.0	Não Classificado
E142	(MEIMARIS; PASTEFANATOS, 2016)							0.0	Não Classificado
E143	(MACEK; RICHTA, 2014)	P	S	N	S	N	Relevante	2.5	Classificado
E144	(BRUMMERMANN; KEUNECKE; SCHMID, 2013)							0.0	Não Classificado
E145	(MAATUK; ALI; ROSSITER, 2011)							0.0	Não Classificado
E146	(AULBACH et al., 2011)							0.0	Não Classificado
E147	(CLEVE; BROGNEAUX; HAINAUT, 2010)	P	S	S	P	N	Relevante	3.0	Classificado
E148	(TERWILLIGER et al., 2010)							0.0	Não Classificado
E149	(MOON; CURINO; ZANIOLO, 2010)							0.0	Não Classificado
E150	(SVENSSON, 2008)							0.0	Não Classificado
E151	(RODDICK; VRIES, 2006)							0.0	Não Classificado
E152	(FAN; POULOVASSILIS, 2004)							0.0	Não Classificado
E153	(RA, 2004)							0.0	Não Classificado
E154	(LIM; CHIANG, 2004)							0.0	Não Classificado
E155	(LU, 2002)							0.0	Não Classificado
E156	(CASTELLI, 1999)							0.0	Não Classificado
E157	(BOMMEL, 1994)	P	N	S	S	N	Relevante	2.5	Classificado
E158	(RODDICK, 1992a)							0.0	Não Classificado
E159	(RODDICK, 1992b)							0.0	Não Classificado
E160	(FIADEIRO; SERNADAS, 1988)							0.0	Não Classificado
E161	(VIANU, 1987)							0.0	Não Classificado
E162	(MOEHRKE, 1971)							0.0	Não Classificado
E163	(ANDANY; L'EONARD; PALISSER, 1991)							0.0	Não Classificado
E164	(MOON et al., 2008)							0.0	Não Classificado
E165	(FOSTER, 1975)							0.0	Não Classificado
E166	(VIANU, 1983)							0.0	Não Classificado
E167	(AGARWAL; SARANGI; DAS, 2003)							0.0	Não Classificado

A.3.1 Identificar o Tipo de SGBD Adotado

Diferentes técnicas para evolução de dados são aplicadas para diferentes tipos de Sistemas Gerenciadores de Banco de Dados. A seguir, apresenta-se os tipos encontrados. Para seguir na resolução do problema, você deve analisar o contexto do problema introduzido e identificar qual tipo o seu problema se enquadra:

- Estudos focados em bancos de dados relacionais (E133, E68, E123, E62, E50, E118, E106, E32, E59). Aqui, alguns bancos relacionais que também foram o foco de alguns estudos, como o PostgreSQL (E110), Oracle (E13) e XML com um modelo relacional (E136);
- Estudos focados somente em banco de dados que utilizam a linguagem SQL para realizar seu gerenciamento da evolução (E74, E19, E80, E107, E113, E108);
- Estudo aplicado ao banco NOSQL MongoDB e práticas ágeis, para realizar a implantação contínua (E25);
- Estudos que também apresentam foco em NOSQL, entretanto, sem a especificação de um banco de dados (E25, E21, E15);
- Estudo com processo de evolução em um nível de abstração mais alto, como em modelos entidade-relacionamento (E23).

A.3.2 Identificar Problemas Específicos de Evolução de Banco de Dados

Os artigos de referência, em geral, tratam de problemas de banco de dados em domínios de sistemas de informação.

Identificar soluções para problemas mais gerais que necessitem de tratamento específico de abordagens de evolução:

- Software como um Serviço (SaaS): O estudo E129 discute o tema e o estudo E47 foca em *multi tenancy*¹, cujo objetivo é operar em (estudo E129), oferecendo os seguintes elementos: segurança, isolamento de dados do cliente, escalabilidade do banco de dados dentre outros;
- *Data Warehouse*: que apresenta um modo de apoiar a evolução em *Data Marts* de acordo com a prática de ETL (*Extraction, Transformation and Load*);
- Variabilidade: Quando a evolução de banco de dados deve ocorrer de acordo com variabilidades de contextos das organizações, que estão direcionadas para técnicas em problemas de linha de produto de software (E64);
- Federação de dados: a federação de dados envolve referenciar dados de diferentes bases. Estudos (E33, E8) que tem o objetivo de auxiliar um Sistema de Banco de Dados e um Sistema de Banco de Dados Federados a realizar a evolução.
- Dados de georreferencia: A aplicação de evolução para dados de SMOs é um tópico de pesquisa bastante interessante (E113).

Identificar soluções para problemas associados com tecnologias específicas:

- Adaptação de Aplicativos (E61);
- Linguagem de Programação JAVA (E147, E15);
- Modularização de Banco de Dados (E93);
- Banco de dados MySQL (E124);

¹ <https://blakebhowe.medium.com/multi-tenant-application-database-design-e4c2d161f3dd>

- Sistemas de informação da Web (E83);
- Banco de Dados atuais (E94).

Estudos para a automação do processo de evolução:

- Novas linguagens para evoluir o banco de dados (E60, E14);
- *Frameworks* para evoluir o banco de dados (E116, E21);
- Avaliar as diferenças de versões de banco de dados, expressando ambas versões do banco e a relação entre elas (E143, E80);
- Outros, como o estudo (E32), que em decorrência de não apresentar uma contribuição substancial, discutem como o engenheiro de software deve proceder para realizar a evolução;
- Na Tabela 8, é possível encontrar mais detalhes relacionados a cada modo mencionado.

A.3.3 Adaptar O Processo Base para Problemas Específicos de Evolução de Banco de Dados

Uma das necessidades dos times de desenvolvimento de software é adaptar o processo de desenvolvimento base para um processo especialista, o que envolve identificar atividades e suporte ferramental que são combinadas em uma cadeia de ferramentas. Neste rumo, os seguintes elementos podem ser considerados:

Identificar as atividades para apoiar a evolução de banco de dados:

- Estabelecer uma abordagem para resolver problemas de evolução. Abordagem caracteriza processos instanciados, escopo da maioria dos tipos de trabalhos anualizados estudados (E13, E15, E59, E61, E94, E103, E110, E121). Tais estudos podem ser uma boa fonte de informação para identificar atividades em específico que devem ser realizadas para garantir que a evolução ocorra sem perda de informações;
- Estabelecer um framework para base de conhecimento da fábrica de software para guiar a evolução de banco de dados (E19, E68, E92, E108). Estabelecer um framework pode ajudar a equipe à evoluir um processo base para processos de software adaptados (instâncias de processos), onde os estudos (E39, E93, E118, E138) apresentam exemplos;
- Estabelecer atividades de apoio à evolução é através do uso de técnicas de evolução (E32) e estratégias (E25).

Tabela 8 – Detalhes das realizações de evolução de banco de dados

ID	Tipo	Descrição
E5	Outro	Ambiente CASE que possui funções para engenharia direta e reversa de banco de dados e assistente no processamento de esquemas.
E7	Abordagem	Abordagem de derivação de esquemas para a evolução através de mudanças no esquema Entidade-Relacionamento.
E14	Linguagem	Linguagem de duas camadas ortogonal às linguagens de especificação baseadas em estado, podendo ser um complemento para tais.
E15	Abordagem	Abordagem automática para inferir o esquema de um banco de dados NoSQL sem esquema e analisar sua evolução ao longo do tempo.

Tabela 8 – Detalhes das realizações de evolução de banco de dados

ID	Tipo	Descrição
E21	<i>Framework</i>	<i>Framework</i> para evolução controlada de esquemas no desenvolvimento de aplicativos em relação a armazenamentos de dados NoSQL.
E23	Abordagem	Abordagem para a evolução do esquema através de alterações no esquema Entidade-Relacionamento (ER) de um banco de dados
E32	Outros	Conciliam ou harmonizam novas regras do banco de dados com as antigas, a fim de manter a consistência.
E33	Abordagem	Abordagem baseada na linguagem Prolog com o objetivo de adicionar restrições no banco de dados através da criação de um modelo.
E39	Abordagem	Abordagem suportada por teste unitário para o código de aplicativo que acessa bancos de dados com o objetivo de avaliar proativamente o código em relação ao banco de dados alterado.
E40	Abordagem	Abordagem baseada no conhecimento para apoiar a evolução do esquema de banco de dados contábil dinâmico em um contexto de modelagem de dados baseado em objeto.
E50	Abordagem	Abordagem para evolução de esquema de banco de dados
E59	Abordagem	Abordagem baseada em modelos para a gerência e realização da evolução
E60	Linguagem	Linguagem de referência para implementações produtivas de evolução de banco de dados em SGBDs denominada CoDEL.
E64	Outros	Kit de ferramentas de evolução de banco de dados para o SPL.
E74	Extensão	Extensão do Visual Studio que suporta a evolução automática de artefatos de mapeamento objeto-relacional no <i>Microsoft Entity Framework</i> .
E80	Modelo	Modelo de tempo de execução que realiza a evolução do esquema de banco de dados e analisa o código fonte facilitando a configuração.
E94	Outros	Esquema concluído estendido que registra atributos juntamente com suas incompatibilidades.
103	Abordagem	Abordagem que adapta o MS e é baseada em um conjunto de operadores algébricos.
E107	Abordagem	Abordagem para evolução de banco de dados, no contexto de Integração Contínua, utilizando versões do mesmo banco.
E108	Extensão	Extensão viável e poderosa para a linguagem SQL adaptada para o gerenciamento da evolução.
E110	Abordagem	Abordagem para implantar vários esquemas ativos simultaneamente e migrar dados entre esses esquemas.
E113	Outros	<i>Schema Modification Operators</i> (SMO), denominado PRISM ++.
E116	<i>Framework</i>	<i>Framework</i> para refatoração de banco de dados, baseado em lógica epistêmica, usa um agente para descobrir e resolver inconsistências e analisa o impacto das alterações nas consultas.

Tabela 8 – Detalhes das realizações de evolução de banco de dados

ID	Tipo	Descrição
E119	Outros	Sistema apresentado pelo artigo utiliza o BiDEL que é uma linguagem de evolução de banco de dados que tem o objetivo de evoluir o banco de forma automatizada depois de especificadas suas alterações.
E124	Abordagem	Abordagem para a evolução de esquemas, que suporta a compatibilidade com as versões anteriores para isso é utilizado as <i>views</i> .
E143	Modelo	Modelo formal capaz de migrar o banco de dados partindo das refatores que foram criadas no código de aplicação.
E147	Abordagem	Abordagem baseada em representação genérica de esquemas e em uma definição formal de transformação e mapeamentos de esquemas.

Identificar a cadeia de ferramentas para gestão da evolução do banco de dados:

- Selecionar ferramentas de apoio à evolução (E21, E31, E47, E74, E83, E107, E123, E147) devem ser selecionadas de modo à prover automação ao processo de evolução, Sistemas (E36, E42, E106, E119) e Tecnologias (E23, E50, E121) são mencionados por quatro estudos e também são utilizados para realizar a evolução;
- Selecionar modelos como apoio à evolução de banco de dados. Seis estudos propõe modelos com a finalidade de expressar mais informações relacionadas à evolução (E7, E31, E65, E80, E95, E136);
- Selecionar boas práticas de evolução. Estudos que usam diferentes elementos de evolução conforme diferentes arquiteturas de software (E67, E129, E133), em mecanismos para apoiar alguma etapa como, por exemplo, a migração de dados entre um esquema antigo para um esquema evoluído (E8, E24);
- Apoiar a evolução juntamente com *Guidelines* (E5).

A.3.4 Identificar Possíveis Vantagens da Evolução de Banco de Dados

Estabelecer as metas organizacionais de acordo com possíveis atributos de qualidade de uma abordagem de evolução de banco de dados:

- Maior rastreabilidade entre evoluções e dados entre as tabelas;
- Maior praticidade na realização da evolução, já que algumas soluções não precisam especificar todos os dados relacionados às mudanças;
- Simplicidade, uma vez que algumas arquiteturas propostas tendem a simplificar toda a evolução do banco de dados;

- Adaptabilidade, que é um ponto forte apresentado foram as migrações de dados, onde torna-se possível migrar os dados entre versões dos bancos, o que sem estas tecnologias torna-se uma tarefa muito complicada;
- Corretude, já que atualmente existem soluções que podem sugerir correções a serem realizadas em cima do esquema do banco de dados e notificar automaticamente possíveis riscos e incompatibilidades, e;
- Manutenibilidade, já que a utilização de *views* permite isolar o problema da evolução com uma camada entre o sistema e o banco, assim quando o banco for evoluído o sistema não é totalmente afetado.

Automatizar o processo de evolução. Os seguintes atributos de qualidade são observados:

1. Apoiar o processo com o controle das versões do banco, com representação visual dessas versões no formato de grafo dirigido;
2. Controlar a versão dos dados. Implica em reduções de esforço, como no caso de Armazém de dados, existem propostas que tem o intuito de propagar alterações realizadas entre todos os seus subconjuntos, minimizando assim o esforço necessário para a evolução;
3. Coevolução de dados e esquemas SQL, já que devido a possibilidade de se conseguir realizar uma fácil relação entre as versões de um esquema de banco de dados, propostas de modelos para especificar transformações de modelos foram muito citadas;
4. Automações do processo, como encontrado em relatos de soluções apresentadas por estudos mais atuais, estão focadas no aumento de desempenho, implantação contínua e evolução do banco durante a manutenção do sistema, e;
5. Alterações em *runtime*, já que estas operações ocorrem sem que banco de dados fique fora do ar, ou seu usuário perceba e alta escalabilidade, fornecendo também outros atributos como liberdade e flexibilidade para realizar alterações.

A.3.5 Identificar Possíveis Desvantagens na Evolução de Banco de Dados

Mesmo com todos os vantagens citadas, ainda existem problemas dentro do cenário de evolução de banco de dados que devem ser considerados pela fábrica de software, como:

1. Imaturidade da tecnologia, já que observa-se uma falta de garantia que as tecnologias propostas funcionam em todos os casos, pois algumas não possuem avaliação ou até mesmo estão em fase de desenvolvimento;

2. Desempenho, uma vez que parte das abordagens não se preocupa com este fator, o que pode gerar gargalos e impactar diretamente no *software* que utiliza o banco de dados evoluído, além de viabilizar possíveis problemas nas consultas;
3. Processo manual para o controle da evolução de banco de dados. Algumas abordagens tentam reduzir a quantidade de escolhas cruciais que o mantenedor precisa fazer, entretanto este profissional ainda é responsável por verificar todo o risco de utilizar alguma das abordagens propostas ou evoluções indicadas;
4. Limitações de escopo, pois as soluções apresentadas muitas vezes estão vinculadas a uma tecnologia ou ferramenta específica o que pode acabar limitando o seu uso;
5. Inconsistências e perda de dados, pois mesmo que algumas soluções realizem a migração de dados, nem todas as necessidades estão mapeadas, podendo acarretar possíveis perdas de dados em alguns casos de coevolução de modelos e dados, e;
6. Incompatibilidade entre versões de sistemas que assistem a evolução, o que pode gerar um alto custo para manter a evolução do banco.

A.3.6 Identificar Possíveis Inovações para Introduzir na Organização

Sabendo dessas dificuldades no cenário de evolução, os estudos mapeados até o momento apresentam algumas tendências:

1. Transformações bidirecionais, ajudando na tradução de um modelo para o outro, gerando de alguma forma um rastro utilizável por sistemas de gerenciamento das atualizações;
2. Modularização, que para o contexto de banco de dados com módulos configura uma tendência de pesquisa. Pode-se destacar aqui a criação de alguma linguagem de definição de módulos para que assim possa se gerar ou extrair dados para a evolução do banco de acordo com os módulos desenvolvidos, além de ferramenta assistivas para aumentar o nível de abstração das entidades do sistema;
3. Isolamento do problema, uma vez que o uso de *views* se apresentou como prática promissora para gerenciar alterações e isolar os problemas decorrentes da execução de novas iterações do processo de desenvolvimento. Entretanto, faltam estudos que explorem diferentes técnicas de utilização de *views*;
4. Suporte ferramental como *plugins*, que caracteriza outra tendência para a inserção de tecnologias de evolução de banco nos SGBDs atuais;
5. *Guidelines*, projetadas com o intuito de ajudar o mantenedor a escolher qual a melhor estratégia de evolução. Uma possível proposta apresentada pelos estudos seria a criação de modelos que capturem as compensações de acordo com a escolha;

6. Visualização de dados, já que foram apresentadas também como promissor estudos tratando do aumento nas técnicas de visualização das evoluções e sua interdependência com artefatos do processo de software, e;
7. O suporte ferramental de evolução para diversos SGBDs, que pode tornar possível a realização de mais testes na indústria, juntamente com configurações dinâmicas.

Tabela 9 – Detalhes dos apoios à evolução de banco de dados

ID	Tipo	Descrição
E5	Outros	Quatro estratégias importantes para lidar com evolução, ou seja, manutenção antecipada, manutenção reversa, engenharia reversa e antecipação do projeto.
E7	Modelo	Modelo entidade relacionamento (ER)
E8	Mecanismo	Mecanismo de visualização aprimorado com recursos de importação/exportação.
E13	Abordagem	Abordagem para adaptar dinamicamente as visualizações de acordo com as mudanças de esquema.
E15	Abordagem	Abordagem detecta e avisa sobre riscos potenciais, como mudanças na estrutura de dados do passado, incompatibilidade de tipos de dados e perdas de dados.
E19	<i>Framework</i>	<i>Framework</i> que suporta a evolução simultânea de modelos de dados orientados a objetos e esquemas relacionais com respeito a um mapeamento relacional de objetos suportado por ferramentas.
E21	Ferramenta	IDE a qual é integrada.
E23	Tecnologia	Esquema Entidade-Relacionamento.
E24	Mecanismo	Mecanismo de apoio de ponta a ponta à evolução de esquema de banco de dados.
E25	Outros	Estratégias para evolução de esquema lento de longo prazo escalável em armazenamentos de dados NoSQL.
E31	Modelo	Modelo para registrar a evolução do esquema e um mecanismo para gerenciar catálogos e conjuntos de dados quando ocorrem mudanças
E32	Técnica	Técnicas de banco de dados para descobrir novas regras.
E33	Linguagem	Linguagem Prolog.
E36	Sistema	Sistema de gerenciamento de versão para evolução de esquema.
E39	Processo	Testes unitários e estratégias de teste.
E42	Sistema	DSS para sistemas de banco de dados (DBS) baseado na filosofia e metodologia de controle adaptativo e confiabilidade. Além disso, arquitetura de quatro esquemas que promete a independência do modelo de dados.
E47	Ferramenta	Ferramenta, a AutonomousDB, que utiliza conceitos de Computação Autônoma aplicada ao SGBDs, para auxiliar o DBA na tarefa de evolução do esquema e atualização de propagação

Tabela 9 – Detalhes dos apoios à evolução de banco de dados

ID	Tipo	Descrição
E50	Tecnologia	Instruções MOFScript com instruções SQL para realizar as transformações.
E59	Abordagem	Abordagem para medir possíveis impactos nas modificações e utiliza tecnologias como eclipse, QVT, <i>model transformations</i> e EMF.
E61	Abordagem	Abordagem (AppCatchupDB) para adaptar automaticamente aplicativos de <i>software</i> à evolução de seus bancos de dados.
E62	Outros	Plataforma que manipula a evolução de dados em bancos de dados orientados por colunas de maneira eficiente (CODS).
E65	Modelo	Modelo baseado em gráficos que captura, de maneira uniforme, relações, visualizações, restrições e consultas.
E67	Arquitetura	Arquitetura para gerenciar a evolução do banco de dados com uma estratégia avançada.
E68	<i>Framework</i>	Estrutura de evolução de esquemas baseada em uma abordagem algébrica (CHiSEL) que introduz operadores relacionais compostos estendidos e de nível superior adaptados à tarefa de evolução de esquemas.
E70	Outros	Metodologia para gerenciar evoluções de banco de dados, explorando duas ontologias.
E74	Ferramenta	Visual Studio apoia já que o MoDEF depende do mesmo para funcionar.
E80	Modelo	Modelo para retirar as informações de mapeamento do estado de tempo de execução do aplicativo.
E83	Ferramenta	PRISM, uma ferramenta que suporta a atividade temporal e suscetível a erros da evolução de esquema.
E92	<i>Framework</i>	Estrutura para gerenciar avidamente a evolução de esquema em armazenamentos de documentos NoSQL sem esquema.
E93	Processo	Processo para conduzir a evolução do esquema de banco de dados, estendendo a modularização do banco de dados para funcionar de maneira evolucionária.
E94	Abordagem	Abordagem paramétrica para resolver inconsistências de acordo com as necessidades do aplicativo.
E95	Modelo	Modelos conceituais para mudanças de esquema condicionais com muitas propriedades desejáveis que são essenciais para relações que são tuplas com diferentes esquemas.
103	Abordagem	Abordagem de duas fases a primeira compara o novo requisito com o MS (o que caracteriza um data mart) para detectar os novos elementos de requisito
E106	Sistema	Sistema chamado Tess que usa modelos para reconhecer mudanças de tipo comparando esquemas e então produz um transformador que pode atualizar dados em um banco de dados para corresponder a uma versão mais nova do esquema.

Tabela 9 – Detalhes dos apoios à evolução de banco de dados

ID	Tipo	Descrição
E107	Ferramenta	Ferramenta para realizar a evolução de bando de dados o QuantumDB, em desenvolvimento.
E108	<i>Framework</i>	<i>Framework</i> coerente baseado em gráficos para capturar o efeito de possíveis mudanças no <i>software</i> de banco de dados de um Sistema de Informações.
E110	Abordagem	Para realizar a criação das tabelas fantasma e criação das views a abordagem utiliza da ferramenta QuantumDB.
E113	Outros	PRISM ++, um <i>Schema Modification Operators</i> (SMO), oferece suporte de ponta a ponta para a adaptação de consultas e atualizações através de mudanças estruturais de esquema e evolução de restrições de integridade.
E116	Outros	Lógica epistêmica é a lógica do conhecimentos. Trata dos mecanismos de raciocínio do conhecimento e do processo de revisão de uma base de crenças.
E118	Processo	Processo, com passo a passo e o que fazer para realizar a evolução do banco de dados utilizando refatoração.
E119	Sistema	Sistema que permite a exploração de múltiplas versões de esquema (VESEL) usando consultas de proveniência.
E121	Tecnologia	Repositório de esquemas que é utilizado juntamente com uma Abordagem Preditiva para Evolução do Esquema do Banco de Dados.
E123	Ferramenta	Ferramenta de <i>software</i> que permite a visualização do impacto da evolução do esquema através do uso de gatilhos e procedimentos armazenados.
E129	Arquitetura	Arquitetura de metadados <i>Multi-version</i> .
E133	Arquitetura	MeDEA, uma arquitetura de evolução de banco de dados com propriedades de rastreabilidade.
E136	Modelo	Modelo axiomático que fornece uma solução geral para o problema de suportar a evolução de um esquema de banco de dados XML.
E138	Processo	Processo para apoiar a evolução do banco de dados propondo uma solução para reduzir a intervenção humana necessária para manter os processos de troca de dados após a evolução do esquema..
E147	Ferramenta	DB-MAIN , uma ferramenta de modelagem genérica dedicada à engenharia de aplicações de bancos de dados e, em particular, ao design de banco de dados, engenharia reversa, reengenharia, integração, manutenção e evolução.
E157	Ferramenta	Ferramenta de projeto de banco de dados chamada de EDO (<i>Evolutionary Database Optimizer</i>) que tem o objetivo de gerar um conjunto inicial de representações internas preliminares para um dado modelo de dados conceitual.

A.4 CONSIDERAÇÕES FINAIS

Devido à necessidade de práticas de entrega contínua nas fábricas de *software* por meio de *DevOps* (LORENZ; HESSE; RUDOLPH, 2016), os processos precisam de automação. Isso impõe um novo desafio à equipe de desenvolvimento: automatizar as atividades de evolução de banco de dados. Tal necessidade motiva as equipes pela busca da perfeição nas tarefas de gerência de configuração, desenvolvimento e teste de sistemas. Estas, por sua vez, são afetadas diretamente por outras atividades de evolução do sistema, temas de estudos exploratórios de outras *guidelines*.

Analisando o contexto prático da indústria na automação dos processos de desenvolvimento (REDGATE, 2018), fica uma dúvida: o que é possível automatizar nas atividades de evolução de banco de dados de modo a agregar agilidade em processos de Integração, Implantação e Entrega Contínua? A sua tarefa é, portanto, adaptar as atividades previstas em seu processo de desenvolvimento para incluir atividades relacionadas com evolução de banco de dados.

A integração contínua e a implantação contínua apresentam a necessidade no cenário atual de formas automatizadas que possibilitem a evolução do banco de dados, ou apoiar alguma etapa muito complexa desta atividade. Esta é considerada a principal tendência atual, alguns estudos já começaram a trabalhar neste rumo, entretanto se mostraram limitados às tecnologias ou banco de dados específicos. Trata-se, portanto, de um problema relevante pelo qual os alunos podem conduzir melhorias em seus processos e práticas de desenvolvimento de software.

A.4.1 Práticas Mínimas Esperadas para a Gerência de Configuração para um Marco de Cinco Semanas

Você deve ser capaz de rapidamente configurar o banco de dados. Assim, defina as estratégias evolutivas do time levando em consideração recomendações mínimas como:

- Deve-se gravar (versionar) cada alteração do banco de dados (Maior, menor, da Sprint, da *release* derivada da iteração);
- Identifique cada mudança como um item único no versionador: a) Versão do banco de dados ou de *schemas*; b) Identificador único do *script* do banco de dados; c) Quando utilizar APIs evolutivas no desenvolvimento (por exemplo, usando JPA e Hibernate para automatizar a persistência dos dados com mapeamento objeto-relacional), utilize um timestamp à cada nova geração do *script*.
- Inclua tabelas de configuração no banco de dados: a) para guardar a versão do banco de dados ou de *schemas*; e b) Log de eventos de mudança.
- Aplique princípios de Domain Driven Design (DDD);

- Coloque todos os arquivos das mudanças num gerenciador de mudanças (Git, SVN, CVS);
- Uma vez que se realizou o *commit* do *script*, não modifique-o. Em vez disso, faça uma nova versão com um novo identificador, pois é fundamental manter versionado o *schema* do banco de dados;
- Automatize a execução dos seus *scripts* de banco de dados.

A.4.2 Como Iniciar?

O primeiro passo é tomar conhecimento de todos os elementos associados com evolução de banco de dados, o que acreditamos ter fornecido neste guideline.

Num segundo momento é preciso identificar quais elementos serão incluídos numa processo de desenvolvimento.

O terceiro passo é definir uma *baseline* do banco de dados. Ou seja, você precisa definir uma configuração de referência. Para maiores informações sobre gerência de configuração, consulte a apresentação da Prof^a. Dr^a. Elisa que trata sobre gerenciamento de configuração de software ².

Para estruturar e criar uma *baseline*, aqui³ constam uns exemplos no SGBD Oracle. Busque informações de como proceder para o banco selecionado no seu time de desenvolvimento.

Após a adaptação do processo, implemente um processo de versionamento do banco de dados e aplique as técnicas/estratégias evolutivas. Lembre-se que fazer isso não é fácil e que relatos informam que se leva até três *sprints* para se ter algo totalmente operacional. A alta gestão geralmente será contra a automação porque isso traz custos. Logo, aponte quais os vantagens e possíveis desvantagens a introdução das práticas de automação podem levar à organização.

Para os times que adotaram alguma implementação como JPA e Hibernate, existem algumas opções de ferramentas para realizar parte das atividades de evolução de banco de dados como:

- Fluent Migrator⁴;
- Visual Studio Database Projects;
- Liquibase⁵;
- MongoDB Atlas.

² https://edisciplinas.usp.br/pluginfile.php/314971/mod_resource/content/1/Aula07_Gerenciamento%20de%20Configuracao

³ <https://www.oracle.com/br/technical-resources/articles/sql/sql-plan-baselines-sql-plus.html>

⁴ <https://fluentmigrator.github.io/articles/quickstart.html?tabs=runner-in-process>

⁵ <https://www.liquibase.org/>

Finalmente, é preciso sincronizar a *release* do banco com *releases* do software com mecanismos de Integração, Implantação e Entrega Contínua. Neste ponto já se percebe a necessidade de outro *guideline*, para tratarmos um pouco mais adiante.

ÍNDICE

API, 30

BDOO, 31

CE, 37

CI, 37

CQ, 37

DC, 23, 24, 49, 60

DSS, 53, 100

EC, 23, 24, 61

ER, 49

ETL, 67

IC, 23, 24, 60, 61

IDE, 49, 54, 100

MDD, 59

NoSQL, 30, 31, 37, 49, 51, 54, 55, 95, 96,
100, 101

QP, 35

RP, 25

SGBD, 51, 53, 59, 64, 96, 100

SMO, 50

SMS, 26, 33

SPL, 49

SQL, 30, 49–51, 53, 58, 96, 101

TCC, 24–26