

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

RAFAEL PEDRONI VIÉGAS

**Aplicação de Conceitos de Bancos de Dados
de Grafos e Relacional na Criação de
Proposta e Análise Comparativa de
Abordagens para Armazenamento de
Processos**

Dissertação apresentada como requisito parcial
para a obtenção do grau de Mestre em Ciência da
Computação

Orientador: Profa. Dra. Renata Galante

Porto Alegre
2018

CIP — CATALOGAÇÃO NA PUBLICAÇÃO

Viéguas, Rafael Pedroni

Aplicação de Conceitos de Bancos de Dados de Grafos e Relacional na Criação de Proposta e Análise Comparativa de Abordagens para Armazenamento de Processos / Rafael Pedroni Viéguas. – Porto Alegre: PPGC da UFRGS, 2018.

122 f.: il.

Dissertação (mestrado) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação, Porto Alegre, BR-RS, 2018. Orientador: Renata Galante.

1. Armazenamento em Gerenciamento de Processos de Negócio, bancos de dados orientados a grafos, bancos de dados relacionais. I. Galante, Renata. II. Título.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. Rui Vicente Oppermann

Vice-Reitora: Prof^a. Jane Fraga Tutikian

Pró-Reitor de Pós-Graduação: Prof. Celso Giannetti Loureiro Chaves

Diretora do Instituto de Informática: Prof^a. Carla Maria Dal Sasso Freitas

Coordenador do PPGC: Prof. João Luiz Dihl Comba

Bibliotecária-chefe do Instituto de Informática: Beatriz Regina Bastos Haro

*“Aqueles que se sentem satisfeitos sentam-se e nada fazem.
Os insatisfeitos são os únicos benfeitores do mundo.”*

— WALTER S. LANDOR

A Proposal for Storage of Processes between Different Databases

ABSTRACT

Business Process Management (BPM) area has been increasingly attracted the interest of the business community because users are looking for documentation and optimization. These documents can be an important method in helping to gain results such as reduced costs and increased productivity. However, to model processes is not enough. It is necessary to pay attention to efficient storage methods, allowing information to be handled and used in a practical and intelligent way. The present article compares the use of relational databases and graph databases, considering aspects such as performance in the execution of operations and proximity of the approach of each of them with the process models. While relational databases are more popular, being used in most of the current applications, graph databases have properties and graphical representations similar to process models. The results of this study can be used to create repositories which can both share process efficiently, and encourage the study of new ways of storing processes.

Keywords: Business Process Management storage, graph databases, relational databases.

RESUMO

Em busca da documentação e otimização de seus processos, a área de *Business Process Management* (BPM) vem cada vez mais atraindo o interesse do meio empresarial, por ser um importante método no auxílio ao ganho de resultados, como redução de custos e aumento de produtividade. Modelar processos, entretanto, não basta. É preciso que se atente para métodos eficientes de armazená-los, permitindo que as informações sejam manipuladas e utilizadas de maneira prática e inteligente. A presente dissertação propõe duas abordagens para armazenamento de modelos de processo, uma em bancos de dados relacionais e outra em bancos de dados orientados a grafos, comparando-os através de aspectos como desempenho na execução das operações e proximidade da abordagem de cada um deles com os modelos de processos. Enquanto os bancos de dados relacionais são mais populares, sendo utilizados na maior parte das aplicações atuais, os bancos de dados orientados a grafos possuem propriedades e representação gráfica semelhantes aos modelos de processos. Foram realizados testes que visam analisar o desempenho de ambas as abordagens, além da facilidade dos usuários em interagir com os modelos propostos. Os resultados deste estudo podem ser utilizados para a criação de repositórios que compartilhem processos de maneira eficiente, bem como incentivar o estudo de novas maneiras para o armazenamento de processos.

Palavras-chave: Armazenamento em Gerenciamento de Processos de Negócio, bancos de dados orientados a grafos, bancos de dados relacionais.

LISTA DE ABREVIATURAS E SIGLAS

ACM	<i>Association for Computing Machinery</i>
API	<i>Application Programming Interface</i>
BPM	<i>Business Process Management</i>
BPEL	<i>Business Process Execution Language</i>
BPMN	<i>Business Process Model Notation</i>
GPL	<i>General Public License</i>
HTTP	<i>Hypertext Transfer Protocol</i>
IEEE	<i>Institute of Electrical and Electronics Engineers</i>
ICIJ	<i>International Consortium of Investigative Journalists</i>
JSON	<i>JavaScript Object Notation</i>
OMG	<i>Object Management Group</i>
PHP	<i>Hypertext Preprocessor</i>
SBC	Sociedade Brasileira de Computação
SGBD	Sistema de Gerenciamento de Banco de Dados
SQL	<i>Structured Query Language</i>
SUS	<i>System Usability Scale</i>
XML	<i>Extensible Markup Language</i>
YAWL	<i>Yet Another Workflow Language</i>

LISTA DE FIGURAS

Figura 1.1 Participação de diferentes modelos de bancos de dados no mercado global (ASAY, 2016).....	15
Figura 2.1 Exemplo de processo representado na notação BPMN (IPROCESS, 2012). 21	
Figura 3.1 Modelo de processo utilizado na exemplificação de mapeamentos dos elementos. (SGANDERLA, 2012)	33
Figura 3.2 Representação do modelo relacional projetado para o armazenamento de processos.....	36
Figura 3.3 Exemplo de aresta associando uma tarefa a um operador XOR, em verde. Circulado em vermelho, exemplo de um ponto pertencente a uma aresta.....	37
Figura 3.4 Preenchimento da tabela “processos”, com os dados de identificação do modelo de processo.	42
Figura 3.5 Armazenamento dos objetos de fluxo contidos no modelo de processo.	42
Figura 3.6 Armazenamento dos objetos de conexão contidos no modelo de processo. .42	
Figura 3.7 Armazenamento dos pontos dos objetos de fluxo como foram plotados na ferramenta de modelagem.	43
Figura 3.8 Exemplo de funcionamento do mapeamento de modelos de processos para grafos.	45
Figura 3.9 Exemplo de grafo que representa um processo mapeado.....	48
Figura 4.1 Modelo de processo utilizado como exemplo na avaliação experimental.	56
Figura 4.2 Gráfico representando relação entre número de elementos do modelo de processo (eixo X) e seus respectivos tempos de armazenamento em milissegundos (eixo Y).....	58
Figura 4.3 Gráfico representando relação entre número de elementos do fragmento de processo buscado (eixo X) e seus respectivos tempos de busca em milissegundos (eixo Y).....	60
Figura 4.4 Gráfico representando relação entre número de elementos do fragmento de processo buscado (eixo X) e seus respectivos tempos de busca das informações para reconstrução em milissegundos (eixo Y).	62
Figura 4.5 Processo de tamanho pequeno, utilizado como base para a criação de questões utilizadas no experimento.....	67
Figura 4.6 Processo de tamanho médio, utilizado como base para a criação de questões utilizadas no experimento.....	68
Figura 4.7 Classificação dos participantes de acordo com a faixa etária e nível de escolaridade.....	69
Figura 4.8 Classificação dos participantes de acordo com os conhecimentos em bancos de dados relacionais, bancos de dados orientados a grafos e BPM.	70
Figura 4.9 Modelo de processo armazenado em banco de dados MySQL utilizado na questão 1 do experimento de armazenamento de processos em banco de dados relacional.....	71
Figura 4.10 Alternativa correta para a primeira questão do experimento de armazenamento de processos em banco de dados relacional.....	72

Figura 4.11 Exemplo de alternativa incorreta para a primeira questão do experimento de armazenamento de processos em banco de dados relacional. O círculo em vermelho destaca a inversão da posição dos elementos Task e Sub Process originalmente utilizados no modelo de processo, tornando a alternativa incorreta.....	72
Figura 4.12 Performance dos usuários na avaliação de armazenamento de modelos de processo em banco de dados relacional.....	73
Figura 4.13 Modelo de processo armazenado em banco de dados Neo4j utilizado na questão 1 do experimento de armazenamento de processos em banco de dados orientado a grafos.....	74
Figura 4.14 Alternativa correta para a primeira questão do experimento de armazenamento de processos em banco de dados orientado a grafos.....	74
Figura 4.15 Exemplo de alternativa incorreta para a primeira questão do experimento de armazenamento de processos em banco de dados orientado a grafos. Os círculos em vermelho destacam a substituição dos elementos XOR originalmente utilizado no modelo de processo, por elementos AND.....	75
Figura 4.16 Performance dos usuários na avaliação de armazenamento de modelos de processo em banco de dados orientado a grafos.....	76
Figura 4.17 Consulta de fragmento de modelo de processo armazenado em banco de dados MySQL utilizado na questão 1 do experimento de consulta de processos em banco de dados relacional.	77
Figura 4.18 Alternativa correta para a primeira questão do experimento de consulta de modelos de processos em banco de dados relacional. O círculo vermelho destaca o fragmento buscado.....	77
Figura 4.19 Exemplo de alternativa incorreta para a primeira questão do experimento de consulta de modelos de processos em banco de dados relacional. O fragmento destacado não se encontra no modelo de processo.	78
Figura 4.20 Performance dos usuários na avaliação de consultas de modelos de processo em banco de dados relacional.....	79
Figura 4.21 Consulta de fragmento de modelo de processo armazenado em banco de dados Neo4j utilizado na questão 1 do experimento de consulta de processos em banco de dados orientado a grafos.	79
Figura 4.22 Alternativa correta para a primeira questão do experimento de consulta de modelos de processos em banco de dados orientados a grafos. O círculo vermelho destaca o fragmento buscado.....	80
Figura 4.23 Exemplo de alternativa incorreta para a primeira questão do experimento de consulta de modelos de processos em banco de dados orientado a grafos. O fragmento destacado não se encontra no modelo de processo.	81
Figura 4.24 Performance dos usuários na avaliação de consultas de modelos de processo em banco de dados orientado a grafos.	82
Figura B.1 Representação de um modelo de processo.....	95
Figura B.2 Representação de um modelo de processo armazenado em banco de dados não-relacional.....	95
Figura B.3	96
Figura B.4	96
Figura B.5	97
Figura B.6 Questão 1 do experimento sobre armazenamento em bancos de dados orientados a grafos	97
Figura B.7 Primeira opção da questão 1 do experimento sobre armazenamento em bancos de dados orientados a grafos	98

Figura B.8 Segunda opção da questão 1 do experimento sobre armazenamento em bancos de dados orientados a grafos	98
Figura B.9 Terceira opção da questão 1 do experimento sobre armazenamento em bancos de dados orientados a grafos	99
Figura B.10 Quarta opção da questão 1 do experimento sobre armazenamento em bancos de dados orientados a grafos	99
Figura B.11 Questão 2 do experimento sobre armazenamento em bancos de dados orientados a grafos	100
Figura B.12 Primeira opção da questão 2 do experimento sobre armazenamento em bancos de dados orientados a grafos	100
Figura B.13 Segunda opção da questão 2 do experimento sobre armazenamento em bancos de dados orientados a grafos	101
Figura B.14 Terceira opção da questão 2 do experimento sobre armazenamento em bancos de dados orientados a grafos	102
Figura B.15 Quarta opção da questão 2 do experimento sobre armazenamento em bancos de dados orientados a grafos	103
Figura B.16 Representação de um modelo de processo.....	104
Figura B.17 Representação de um modelo de processo armazenado em banco de dados relacional.....	105
Figura B.18	106
Figura B.19	106
Figura B.20	107
Figura B.21	107
Figura B.22 Questão 1 do experimento sobre armazenamento em bancos de dados relacionais.....	108
Figura B.23 Primeira opção da questão 1 do experimento sobre armazenamento em bancos de dados relacionais	108
Figura B.24 Segunda opção da questão 1 do experimento sobre armazenamento em bancos de dados relacionais	109
Figura B.25 Terceira opção da questão 1 do experimento sobre armazenamento em bancos de dados relacionais	109
Figura B.26 Quarta opção da questão 1 do experimento sobre armazenamento em bancos de dados relacionais	109
Figura B.27 Questão 2 do experimento sobre armazenamento em bancos de dados relacionais.....	110
Figura B.28 Primeira opção da questão 2 do experimento sobre armazenamento em bancos de dados relacionais	111
Figura B.29 Segunda opção da questão 2 do experimento sobre armazenamento em bancos de dados relacionais	111
Figura B.30 Terceira opção da questão 2 do experimento sobre armazenamento em bancos de dados relacionais	112
Figura B.31 Quarta opção da questão 2 do experimento sobre armazenamento em bancos de dados relacionais	112
Figura B.32 Representação de um modelo de processo.....	113
Figura B.33	113
Figura B.34	113
Figura B.35	114
Figura B.36 Questão 1 do experimento sobre consulta por fragmentos em bancos de dados orientados a grafos	114
Figura B.37 Primeira opção da questão 1 do experimento sobre consulta por fragmentos em bancos de dados orientados a grafos.....	114

Figura B.38 Segunda opção da questão 1 do experimento sobre consulta por fragmentos em bancos de dados orientados a grafos.....	114
Figura B.39 Terceira opção da questão 1 do experimento sobre consulta por fragmentos em bancos de dados orientados a grafos.....	115
Figura B.40 Quarta opção da questão 1 do experimento sobre consulta por fragmentos em bancos de dados orientados a grafos.....	115
Figura B.41 Questão 2 do experimento sobre consulta por fragmentos em bancos de dados orientados a grafos	115
Figura B.42 Primeira opção da questão 2 do experimento sobre consulta por fragmentos em bancos de dados orientados a grafos.....	116
Figura B.43 Segunda opção da questão 2 do experimento sobre consulta por fragmentos em bancos de dados orientados a grafos.....	116
Figura B.44 Terceira opção da questão 2 do experimento sobre consulta por fragmentos em bancos de dados orientados a grafos.....	117
Figura B.45 Quarta opção da questão 2 do experimento sobre consulta por fragmentos em bancos de dados orientados a grafos.....	117
Figura B.46 Representação de um modelo de processo.....	118
Figura B.47	118
Figura B.48	118
Figura B.49	119
Figura B.50 Questão 1 do experimento sobre consulta por fragmentos em bancos de dados relacionais	119
Figura B.51 Primeira opção da questão 1 do experimento sobre consulta por fragmentos em bancos de dados relacionais.....	119
Figura B.52 Segunda opção da questão 1 do experimento sobre consulta por fragmentos em bancos de dados relacionais.....	120
Figura B.53 Terceira opção da questão 1 do experimento sobre consulta por fragmentos em bancos de dados relacionais.....	120
Figura B.54 Quarta opção da questão 1 do experimento sobre consulta por fragmentos em bancos de dados relacionais.....	120
Figura B.55 Questão 2 do experimento sobre consulta por fragmentos em bancos de dados relacionais	121
Figura B.56 Primeira opção da questão 2 do experimento sobre consulta por fragmentos em bancos de dados relacionais.....	121
Figura B.57 Segunda opção da questão 2 do experimento sobre consulta por fragmentos em bancos de dados relacionais.....	121
Figura B.58 Terceira opção da questão 2 do experimento sobre consulta por fragmentos em bancos de dados relacionais.....	122
Figura B.59 Quarta opção da questão 2 do experimento sobre consulta por fragmentos em bancos de dados relacionais.....	122

LISTA DE TABELAS

Tabela 2.1	Categorias dos trabalhos relacionados.....	28
Tabela 2.2	Comparativo entre os principais trabalhos relacionados.	31
Tabela 3.1	Comparativo entre os principais trabalhos relacionados e os métodos propostos.	50
Tabela 4.1	Comparativo entre o modelo relacional e o orientado a grafos nos experimentos de armazenamento de modelos de processos.....	80
Tabela 4.2	Comparativo entre o modelo relacional e o orientado a grafos nos experimentos de consultas de fragmentos de modelos de processos.	80
Tabela 4.3	Comparação de resultados do método SUS para os experimentos de satisfação realizados.....	81

SUMÁRIO

1 INTRODUÇÃO	14
1.1 Objetivos e Contribuições	16
1.2 Organização da Dissertação	17
2 FUNDAMENTAÇÃO TEÓRICA E TRABALHOS RELACIONADOS	18
2.1 Modelos de Processos e Novas Tecnologias.....	18
2.1.1 Modelos de Processos	18
2.1.1.1 A Notação BPMN	19
2.1.1.2 Armazenamento de Modelos de Processo	21
2.1.2 NoSQL	21
2.1.3 Neo4j.....	23
2.2 Trabalhos Relacionados.....	25
2.2.1 Principais Questionamentos.....	25
2.2.2 Método de Pesquisa	26
2.2.3 Trabalhos em Destaque.....	27
2.2.4 Considerações Finais	30
3 MAPEANDO MODELOS DE PROCESSOS PARA MODELOS LÓGICOS DE BANCOS DE DADOS RELACIONAL E DE GRAFOS	32
3.1 Modelo de Processo: Exemplo e Considerações.....	32
3.2 Armazenamento Relacional	33
3.2.1 Mapeamento de Elementos	34
3.2.2 Modelo de Dados	35
3.2.3 Restrições de Integridade	37
3.2.4 Algoritmo de Mapeamento de Modelo de Processo para Grafo Equivalente.....	38
3.2.5 Exemplo de armazenamento	41
3.3 Armazenamento em Bancos de Dados Orientados a Grafos	42
3.3.1 Mapeamento de elementos.....	43
3.3.2 Modelo de dados	44
3.3.3 Algoritmo de Mapeamento de Modelo de Processo para Banco Relacional.....	45
3.3.4 Exemplo de armazenamento	47
3.4 Considerações Finais	48
4 AVALIAÇÃO EXPERIMENTAL	51
4.1 Experimentos de Mapeamento Físico dos Processos	51
4.1.1 Cenário de Avaliação	52
4.1.2 Base de Dados.....	53
4.1.3 Mapeamento para o banco de dados físico MySQL	53
4.1.4 Mapeamento para o banco de dados físico Neo4j	54
4.1.5 Avaliação do Armazenamento de Processos.....	55
4.1.6 Avaliação da Busca por Fragmento de Processos.....	59
4.1.7 Avaliação da Reconstrução de Processos.....	61
4.1.8 Análise Geral dos Resultados	63
4.2 Experimentos com Usuários.....	64
4.2.1 Procedimento	64
4.2.2 Participantes	67
4.2.3 Avaliação de Armazenamento em Banco de Dados Relacional	68
4.2.3.1 Performance	69
4.2.3.2 Satisfação	70
4.2.4 Avaliação de Armazenamento em Banco de Dados Orientado a Grafos.....	71
4.2.4.1 Performance	72

4.2.4.2 Satisfação	73
4.2.5 Avaliação de Consultas em Banco de Dados Relacional	74
4.2.5.1 Performance	75
4.2.5.2 Satisfação	75
4.2.6 Avaliação de Consultas em Banco de Dados Orientado a Grafos	76
4.2.6.1 Performance	77
4.2.6.2 Satisfação	78
4.3 Análise Geral dos Resultados.....	78
5 CONCLUSÃO	83
REFERÊNCIAS	85
APÊNDICE A — SCRIPT EM LINGUAGEM SQL PARA MAPEAMENTO DE MODELOS DE PROCESSO PARA O BANCO DE DADOS MYSQL 88	
APÊNDICE B — DOCUMENTOS USADOS NAS AVALIAÇÕES DE USUÁRIOS93	
B.1 Perfil do Usuário.....	93
B.2 Questionário SUS - System Usability Scale	94
B.3 Avaliação de Armazenamento de Modelo de Processo em Banco Orien- tado a Grafos	94
B.3.1 Treinamento Realizado.....	94
B.3.2 Questão 1	97
B.3.3 Questão 2.....	97
B.4 Avaliação de Armazenamento de Modelo de Processo em Banco Relacional ..	97
B.4.1 Treinamento Realizado.....	97
B.4.2 Questão 1	101
B.4.3 Questão 2.....	101
B.5 Consulta de Modelo de Processo em Banco de Dados Orientado a Grafos	101
B.5.1 Treinamento Realizado.....	101
B.5.2 Questão 1	103
B.5.3 Questão 2.....	103
B.6 Consulta de Modelo de Processo em Banco de Dados Relacional	103
B.6.1 Treinamento Realizado.....	103
B.6.2 Questão 1	105
B.6.3 Questão 2.....	105

1 INTRODUÇÃO

Obter uma posição de destaque no mercado exige que as empresas forneçam serviços ou produtos diferenciados, a preços competitivos e capazes de, não apenas chamar a atenção de seus clientes, mas também fidelizá-los. Para alcançar estes objetivos, entretanto, é necessário que a organização interna da corporação seja qualificada, permitindo a redução de custos, o aproveitamento melhor de recursos e o aumento de produtividade. Neste contexto, um número crescente de empresas vem tomando consciência da importância de documentar e promover melhorias em seus processos na busca de melhores resultados. Empresas responsáveis por promover boletins de estudos de mercado e análise industrial, tais como a *Orbis Research* (RESEARCH, 2017) e a *Markets And Markets* (MARKETS; MARKETS, 2016), estimam que o mercado de *Business Process Management* (BPM) tenha uma taxa de crescimento anual entre 13,52% e 14,2% entre 2016 e 2021. Isto significaria um crescimento, em valor de mercado, de 6,96 bilhões de dólares em 2016, para até 14,89 bilhões de dólares em 2021. Estes fatores fazem com que o gerenciamento de processos de negócio se consolide como um assunto extremamente importante no meio acadêmico e empresarial.

Para que os processos sejam otimizados, é fundamental conhecê-los da maneira mais detalhada possível. Mapeá-los e documentá-los é imprescindível para que esse objetivo seja alcançado, além de possibilitar que o conhecimento a respeito do funcionamento da empresa se torne perene. Esta nem sempre é uma tarefa trivial, uma vez que, em geral, os processos são extensos, dificultando a identificação de fragmentos semelhantes e, portanto, reutilizáveis. Além disso, tais processos devem ser dinâmicos, já que são constantemente modificados para atender as novas necessidades da empresa. É preciso, então, que se atente não apenas para formas eficientes de modelar, como também de armazenar os processos, possibilitando efetuar operações como a busca, comparação e compartilhamento de informações entre pessoas de uma mesma organização ou até mesmo de organizações diferentes, de maneira facilitada. Neste contexto, analisar os diferentes meios e tecnologias disponíveis para o armazenamento de processos, bem como definir as vantagens de cada um deles, torna-se uma significativa proposta. Executar operações de maneira eficiente é primordial, por interferir positivamente na redução de custos para a implantação de sistemas de BPM.

A análise de trabalhos relacionados a BPM permite classificá-los entre diversas categorias, como modelos de processos sendo armazenados em nuvem, criação de repo-

sitórios de processos, detecção de similaridades entre dois processos e artigos do tipo *survey*, que fornecem panoramas a respeito da área atualmente. Apesar de uma diversidade de abordagens, poucos tratam especificamente do armazenamento de processos em banco de dados. São ainda mais raros os que avaliam técnicas diferentes de armazenamento, que não o modelo de banco de dados relacional. Esta tendência à utilização do modelo relacional é compreensível, uma vez que se trata do mais consolidado entre os existentes, responsável pela grande fatia de mercado de banco de dados. Entretanto, estudos apontam um rápido crescimento dos bancos de dados NoSQL, que deixaram de ser responsáveis por uma insignificante parcela do mercado em 2012, para tomarem conta, junto com o Hadoop, de cerca de 3% do total do mercado. A Figura 1.1, retirada de (ASAY, 2016), exhibe este crescimento. Além disso, espera-se que o mercado de bancos de dados NoSQL cresça a uma taxa anual de 35,1% entre 2014 e 2020 (PERSON, 2015).

Figura 1.1: Participação de diferentes modelos de bancos de dados no mercado global (ASAY, 2016).

Exhibit 5
Hadoop and NoSQL accounted for only ~3% of total market, but are expected to grow rapidly going forward.



Source: IDC, Bernstein analysis

Diante de tamanha expansão de outras tecnologias para armazenamento de dados, diferentes do modelo relacional, aliado ao crescimento contínuo da área de BPM, torna-se fundamental vincular ambos os assuntos e buscar soluções que agreguem ao gerenciamento de processos. Analisar possíveis benefícios que a utilização de bancos de dados NoSQL pode trazer à área de BPM é um nicho importante a ser explorado, a fim de investigar sobre como as características peculiares aos bancos não relacionais podem ser benéficas nas principais operações que envolvem os processos. Dentre os diferentes tipos de bancos de dados NoSQL, um deles se destaca de maneira importante neste contexto:

o banco de dados baseado em grafos. Apesar de se mostrar o que possui menor fatia de mercado entre seus relacionados (PERSON, 2015), sua relevância se dá por possuir características bastante semelhantes aos processos. A facilidade em expressar explicitamente relações, atributo tão presente nos modelos de processos, é algo que leva a crer que é possível obter ganhos tanto em simplicidade de representação, como até de performance, uma vez que as operações tendem a ser menos custosas.

1.1 Objetivos e Contribuições

O presente trabalho tem por objetivo propor mapeamentos de modelos de processos para que sejam armazenados em bancos de dados relacionais e orientados a grafos, comparando-os através de aspectos como desempenho na execução das operações, proximidade da abordagem de cada um deles com os modelos de processos, facilidade de expressar consultas e possibilidade prática de identificação visual dos processos. A proposta é comparar o desempenho no tempo de execução das três principais operações, o armazenamento e a recuperação dos modelos de processo e a busca por fragmentos, detectando a existência de determinada parte de um processo no banco de dados. Também é comparada a praticidade para manipular e identificar os dados em cada um dos modelos estudados.

Para a criação das propostas, foi realizado um levantamento bibliográfico, buscando artigos relevantes em conceituados repositórios eletrônicos, tais como Scopus¹, IEEE Xplore², ACM³ e Web of Science⁴. Enquanto os bancos de dados relacionais são mais populares, sendo utilizados na maior parte das aplicações atuais, o banco de dados baseado em grafos possui propriedades e representação gráfica semelhantes aos modelos de processos. Os resultados deste estudo podem ser utilizados para a criação de repositórios que compartilhem processos de maneira eficiente, bem como incentivar o estudo de novas maneiras para o armazenamento de processos.

¹<https://www.scopus.com>

²<https://ieeexplore.ieee.org>

³<http://dl.acm.org>

⁴<https://webofknowledge.com>

1.2 Organização da Dissertação

O restante da dissertação está organizada como segue. O Capítulo 2 aborda os principais conceitos necessários para a compreensão da totalidade do trabalho proposto, incluindo um estudo sobre importantes trabalhos relacionados ao assunto, capazes de auxiliar na identificação dos principais obstáculos existentes na criação de soluções para armazenamento de processos. Uma proposta para armazenamento de processos em diferentes tipos de bancos de dados é realizada no Capítulo 3, seguido de uma explicação a respeito das experiências realizadas e de uma análise dos resultados obtidos, no Capítulo 4. O trabalho é finalizado com a descrição das conclusões obtidas durante seu desenvolvimento e apontamento de desdobramentos desta pesquisa e trabalhos futuros, presente no Capítulo 5.

2 FUNDAMENTAÇÃO TEÓRICA E TRABALHOS RELACIONADOS

Neste capítulo, são apresentados os principais conceitos necessários para a compreensão deste trabalho. Também tem como objetivo fornecer um panorama a respeito dos principais questionamentos levantados pelo estudo, bem como os trabalhos relacionados e o método de busca utilizado para coletá-los. Apresenta, ainda, os trabalhos considerados de maior destaque e relevância, dentre àqueles que possuem relação com o assunto abordado.

A Seção 2.1 aborda os conceitos relacionados aos modelos de processos e novas tecnologias, esclarecendo as principais definições na área de modelagem de processos e de bancos de dados NoSQL, em especial o Neo4j. A Seção 2.2, por sua vez, aborda os principais questionamentos que fundamentam a pesquisa, apontando os principais interesses com relação ao tema e a motivação de cada um deles. Também aponta quais os caminhos percorridos para que fossem encontrados trabalhos relevantes sobre o tema, que representassem, de fato, o estado da arte no armazenamento de processos. Por fim, promove um estudo mais aprofundado a respeito dos trabalhos encontrados, destacando os que trazem contribuições mais significativas e descrevendo maiores detalhes a respeito deles.

2.1 Modelos de Processos e Novas Tecnologias

2.1.1 Modelos de Processos

Processos se fazem presentes de forma contínua no cotidiano das pessoas. Seja para executar tarefas mais complexas ou mesmo as mais simples, é realizado uma sequência de passos a fim de completar os objetivos. Uma simples compra em um supermercado, por exemplo, envolve atividades como escolher o produto, levá-lo até o caixa, fazer a escolha entre dinheiro ou cartão e realizar o pagamento. Esta quantidade de processos, entretanto, é ainda maior dentro das organizações – nas quais se incluem empresas, órgãos governamentais, entre outros – onde todos eles devem ser bem gerenciados para o alcance de melhores resultados. Esta, porém, é uma empreitada difícil, uma vez que envolve diversas pessoas, com diferentes níveis de conhecimento e uma grande quantidade de metas a serem cumpridas.

Para auxiliar na documentação, organização e melhoria dos processos de uma or-

ganização, a área de BPM fornece uma série de elementos capazes de representar o fluxo de atividades realizadas dentro de um processo. (DUMAS et al., 2013) definem BPM como um conjunto de métodos, técnicas e ferramentas para descobrir, analisar, redesenhar, executar e monitorar processos de negócio. Processo de negócio, por sua vez, é definido como uma coleção de eventos, atividades e pontos de decisão, relacionados entre si e que envolvem atores e objetos, que em conjunto conduzem a um resultado que é de valor para pelo menos um cliente. A seguir, serão descritos cada um destes elementos participantes de um processo de negócio.

- Atividades – passos a serem seguidos para que um objetivo seja cumprido.
- Eventos – acontecem atômicamente, o que significa que eles não possuem duração. Podem gerar a execução de uma série de atividades.
- Pontos de decisão – pontos de tempo em que uma decisão é feita e afeta o caminho em que o processo é executado.
- Atores – pessoas, organizações ou sistemas computacionais que possuem um papel dentro do processo.
- Objetos – podem ser físicos (como equipamentos, materiais, produtos ou documentos em papel) ou imateriais (como documentos eletrônicos, por exemplo).
- Resultados – é o fruto resultante do processo. Pode ser positivo (quando agrega valor ao consumidor) ou negativo (quando não produz o resultado esperado).
- Consumidor – trata-se de um ator especial, o qual possui um papel especial, uma vez que o processo é voltado para produzir resultados que o satisfaçam.

Um processo de negócio possui um ou mais eventos, atividades e pontos de decisão, além de quantos atores e objetos forem necessários. Geram um ou mais resultados, que podem ser positivos ou negativos para um ou mais consumidores.

2.1.1.1 A Notação BPMN

A área de BPM possui demanda de ferramentas que auxiliem na documentação de processos, de maneira clara e sem nenhum tipo de ambiguidade. Para isso, existem diversas notações que permitem representar de forma padrão, através de uma gramática de símbolos, os processos de negócio de uma organização. Por estar sob responsabilidade da *Object Management Group* (OMG) - a principal organização internacional que aprova padrões abertos para aplicações orientadas a objetos - a *Business Process Model Notation*

(BPMN) é, atualmente, a mais destacada notação utilizada na área. Dados apontam que em 2013 a BPMN já se consolidava como a notação mais adotada para modelagem de processos nas organizações brasileiras (BPM GLOBAL TRENDS et al. 2013).

De acordo com (OMG, 1997), a BPMN auxilia na compreensão dos processos de negócio, através de uma representação gráfica que permite que as organizações entendam a si mesmas e a seus participantes no negócio, facilitando possíveis ajustes a serem feitos em seus processos. A BPMN está intimamente ligada com a fase de design de processos, uma vez que é capaz de modelar os processos na forma com que eles ocorrem dentro das organizações. Existem diversas ferramentas que trabalham com esta notação, tais como Bizagi Process Modeler, Aris e Oracle BPM.

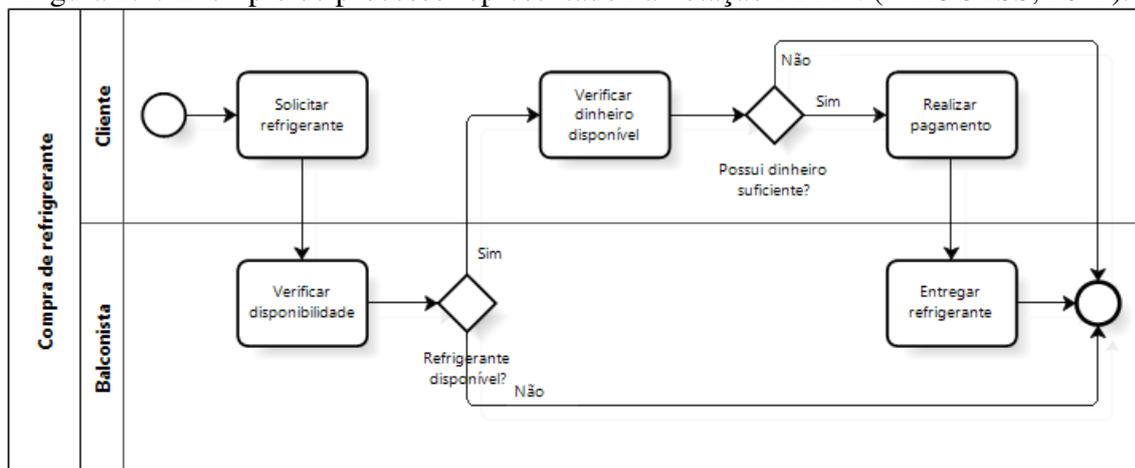
A BPMN possui diversas construções que, conectadas, formam seus diagramas. A seguir, estão exemplificadas as mais importantes, descritas conforme consta em (OMG, 2011):

- Piscina (*Pool*) – Representa cada um dos participantes de um processo, tendo todas as suas atividades agrupadas dentro dela.
- Raia (*Lane*) – Trata-se de uma subdivisão da piscina, utilizada para organizar e categorizar as atividades.
- Atividade – Uma atividade descreve um trabalho que é realizado dentro de um processo de negócio. Existem dois tipos de atividades: tarefas (atômicas) e sub-processos (não-atômicas, incluindo uma série de trabalhos que, em conjunto, formam a atividade).
- Evento – É considerado algo que “acontece” durante o processo, afetando o seu fluxo e geralmente possuindo uma causa ou um impacto. Se dividem em três tipos diferentes - eventos de início, eventos intermediários e eventos de término.
- Gateway – Indica uma tomada de decisão, que fará o fluxo seguir conforme o resultado da atividade anterior sobre o gateway. Representa, dentro da BPMN, especialmente os operadores lógicos XOR, AND e OR.
- Fluxo de sequência – Indica a ordem com que as atividades devem ser executadas dentro do modelo de processo.
- Fluxo de mensagem – Sinaliza a troca de mensagens entre dois participantes. É importante ressaltar que é utilizado apenas para a comunicação entre *pools* distintas.

Maiores detalhes a respeito dos elementos da BPMN podem ser encontrados na documentação oficial da notação, em sua versão mais atual (OMG, 2011). A Figura 2.1

mostra um exemplo de processo representado na notação BPMN. Trata-se de um simples processo de compra de refrigerante, que aponta todas as atividades e eventos ocorridos na interação entre o cliente e o balconista.

Figura 2.1: Exemplo de processo representado na notação BPMN (IPROCESS, 2012).



2.1.1.2 Armazenamento de Modelos de Processo

Atualmente, não há padrão para o armazenamento de modelos de processos em banco de dados. A maioria dos trabalhos tem como objetivo principal realizar estudos acerca da modelagem de processos, sem se preocupar com a forma com que eles são armazenados. É pequena a quantidade de trabalhos que ao menos menciona essa questão. Aqueles que o fazem, o fazem de forma superficial, tendo por objetivo apenas informar as tecnologias utilizadas, sem explicitar o método. Outro ponto a se destacar é que o banco de dados relacional é sempre o mais utilizado nestes casos.

Aprofundar a discussão a respeito do armazenamento de processos é a motivação do presente trabalho. Em particular, sugerir a utilização de modelos não-relacionais, especialmente baseados em grafos. Por lidarem de forma mais natural com as relações existentes nos modelos de processo, a ideia é validar a hipótese de que este modelo de banco de dados pode trazer benefícios nesta área.

2.1.2 NoSQL

NoSQL – acrônimo de “Not Only SQL” é uma classe de bancos de dados que rompe com as tradicionais regras do modelo relacional. É projetada para promover uma significativa escalabilidade horizontal, sendo capaz de tratar uma grande quantidade de

dados em operações distribuídas em diferentes servidores. Em um cenário atual, onde os conceitos de *Big Data* e *Cloud Computing* são cada vez mais requisitados, tais qualidades do modelo NoSQL tornam-se bastante importantes.

Para promover maior escalabilidade e performance, porém, os bancos de dados NoSQL geralmente não fornecem as garantias conhecidas como ACID – atomicidade, consistência, isolamento e durabilidade – presentes no modelo relacional. Ao invés disso, possuem características conhecidas como BASE – acrônimo de *Basically available, Soft state* e *Eventual consistency*, definido por Eric Brewer, criador do Teorema CAP (BREWER, 2000).

Existem quatro arquiteturas para tratamento de dados NoSQL: chave-valor, colunar, documento e grafo (HECHT; JABLONSKI, 2011). Elas são descritas da seguinte maneira:

- Chave-valor – Trata-se da mais simples entre todas, onde um valor é associado a uma chave. Por possuir uma estrutura tão simplificada, a velocidade de consulta tende a ser maior do que nas bases relacionais, além de suportar maior quantidade de dados e concorrência. As bases de dados mais populares que se utilizam desta arquitetura em seu funcionamento são a Voldemort Project¹, Redis² e Amazon DynamoDB³.
- Orientado a colunas – Utilizam tabelas como modelo de dado, mas não permitem associações entre elas. Ao contrário do modelo relacional, em que os registros são salvos de maneira contígua, o modelo orientado a colunas os salva em colunas contínuas no disco. Foram desenvolvidas para solucionar problemas com grande número de colunas e mudanças frequentes no esquema, trabalhando com dados não-normalizados para obter vantagens em dados distribuídos. Os principais bancos de dados orientados a colunas são Apache HBase⁴, Cassandra⁵, Hypertable⁶ e MonetDB⁷.
- Documentos – Nesta arquitetura, os pares de chave-valor são encapsulados como documentos em formato JSON ou XML. São, portanto, bastante semelhantes à arquitetura chave-valor em termos de estrutura, mas não de semântica. Considera

¹<http://www.project-voldemort.com/voldemort>

²<http://redis.io>

³<http://aws.amazon.com/pt/dynamodb>

⁴<http://hbase.apache.org>

⁵ <http://cassandra.apache.org>

⁶ <http://hypertable.com>

⁷<https://www.monetdb.org>

o documento como um todo, ao invés de uma série de pares chave-valor. Outra diferença entre eles é que bases de dados orientadas a documentos suportam índices em valores secundários. Como exemplos de bancos de dados baseados nesta arquitetura, podem ser citados o MongoDB⁸ e o CouchDB⁹

- Grafos – Em contraste com bancos de dados relacionais, os bancos de dados orientados a grafos são especializados em gerenciar dados fortemente associados entre eles, que possuam diversos relacionamentos. Este cenário é desfavorável aos bancos relacionais, que precisam realizar custosas operações como *joins* recursivos, o que compromete a eficiência das consultas. Os bancos orientados a grafos, por outro lado, tratam as informações como um conjunto de vértices e arestas, em um conceito baseado na Teoria dos Grafos. Cada vértice representa uma entidade, enquanto as arestas representam o relacionamento entre dois vértices. Seus conceitos são muito utilizados em redes sociais, uma vez que conseguem gerenciar dados relativos às pessoas e seus níveis de amizade de maneira natural. Os principais bancos de dados orientados a grafos são o Neo4j¹⁰, o AllegroGraph¹¹ e o GraphDB¹².

Para o presente trabalho, a arquitetura NoSQL escolhida é a orientada a grafos, dentre as apresentadas. Esta decisão se dá pelo fato de que modelos de processo podem facilmente serem projetados para grafos, uma vez que os diferentes elementos que os compõem se relacionam através de objetos de conexão. Traçando um paralelo, é como a relação entre os vértices de um grafo se dá através das arestas. Desta maneira, espera-se que o banco de dados orientado a grafos possa gerenciar as informações dos modelos de processo de maneira mais natural, evitando as custosas operações realizadas pelo modelo relacional.

2.1.3 Neo4j

O Neo4j é um banco de dados NoSQL, orientado a grafos, de código aberto. É desenvolvido pela empresa Neo Technology, tendo sua primeira versão lançada em 2007, na Suécia. De acordo com o DB-Engines ranking, trata-se do banco de dados orientado a

⁸<https://www.mongodb.org>

⁹<http://couchdb.apache.org>

¹⁰<https://neo4j.com>

¹¹<https://franz.com/agraph/allegrograph>

¹²<http://graphdb.ontotext.com>

grafos mais popular do mundo ¹³. Existem diferentes bancos de dados orientados a grafos, porém o Neo4j foi escolhido para o presente trabalho justamente por essa popularidade, além de possuir uma versão grátis, de código aberto, disponibilizar ferramentas interessantes para executar consultas e medir o desempenho de cada uma delas. Qualquer outro banco de dados orientado a grafos poderia ser selecionado para este trabalho, uma vez que o objetivo é avaliar o modelo de grafo e não o SGBD. Entretanto, por possuir todos os recursos necessários, o Neo4j foi a opção realizada.

É distribuído em três diferentes versões, que trazem como requisito a compra ou não de uma licença, dependendo da necessidade de utilização do usuário. A versão mais simples é chamada Community, sendo funcional, distribuída de forma gratuita através da licença GPLv3, mas trazendo algumas limitações de performance e escalabilidade. A versão chamada Enterprise libera tais limitações, oferecendo ganhos em termos principalmente de *backup* e *clustering*. Exige, porém, a aquisição de uma licença, a menos que a aplicação que a utiliza seja construída com código aberto. Por fim, a versão chamada Government estende a versão Enterprise, adicionando serviços específicos governamentais, tais como certificações, por exemplo.

As informações são armazenadas no banco de dados em forma de nodos, arestas ou atributos, sendo que cada nodo ou aresta pode ter uma quantidade ilimitada de atributos. Os atributos se constituem em pares chave-valor. Os nodos se relacionam através de arestas, tal como acontece na Teoria dos Grafos. Podem ser nomeados, auxiliando posteriormente nas buscas. As arestas são um tipo de relacionamento, não havendo limitação com relação ao número de arestas existentes entre dois nodos.

O Neo4j possui capacidade de lidar com grande quantidade de dados. A título de curiosidade, tal capacidade foi utilizada para analisar as informações contidas nos documentos do caso conhecido como “Paradise Papers”, que revelaram uma série de registros eletrônicos confidenciais de natureza fiscal relativas a diversas empresas e autoridades no mundo. Este caso ocorreu em novembro de 2017 e foram analisados, com o auxílio do Neo4j, cerca de 11,5 milhões de documentos, envolvendo 120000 pessoas e empresas. Tal serviço foi solicitado pelo Consórcio Internacional de Jornalistas Investigativos (ICIJ) e, após comprovado o sucesso da utilização da ferramenta, ela continua sendo aproveitada por grandes grupos de mídia como BBC, The Guardian e The New York Times. Isso mostra a capacidade de processamento de informações e a grande utilidade do Neo4j em áreas diversas, tais como o jornalismo. (INSIDER, 2017)

¹³<http://db-engines.com/en/ranking/relational+dbms>

O Neo4j é desenvolvido em Java e acessível por outros softwares através de uma Java API ou RESTful HTTP API. A linguagem de consultas de dados contidos no banco é a Cypher. Trata-se de uma linguagem declarativa, voltada especialmente para grafos, permitindo que eles sejam descritos de forma compreensível e mais simples possível (NEO4J, 2007). Foi inicialmente desenvolvida pela própria Neo Technology, porém em outubro de 2015 tornou-se de código aberto através do projeto OpenCypher e hoje é utilizada em diversos projetos, como SAP HANA, Redis e AgensGraph, além do próprio Neo4j (NEO4J, 2015). Seu padrão de representação é inspirado nos grafos tradicionais, com círculos e setas. Os vértices são representados por parênteses, enquanto as arestas são indicadas por colchetes associados a símbolos de maior e menor, que “apontam” a direção do arco em questão. Como exemplo, a expressão $(v1)-[a1:JOIN]->(v2)$ indica a relação entre os vértices nomeados como “v1” e “v2”, ligados através de uma aresta nomeada “a1”, com tipo de relação “JOIN” e origem em “v1” e destino em “v2”.

2.2 Trabalhos Relacionados

A presente Seção tem como objetivo fornecer um panorama a respeito dos principais questionamentos levantados pelo estudo, bem como os trabalhos relacionados e o método de busca utilizado para coletá-los. Apresenta, ainda, os trabalhos considerados de maior destaque e relevância, dentre àqueles que possuem relação com o assunto abordado.

A Subseção 2.2.1 aborda os principais questionamentos que fundamentam a pesquisa, apontando os principais interesses com relação ao tema e a motivação de cada um deles. A Subseção 2.2.2 aponta quais os caminhos percorridos para que fossem encontrados trabalhos relevantes sobre o tema, que representassem, de fato, o estado da arte no armazenamento de processos. Por fim, a Subseção 2.2.3 promove um estudo mais aprofundado a respeito dos trabalhos encontrados, destacando os que trazem contribuições mais significativas e descrevendo maiores detalhes a respeito deles. A Subseção 2.2.4 finaliza a Seção com uma análise comparativa dos trabalhos relacionados.

2.2.1 Principais Questionamentos

O estudo do Gerenciamento de Modelos de Negócio, em geral, possui foco especialmente na modelagem dos processos em si, buscando técnicas para extrair informações

relevantes do usuário e representá-las de maneira eficiente ao ponto de refletir os procedimentos adotados em uma organização. Muitos destes processos, porém, contém fragmentos extremamente semelhantes entre si, que podem ser reutilizados na composição de novos processos. Pensando nisso, é possível imaginar a criação de um repositório, onde estes subprocessos podem ser acessados por diferentes setores ou até mesmo diferentes organizações, sendo possível compartilhar tais fragmentos. Ao passo que o tamanho dos processos cresce e a quantidade de dados se torna relevante, é preciso pensar não apenas na modelagem do processo, mas também em seu armazenamento e em operações eficientes, capazes de encontrar semelhanças entre os subprocessos. Diante disso, foram realizados os seguintes questionamentos:

- Como é o estado da arte no que se refere ao armazenamento de processos?
- Como é feita a detecção de similaridade entre processos para sua possível reutilização?
- Quais são as tecnologias utilizadas para armazenar um processo? Existe algum estudo em que sejam comparadas duas ou mais tecnologias a fim de analisar a mais eficiente?
- Existem estudos que relacionem BPM com computação em nuvem, a fim de prover maneiras seguras e expansíveis para o compartilhamento de processos?

Com estes questionamentos em mente, foi realizada a pesquisa descrita a seguir.

2.2.2 Método de Pesquisa

Com o objetivo de encontrar trabalhos relevantes da área, capazes de refletir de forma confiável os estudos existentes, buscou-se adotar métodos de pesquisa bem definidos. Foram estudados artigos que abordam a revisão sistemática, tais como (KITCHENHAM et al., 2009) e (MAGDALENO; WERNER; ARAUJO, 2012). Embora não tenha sido realizada uma revisão sistemática exaustiva, foram adotadas diversas ideias presentes em tais artigos.

A busca por artigos se deu em conceituados repositórios eletrônicos, tais como Scopus¹⁴, IEEE Xplore¹⁵, ACM¹⁶ e Web of Science¹⁷. A biblioteca virtual que trouxe

¹⁴<https://www.scopus.com>

¹⁵<https://ieeexplore.ieee.org>

¹⁶<http://dl.acm.org>

¹⁷<https://webofknowledge.com>

resultados mais relevantes foi a Scopus, até mesmo porque diversos artigos contidos nela encontravam-se replicados nas demais bases de dados.

Como exemplos de palavras-chave utilizadas, estão “database”, “relational database”, “sql”, “graph”, “repository”, “cloud”, “clone”, “reuse”, “storage”, “similarity” e “process collection”. Todas estas palavras foram utilizadas em conjunto com termos relacionados a modelos de processos, como “bpm”, “business process model”, “business process management”. O termo “repository”, foi o que mais trouxe resultados relevantes, seguido por “cloud” e “clone”. Os demais termos não apresentaram resultados tão interessantes.

Ao realizar a busca, a primeira análise foi realizada no título dos artigos. Caso este mostrasse algo relevante ao assunto, o conteúdo do *abstract* e das palavras-chave também foi analisado. Os principais trabalhos foram separados para que, posteriormente, tivessem seu conteúdo estudado de forma mais aprofundada. Buscou-se estudos capazes de auxiliar os questionamentos apontados na Seção 2.2.1, sendo alguns deles destacados a seguir.

2.2.3 Trabalhos em Destaque

Ao término da pesquisa, foram encontrados cerca de trinta trabalhos importantes, os quais foram separados em categorias, a fim de agrupar aqueles que possuem foco semelhante e facilitar a identificação de possíveis contribuições para a corrente pesquisa. Estas categorias, bem como a justificativa de suas respectivas importâncias, são apresentadas na tabela 2.1.

Dentre os artigos encontrados, destaca-se o de (ROSA et al., 2011) que apresenta o APROMORE, um repositório de modelos de processos criado com o objetivo de disponibilizar ferramentas capazes de lidar com grandes coleções, suportando a complexidade existente nas constantes modificações a que os processos sofrem. Esta capacidade em lidar com grandes coleções de modelos de processos, inclusive, é abordada de maneira mais profunda no artigo (FAUVET et al., 2010). Trata-se de um relevante trabalho para a presente pesquisa, pois aborda as dificuldades em lidar com grandes quantidades de dados em processos, respeitando a capacidade de armazenar diferentes notações de modelagem de processos, incluindo BPMN, BPEL e YAWL. Porém, trata-se de um artigo que utiliza o modelo relacional no armazenamento dos processos, sem avaliar possíveis ganhos caso eles fossem salvos em bancos de dados com diferentes abordagens, como os orientados a

Tabela 2.1: Categorias dos trabalhos relacionados.

Categoria	Contribuição para o problema
BPM x Cloud	Buscou-se identificar os métodos de armazenamento utilizados, especialmente no que diz respeito às medidas de segurança adotadas, fator importante ao manipular dados em nuvem.
BPM <i>repositories</i>	Buscou-se descobrir quais as principais operações necessárias para a manipulação de modelos em um repositório. Além disso, o estudo desta categoria teve como objetivo analisar quais técnicas de armazenamento são mais comuns, bem como se o foco dos trabalhos se encontra no armazenamento ou na manipulação dos modelos de processo.
<i>Survey</i>	Foram procurados trabalhos desta categoria, pois são capazes de fornecer um panorama a respeito do estado da arte, mesmo que não se aprofundem tanto no assunto. Bastante úteis para dar um ponto de partida à pesquisa, indicando quais os aspectos principais sobre o assunto.
Similaridade de processos	Embora aparentemente não tenha relação direta com armazenamento, trata-se de um assunto de extrema relevância, pois é fundamental definir métricas de similaridade de processos para que seu reuso seja eficiente. Armazenar e reconstruir processos com base nestas métricas facilita a reutilização dos dados contidos no repositório.
Bancos Relacionais X Bancos Não-Relacionais	Buscou-se descobrir trabalhos capazes de comparar os modelos relacionais e não-relacionais de bancos de dados, analisando as vantagens de cada abordagem no contexto de modelagem de processos.

grafos, por exemplo.

Outro artigo bastante relevante é o de (YAN; DIJKMAN; GREFEN, 2012), principalmente por fornecer um panorama a respeito dos repositórios existentes atualmente, incluindo um resumo sobre as técnicas e tecnologias utilizadas por eles. No principal trecho do trabalho, uma tabela relaciona, dentre outras informações, quais os métodos usados para o armazenamento de processos, onde se conclui que a grande maioria deles trabalha com banco de dados relacionais, em especial o MySQL. Neste ponto, surgiu mais um questionamento: qual o principal motivo para que não sejam utilizados banco de dados orientados a grafos, uma vez que aparentam ser mais coerentes em sua relação com modelos de processos?

Poucos artigos mostram ser capazes de trazer uma resposta satisfatória sobre a maior utilização de bancos relacionais, se comparado com os não relacionais, no armazenamento de modelos de processos. Entre eles, está o (JIN et al., 2013), que é um estudo

sobre como encontrar fragmentos de processos como subgrafos. Porém, a exemplo dos demais, possui sua tecnologia de experimentos baseado em MySQL, aliado à linguagem de programação Java e processos sendo salvos em campos do tipo texto na base relacional, o que contraria a intuição de que a utilização de um banco baseado em grafos seria mais interessante neste caso. Em linha semelhante, o artigo (BECKER; LAUE, 2012) traz técnicas para mapear modelos de processos para grafos. Porém, seu foco principal é medir semelhança entre processos, ao invés de armazenamento dos mesmos. Sendo assim, não se aprofunda na questão de comparar o impacto de diferentes abordagens de bancos de dados no armazenamento de processos.

O artigo (LI et al., 2013) traz um estudo a respeito de técnicas para versionamento eficiente de processos. Apresenta cinco propostas para salvar apenas informações relevantes das modificações sofridas pelos processos entre suas versões. Apesar de ser focado em armazenamento de processos, é um trabalho que abrange prioritariamente a identificação de quais as informações devem ser salvas. Não cita uma abordagem de banco de dados específica a ser utilizada, seja ela relacional ou não.

Por fim, existe uma considerável quantidade de artigos que comparam bancos de dados relacionais e não-relacionais, o que mostra uma preocupação grande com o assunto atualmente. Entretanto, a grande maioria deles abordam o assunto de maneira genérica, sem apontar para uma área específica, muito menos relacionada à modelagem de processos. É o caso do artigo (NAYAK; PORIYA; POOJARY, 2017), por exemplo, que traz um panorama muito interessante a respeito dos bancos de dados NoSQL, mostrando seus principais tipos, bem como suas vantagens e desvantagens com relação aos modelos relacionais. Outros trabalhos são igualmente consistentes na apresentação dos modelos não-relacionais, tais como (KHAN; AHMED; SHAHZAD, 2017), (KUNDA; PHIRI, 2017) e (RAUT, 2017). Porém, nota-se uma tendência neles em citar grandes companhias que utilizam tal modelo de bancos de dados para o gerenciamento de redes sociais e grandes volumes de dados, tais como Facebook, Google, Yahoo e Amazon. Com isso, passam a impressão de analisar as qualidades dos bancos de dados NoSQL de acordo com as peculiaridades deste tipo de aplicação, deixando em segundo plano as diversas áreas que este modelo pode beneficiar. É o caso por exemplo de BPM que, por possuir característica de elevada quantidade de conectividade entre os elementos, tende a se favorecer do suporte fornecido pelos bancos de dados orientados a grafos.

2.2.4 Considerações Finais

Este capítulo explicitou as principais características dos principais trabalhos relacionados. Através do estudo de seus propósitos e da análise de suas individualidades, se possibilitou a construção da Tabela 2.2, que resume o comparativo entre eles. Tal estudo demonstra que, de fato, a preocupação maior é com modelagem de processos, deixando o armazenamento em segundo plano. Mesmo dentre as pesquisas que abordam armazenamento, a grande maioria utiliza bancos de dados relacionais, ignorando os possíveis benefícios gerados pela utilização de um banco de dados baseado em grafos. Com base nesta motivação, foram propostos dois modelos de armazenamento, um para cada tipo de banco de dados, a fim de comparar a eficiência de cada um deles nas operações mais elementares sobre modelos de processos, bem como avaliar eventuais ganhos em termos de praticidade na realização de consultas e identificação visual dos elementos, sempre dentro do contexto de modelagem de processos e levando em consideração suas peculiaridades.

Tabela 2.2: Comparativo entre os principais trabalhos relacionados.

	Foco em armazenamento?	Usa banco de dados relacional?	Usa banco de dados baseado em grafos?	Propõe abordagens diferentes do modelo relacional para manipular os processos?	Busca solucionar o problema de lidar com grandes coleções de processos?
(ROSA et al., 2011)	Sim	Sim	Não	Não	Sim
(LI et al., 2013)	Sim, porém atua mais no versionamento de processos	Não especificado	Não especificado	Não	Sim
(JIN et al., 2013)	Não	Sim	Não	Sim, trazendo um estudo sobre como encontrar fragmentos de processos como subgrafos	Sim
(BECKER; LAUE, 2012)	Não, tem foco em medir a semelhança entre processos	Não	Não	Sim, trazendo técnicas para mapear modelos de processos para grafos.	Não
(NAYAK; PORIYA; POOJARY, 2017)	Não, tem foco apresentar de maneira genérica os modelos NoSQL	Sim	Sim	Não	Não

3 MAPEANDO MODELOS DE PROCESSOS PARA MODELOS LÓGICOS DE BANCOS DE DADOS RELACIONAL E DE GRAFOS

Neste Capítulo são descritas as duas abordagens propostas para o mapeamento lógico de modelos de processos para bancos de dados com foco no armazenamento, sendo a primeira delas em banco de dados relacional e a segunda em banco de dados baseado em grafos. As Seções 3.2 e 3.3 apresentam todos os conceitos utilizados para o armazenamento de processo em banco relacional e grafos respectivamente, especificando o mapeamento de elementos, o modelo de dados, as restrições de integridade e o algoritmo de mapeamento. Essas duas seções são finalizadas com um exemplo prático do mapeamento realizado. A escolha dos modelos relacionais e baseados em grafo, em detrimento de outros, se deu para que fosse possível a comparação entre o modelo mais popular nas aplicações atuais (relacional), conforme apresentado na pesquisa descrita na Seção 2.2 e o modelo baseado em grafos, que permite um relacionamento mais direto entre os elementos muito próximo ao que ocorre nos modelos de processos. Além disso, os modelos orientados a grafos têm sido usados extensivamente em aplicações atuais, como, por exemplo, redes sociais, por permitir consultas mais diretas seguindo simplesmente os caminhos no grafo.

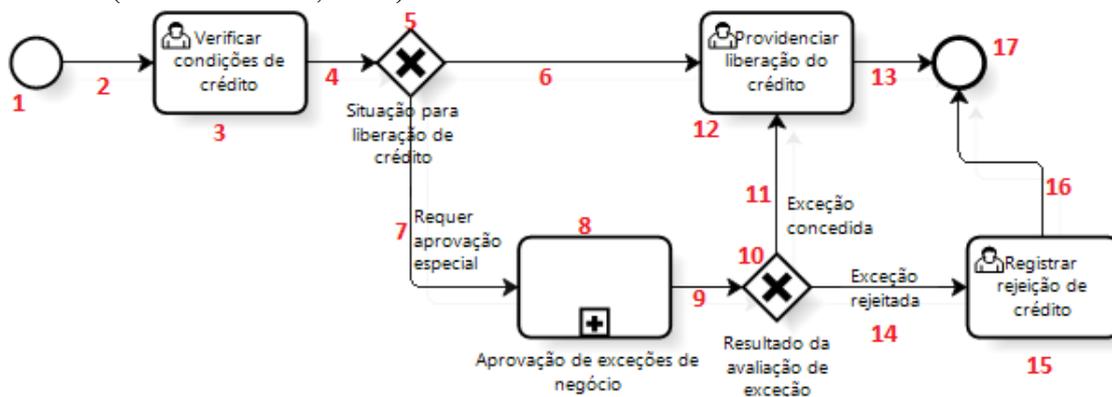
3.1 Modelo de Processo: Exemplo e Considerações

Durante a especificação das abordagens propostas, será utilizado o processo exibido na Figura 3.1. Este processo foi obtido na página iProcess (SGANDERLA, 2012), tendo sido escolhido por conter uma boa variedade de elementos, apesar de seu pequeno tamanho. Além disso, trata-se de um modelo de processo público, que não possui informações sigilosas.

O domínio do problema versa a respeito do processo de avaliação de crédito, uma operação bastante familiar e simples de compreender, facilitando a exemplificação dos mapeamentos. Nele, em suma, ocorre a verificação das condições de crédito, a análise se há ou não a necessidade de aprovação especial e, por fim, a liberação ou rejeição do crédito solicitado.

Os dois modelos propostos para mapeamento lógico de processos de negócio buscam contemplar o armazenamento dos tipos de elementos que compõe o BPMN – objetos

Figura 3.1: Modelo de processo utilizado na exemplificação de mapeamentos dos elementos. (SGANDERLA, 2012)



de fluxo, objetos de conexão, dados, partições e artefatos. Embora não seja apresentado o mapeamento de todos os elementos existentes nesta notação, foram escolhidos os principais elementos, os quais são os mais utilizados e que representam o grupo mínimo de elemento para modelagem de processos que permite modelar qualquer processo de negócio. Tais elementos selecionados são “Tarefa”, “Sub Processo”, “XOR”, “AND”, “Raia”, “Evento de início”, “Evento de fim” e “Fluxos de seqüência”. Os demais elementos possuem comportamento semelhante a estes e, portanto, possuem mapeamento análogo.

O armazenamento de modelos de processos possui algumas peculiaridades que foram levadas em consideração. A primeira delas é o número de elementos existentes em cada modelo armazenado, que, em processos bem modelados, tende a ser pequeno. Recomenda-se decompor os processos com mais de 50 elementos (MENDLING; REIJERS; AALST, 2010). Apesar dos modelos propostos conseguirem lidar com processos mais extensos, o foco principal é o armazenamento de fragmentos de tamanho moderado.

3.2 Armazenamento Relacional

Projetar o armazenamento de processos em um banco relacional exige a construção de um modelo em que constem as principais entidades capazes de representar os elementos existentes nos processos, bem como organizar a maneira com que elas estão associadas. As subseções a seguir são responsáveis por descrever as regras criadas para mapear cada uma destas categorias de elementos para o modelo relacional, bem como apresentar um modelo de entidade-relacional referente a elas. Além disso, são abordadas as restrições de integridade necessárias para o modelo proposto, bem como o algoritmo capaz de mapear os elementos de um modelo de processo para um modelo relacional e

um exemplo deste mapeamento sendo realizado.

3.2.1 Mapeamento de Elementos

O mapeamento de processos tem como objetivo agrupar elementos de modelagem de processos em entidades e relacionamentos que possam representá-los. Para isso, foi realizado um estudo que permite analisar o comportamento de cada uma das categorias dos elementos gráficos da modelagem de processos, identificando pontos em comum entre eles. A seguir, estão descritos os detalhes de cada uma destas categorias.

- **Objetos de fluxo** - os objetos de fluxo se dividem entre eventos, atividades e desvios. Apesar de possuírem diferentes funções dentro da modelagem de processos, todos eles podem ser considerados com semelhante comportamento, em termos de modelagem. Isto significa que os objetos de fluxo são representados por uma entidade que, em geral, está associada a outras e que representa uma ação a ser tomada. Sendo assim, para efeitos de mapeamento destes elementos da modelagem de processos para entidades de um modelo para armazenamento relacional, todos os objetos de fluxo foram agrupados em uma mesma categoria e, portanto, são representados por uma única entidade.
- **Dados** - a representação de dados na modelagem de processos é bastante semelhante aos objetos de fluxo, pois também consiste em entidades que podem ser relacionadas a outras semelhantes. Desta maneira, o mapeamento para a criação de um modelo de armazenamento relacional faz com que os dados sejam unidos aos objetos de fluxo em uma mesma entidade, sendo diferenciados através dos respectivos atributos.
- **Objetos de conexão** - os objetos de conexão, por sua vez, possuem comportamento diferenciado em comparação aos objetos de fluxo e dados. Esta categoria inclui fluxo de sequência, fluxo de mensagem e associações, sendo responsável por criar um relacionamento entre dois objetos de fluxo ou dados. Indicam, por exemplo, a ordem em que os objetos de fluxo ou dados são definidos, bem como a relação entre eles. Por possuírem esta função de representar a forma de associação entre dois elementos, ao invés de retratar ações que podem ser interligadas, os objetos de conexão devem ser mapeados em entidade própria.
- **Partições** - as partições, compostas por raias e piscinas, têm como comportamento

fazer com que os demais elementos pertençam a elas. Podem ser vistas, por exemplo, como elementos-pai dos demais, uma vez que sua função é agrupá-los. Em termos de mapeamento para um modelo relacional, seu comportamento não se assemelha aos controles de conexão, pois não são capazes de relacionar dois elementos, indicando ordem de execução, por exemplo. Por outro lado, podem ser unidos à mesma entidade dos controles de fluxo e dados, uma vez que sua existência é independente de outros elementos. Como particularidade, porém, é preciso tomar cautela para que não sejam associados a outros elementos através da entidade que representa os controles de conexão. Além disso, a relação entre as partições e seus elementos-filho deve ser expressa através de um atributo específico dentro da própria entidade.

- Artefatos - os artefatos são elementos adicionais para sinalização do processo, mas que não influenciam no seu fluxo. No que se refere aos grupos, possuem comportamento extremamente semelhante às partições, uma vez que agrupam uma série de elementos, que em geral são objetos de fluxo, conexão ou dados. Além disso, não são responsáveis por associar os demais elementos em termos de ordem de execução, se afastando, em comportamento em um modelo relacional, dos objetos de conexão. Desta forma, o mapeamento segue o que foi definido para as partições, ou seja, os artefatos pertencem a mesma entidade, porém sem serem relacionados através da entidade dos objetos de conexão e possuindo o relacionamento com seus elementos-filho definidos através de atributo específico dentro da própria entidade.

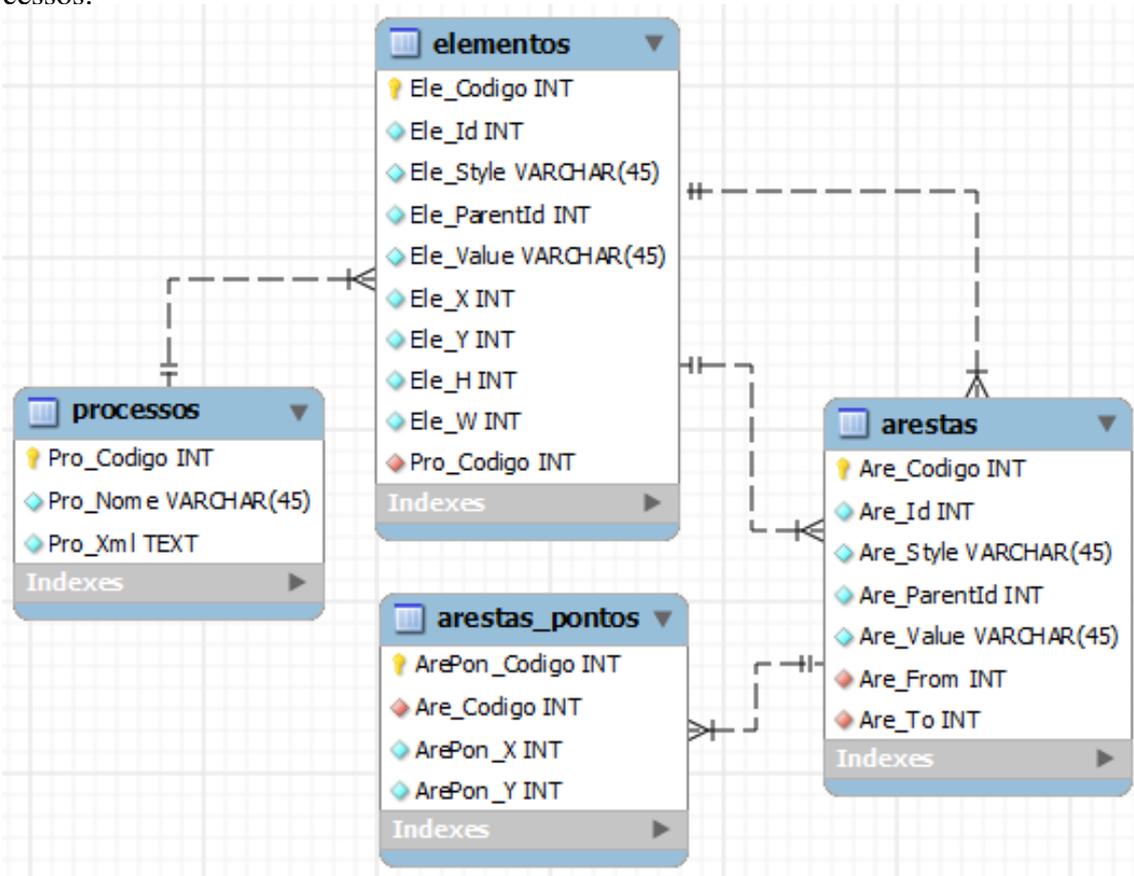
3.2.2 Modelo de Dados

Baseado neste estudo da subseção anterior, foi projetado um modelo entidade relacionamento, a fim de representar o armazenamento da totalidade de informações capazes de representar um processo. Este modelo está representado na Figura 3.2.

É importante salientar que todas as tabelas possuem uma chave primária, com um valor sequencial que identifica o registro e facilita as operações de consulta, em fase posterior à de armazenamento. A tabela denominada “processos” não representa nenhuma entidade estudada, tendo sido criada para armazenar o código identificador o nome de determinado processo a ser armazenado.

Ligada à tabela “processos”, encontra-se a tabela “elementos”, que reproduz a entidade que agrupa os objetos de fluxo, dados, partições e artefatos. Seus atributos re-

Figura 3.2: Representação do modelo relacional projetado para o armazenamento de processos.



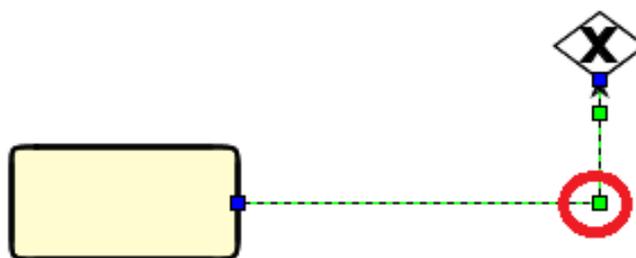
presentam o código identificador do elemento dentro do processo; o tipo ao qual pertence (que pode ser tarefa, sub processo, piscina, entre outros); seu elemento pai – utilizado principalmente em elementos pertencentes a uma partição ou grupo, sendo estes considerados elemento pai; seu valor atribuído pelo usuário, se for o caso; seus pontos X e Y, que definem as posições horizontal e vertical do diagrama em que se encontram e, por fim, suas medidas de largura e altura, para que a representação do elemento seja fiel ao construído pelo usuário inclusive em suas dimensões. O atributo “Pro_Codigo” é uma chave estrangeira responsável por associar o elemento armazenado ao seu respectivo processo, existente na tabela “processos”.

A tabela “arestas” reproduz a entidade a qual pertence os objetos de conexão, que criam a associação existente entre dois elementos da modelagem de processos, conhecidas como transições. A Figura 3.3 exemplifica esta relação. Seus atributos representam o código identificador da aresta dentro do processo; seu estilo (que pode ser associação, fluxo de sequência, fluxo de mensagem, entre outros); seu elemento pai; valor atribuído pelo usuário, se for o caso. Existem também duas chaves estrangeiras, que representam os elementos de origem e destino da transição, criando, ainda, uma relação entre as tabelas

“elementos” e “arestas”.

Por fim, a tabela “arestas_pontos” é responsável por armazenar os pontos em que os objetos de fluxo foram plotados no modelo de processo, dentro da ferramenta de modelagem. Não são informações importantes para a estrutura do modelo, mas permitem recuperar fielmente a sua construção dentro da ferramenta de modelagem. É possível ver na Figura 3.3 um exemplo de um ponto, circulado em vermelho, sendo as coordenadas de seus eixos sendo salvas nos campos “ArePon_X” e “ArePon_Y”. Além destes registros, a tabela ainda contém o identificador único de cada ponto (“ArePon_Codigo”) e uma chave estrangeira que relaciona o ponto à sua respectiva aresta contida na tabela “arestas” (“Are_Codigo”).

Figura 3.3: Exemplo de aresta associando uma tarefa a um operador XOR, em verde. Circulado em vermelho, exemplo de um ponto pertencente a uma aresta.



3.2.3 Restrições de Integridade

O modelo proposto foi projetado de maneira a garantir todas as restrições de integridade necessárias para o caso em questão. Apesar de ser um modelo relacional pequeno, dele fazem parte 4 das 5 restrições de integridade principais, conforme listadas a seguir:

- Integridade de entidade: todas as tabelas possuem chaves primárias, que permitem identificar de maneira única seus elementos. São compostas por valores inteiros, que se auto incrementam a cada registro salvo. Como exemplo, os atributos “Pro_Codigo”, “Ele_Codigo”, “Are_Codigo” e “ArePon_Codigo”, que são as respectivas chaves primárias das tabelas “processos”, “elementos”, “arestas” e “arestas_pontos”.
- Integridade referencial: todas as chaves estrangeiras existentes no modelo proposto fazem referência à chave primária da tabela de origem. Assim, garantem que se tratam de informações com o mesmo tipo e tamanho de dados em ambas as tabe-

las. Como exemplo, o atributo “Pro_Codigo”, existente na tabela “elementos” e que referencia o atributo de mesmo nome, na tabela “processos”. Ambos são inteiros de tamanho 11, além de possuírem o mesmo nome como forma de facilitar a identificação da origem da chave primária original.

- **Integridade de domínio:** todos os atributos existentes na tabela possuem explícita declaração a respeito do tipo e tamanho de dados que devem conter. Como os processos possuem uma estrutura que não varia muito, não houve necessidade de permitir que algum dos campos propostos aceite valores nulos. Sendo assim, ficou determinado que todos os campos das tabelas propostas devem ser preenchidos. Como exemplo, é possível citar o campo “Pro_Nome”, da tabela “processos”, que se trata de um VARCHAR de tamanho 45, não nulo. Isto permite afirmar que o domínio deste campo aceita caracteres alfa numéricos e que existe obrigatoriedade em fornecer um nome para cada processo armazenado.

3.2.4 Algoritmo de Mapeamento de Modelo de Processo para Grafo Equivalente

Diante do modelo proposto, foi criado um algoritmo responsável por mapear todos os elementos existentes nos modelos de processo para a criação de um grafo equivalente a ele, em termos de estrutura e representação de seus atributos. Buscou-se a possibilidade de armazenar os dados de maneira a garantir boa eficiência de desempenho nas consultas, além de manter a totalidade ou, ao menos, o máximo possível de informações do processo armazenadas. O Algoritmo 1 exhibe a ideia proposta, que é explicada passo a passo a seguir:

- **Linha 2:** o algoritmo recebe uma lista de elementos existentes no modelo de processo, iterando sobre eles.
- **Linha 3:** caso o elemento seja um objeto de conexão, ele é mapeado para uma aresta e os passos contidos entre as linhas 4 e 16 são seguidos. Senão, o algoritmo pula para a linha 18.
- **Linhas 4 a 9:** uma aresta é criada e recebe os atributos do objeto de conexão existente no modelo de processo.
- **Linha 10:** esta aresta é inserida na tabela “arestas”.
- **Linhas 11 a 16:** os objetos de conexão possuem pontos intermediários, que indicam a forma com que devem ser plotados. Para manter a recuperação visualmente

fiel das arestas, esses pontos são armazenados e, posteriormente, redesenhados no mesmo lugar de origem. Entre as linhas 11 e 16, é criado cada um desses pontos, com suas respectivas coordenadas x e y e armazenado na tabela “arestas_pontos”.

- Linha 18: caso o elemento do modelo de processo não seja um objeto de conexão, ele é tratado como vértice durante o mapeamento.
- Linhas 19 a 28: um vértice é criado e recebe os atributos do elemento existente no modelo de processo.
- Linha 29: este vértice é inserido na tabela “elementos”.

O algoritmo proposto tem complexidade quadrática, o que aponta para a sua viabilidade computacional e, conseqüentemente, a possibilidade de representar modelos de processo em bancos de dados relacionais. Como os modelos de processos tendem a pos-

suir 50 elementos ou menos, isso tornaria ainda mais viável a execução do algoritmo.

Algoritmo 1: MAPEAMENTO EM MODELO RELACIONAL

Entrada: Lista de elementos do modelo de processo

```

1 início
2   para cada elemento faça
3     se elemento for um objeto de conexão então
4       aresta ← vazio
5       aresta[id] ← elemento[id]
6       aresta[tipo] ← elemento[tipo]
7       aresta[valor] ← elemento[valor]
8       aresta[id – origem] ← elemento[id – origem]
9       aresta[id – destino] ← elemento[id – destino]
10      inserir aresta na tabela “arestas”
11      para cada ponto do objeto de conexão faça
12        ponto ← vazio
13        ponto[x] ← vazio
14        ponto[y] ← vazio
15        inserir ponto na tabela “arestas_pontos”
16      fim
17    fim
18    senão
19      vertice ← vazio
20      vertice[id] ← elemento[id]
21      vertice[tipo] ← elemento[tipo]
22      vertice[id – pai] ← elemento[id – pai]
23      vertice[valor] ← elemento[valor]
24      vertice[id – processo] ← elemento[id – processo]
25      vertice[x] ← elemento[x]
26      vertice[y] ← elemento[y]
27      vertice[h] ← elemento[h]
28      vertice[w] ← elemento[w]
29      inserir vertice na tabela “elementos”
30    fim
31  fim
32 fim

```

3.2.5 Exemplo de armazenamento

Para exemplificar a prática do modelo proposto, é utilizado o processo descrito na Figura 3.1. Os elementos existentes no processo, bem como a forma com que são armazenados, estão organizados conforme listado a seguir:

- Elementos 1 e 17 – por serem eventos (objetos de fluxo), são armazenados dentro da tabela “elementos”, sendo importante que seus atributos “Ele_Style” deixem claros que se tratam de eventos e que “Ele_Id” contenham os valores 1 e 17, respectivamente, que os identificam dentro do processo.
- Elemento 2 – por ser uma associação (objeto de conexão), é armazenado na tabela “arestas”, sendo importante que seu atributo “Ele_Style” deixe claro que se trata de uma associação e que “Ele_Id” contenha o valor 2, que o identifica dentro do processo. Por fim, os atributos “Are_From” e “Are_To” devem conter as chaves estrangeiras dos elementos de saída e de entrada da conexão, respectivamente. Os elementos 4, 6, 7, 9, 11, 13, 14 e 16 seguem o mesmo raciocínio do elemento 2, pois possuem as mesmas funções dentro do processo e, portanto, o armazenamento é semelhante.
- Elemento 3 – por ser uma atividade (objetos de fluxo), é armazenado dentro da tabela “elementos”, sendo importante que seu atributo “Ele_Style” deixe claro que se trata de uma atividade e que “Ele_Id” contenha o valor que o identifica dentro do processo. Os elementos 12 e 15, por também serem atividades, são armazenados de maneira semelhante. O elemento 8, uma sub-tarefa, também se comporta da mesma maneira, mas é preciso identificar que se trata de uma sub-tarefa através do atributo “Ele_Style”.
- Elementos 5 e 10 – são desvios (objetos de fluxo) e, portanto, se comportam de maneira semelhante ao elemento 3. Porém, devem identificar no atributo “Ele_Style” que se tratam de desvios do tipo XOR.

As Figuras a seguir mostram o preenchimento dos registros em cada uma das tabelas, considerando o exemplo trabalhado. A Figura 3.4 exhibe o preenchimento da tabela “processos”, com o nome e o identificador único do processo, bem como o seu documento XML na íntegra. A Figura 3.5 mostra o armazenamento dos objetos de fluxo do processo, convertidos para a tabela “elementos”. A Figura 3.6 reflete os objetos de conexão dos processos, armazenados na tabela “arestas”. Por fim, a Figura 3.7 exhibe o

armazenamento dos pontos dos objetos de fluxo foram plotados no modelo e processo, a fim de poder recuperar fielmente sua construção na ferramenta de modelagem.

Figura 3.4: Preenchimento da tabela “processos”, com os dados de identificação do modelo de processo.

	Pro_Codigo	Pro_Nome	Pro_Xml
▶	1	Exemplo de armazenamento	<bpmnGraph> <BPMNElement> <style>Start-...
★	NULL	NULL	NULL

Figura 3.5: Armazenamento dos objetos de fluxo contidos no modelo de processo.

	Ele_Codigo	Ele_Id	Ele_Style	Ele_ParentId	Ele_Value	Ele_X	Ele_Y	Ele_H	Ele_W	Pro_Codigo
▶	1	1	Start-Event	1		50	80	30	30	1
	2	2	Start-Event	1		50	80	30	30	1
	3	3	Task	1		140	70	100	30	1
	4	4	Gateway-Exclusive-XOR-Data-Based	1		290	80	40	30	1
	5	5	Task	1		420	70	100	50	1
	6	6	End-Event	1		600	80	30	30	1
	7	7	Collapsed-Sub-Process	1		260	220	100	50	1
	8	8	Gateway-Exclusive-XOR-Data-Based	1		450	230	40	30	1
	9	9	Task	1		620	220	100	50	1
★	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Figura 3.6: Armazenamento dos objetos de conexão contidos no modelo de processo.

	Are_Codigo	Are_Id	Are_Style	Are_ParentId	Are_Value	Are_From	Are_To
▶	1	10	Edge	1		2	3
	2	11	Edge	1		3	4
	3	12	Edge	1		4	5
	4	13	Edge	1		5	6
	5	14	Edge	1		4	7
	6	15	Edge	1		7	8
	7	16	Edge	1		8	5
	8	17	Edge	1		8	9
	9	18	Edge	1		9	6
★	NULL	NULL	NULL	NULL	NULL	NULL	NULL

3.3 Armazenamento em Bancos de Dados Orientados a Grafos

O objetivo desta Seção é apresentar uma proposta de modelagem e armazenamento de processos em banco de dados baseado em grafos. As subseções a seguir explicitam todos os fatores levados em conta para a realização do armazenamento, desde a criação

Figura 3.7: Armazenamento dos pontos dos objetos de fluxo como foram plotados na ferramenta de modelagem.

	ArePon_Codigo	ArePon_X	ArePon_Y	Are_Codigo
▶	1	80	95	1
	2	110	95	1
	3	140	95	1
	4	240	95	2
	5	265	95	2
	6	290	95	2
	7	330	95	3
	8	375	95	3
	9	420	95	3
	10	520	95	4
	11	560	95	4
	12	600	95	4
	13	310	110	5
	14	310	165	5
	15	310	220	5
	16	360	245	6
	17	405	245	6
	18	450	245	6
	19	470	230	7
	20	470	175	7
	21	470	120	7
	22	490	245	8
	23	555	245	8
	24	620	245	8
	25	670	220	9
	26	670	165	9
	27	615	165	9
	28	615	110	9
*	NULL	NULL	NULL	NULL

de regras de mapeamento entre os dois modelos, o modelo de dados que represente esta proposta, o algoritmo proposto para mapeamento dos elementos e um exemplo de armazenamento de processo, conforme as regras propostas, em um banco de dados baseado em grafos.

3.3.1 Mapeamento de elementos

Para o mapeamento de elementos, tomou-se como prioridade a identificação de comportamentos inerentes aos elementos gráficos da modelagem de processos que sejam semelhantes a vértices ou a arestas. Este estudo, entretanto, trata-se exatamente do

mesmo já realizado para o modelo relacional, o qual identificou elementos gráficos responsáveis por representar ações, enquanto outros servem para criar uma associação entre estas ações. No contexto dos grafos, estes comportamentos podem ser transpostos para vértices e arestas, respectivamente.

Desta forma, os objetos de fluxo, dados, partições e artefatos, tratados como uma mesma entidade no modelo relacional, se mantêm sendo considerados de mesma categoria em termos de mapeamento. Neste contexto de grafos, assumem o papel de vértices, pois representam ações, acontecimentos ou informações que existem de maneira independente das demais. Por outro lado, os objetos de conexão são tratados como arestas de um grafo, pois se responsabilizam por criar uma associação entre os outros elementos, tendo sua própria existência condicionada aos demais. Importante ressaltar que tais arestas são unidirecionais e não possuem pesos, pois o objetivo é simplesmente indicar a origem e o destino da aresta, tal como acontece com os objetos de conexão nos modelos de processos.

3.3.2 Modelo de dados

Para que fique visualmente clara a diferenciação entre as várias categorias de elementos gráficos para modelagem de processos, foi definido que cada uma delas receberia uma cor diferente. Esta separação pode ser, ainda, realizada por subcategorias dentro de cada classe (eventos, atividades e desvios, dentro da categoria de objetos de fluxo, por exemplo). Sendo assim, o resultado é a formação de um grafo com diferentes cores, cada uma identificando um elemento existente na modelagem de processos. As arestas, por sua vez, devem receber em seu nome a identificação da subcategoria dos objetos de conexão ao qual representam (associação, fluxo de mensagem e fluxo de sequência). Os demais atributos de cada elemento da modelagem de processo são mapeados para os próprios atributos de cada vértice ou grafo, dependendo de sua categoria.

Desta forma, o modelo conceitual proposto mantém o foco no relacionamento entre os elementos do processo, permitindo que eles se tornem explícitos ao ponto de realizar consultas que identifiquem claramente o comportamento de cada processo. Isso facilita na constatação de processos semelhantes dentro de uma base de dados, de maneira mais eficiente possível.

A Figura 3.8 exemplifica a ideia proposta para mapear os elementos pertencentes a um modelo de processo para grafos. Neste exemplo, foi escolhida a cor verde para tingir todos os vértices que representam os eventos de início que aparecem no modelo

de processo. De maneira análoga, foram escolhidas as cores preto para atividades, cinza para *gateways* exclusivos, azul para subprocessos e vermelho para os eventos finais. As cores foram selecionadas de maneira aleatória, podendo ser modificadas. A restrição é de que, uma vez associadas a determinado tipo de elemento dentro do processo, representem apenas ele, sem nenhum tipo de repetição de cor para diferentes tipos de elementos no modelo de processo.

Figura 3.8: Exemplo de funcionamento do mapeamento de modelos de processos para grafos.

Elemento BPMN	Representação BPMN	Cor Mapeamento	Número da Cor	Representação Mapeamento
Evento inicial		Verde	1	
Atividade		Preto	2	
Gateway Exclusivo (XOR)		Cinza	3	
Subprocesso		Azul	4	
Evento Final		Vermelho	5	

3.3.3 Algoritmo de Mapeamento de Modelo de Processo para Banco Relacional

Diante do modelo proposto, foi criado um algoritmo responsável por mapear todos os elementos existentes nos modelos de processos para a inserção dos mesmos em suas respectivas tabelas no banco de dados relacional. Buscou-se a possibilidade de armazenar os dados de maneira a garantir boa eficiência de desempenho nas consultas, além de manter a totalidade ou, ao menos, o máximo possível de informações do processo armazenadas. O Algoritmo 2 exibe a ideia proposta, que é explicada passo a passo a seguir:

- Linha 2: o algoritmo recebe uma lista de elementos existentes no modelo de processo, iterando sobre eles.
- Linha 3: caso o elemento seja um objeto de conexão, ele é mapeado para uma aresta e os passos contidos entre as linhas 4 e 11 são seguidos. Senão, o algoritmo pula para a linha 13.
- Linhas 4 a 6: uma aresta é criada e recebe como origem o elemento análogo ao

originário de seu equivalente, dentro do modelo de processo. De forma semelhante, o destino da aresta é definido, de acordo com o elemento de destino de seu análogo, dentro do modelo de processo.

- Linhas 7 a 9: os objetos de conexão possuem pontos intermediários, que indicam a forma com que devem ser plotados. Para manter a recuperação visualmente fiel das arestas, esses pontos são armazenados e, posteriormente, redesenhados no mesmo lugar de origem. Entre as linhas 7 e 9, cada um desses pontos intermediários é plotado de acordo com sua coordenada X e Y.
- Linha 10: a aresta é plotada, ligando o vértice de origem ao de destino, passando pelos pontos intermediários plotados anteriormente.
- Linha 13: caso o elemento do modelo de processo não seja um objeto de conexão, ele é tratado como vértice durante o mapeamento.
- Linhas 14 a 16: verifica-se se o tipo do elemento (objeto de fluxo, dado, objeto de conexão, partição ou artefato) já ocorreu anteriormente no modelo de processo em questão. Se já ocorreu, pintar este vértice do grafo com a mesma cor utilizada no vértice que possui seu mesmo tipo.
- Linhas 17 a 20: para casos em que o tipo de elemento não ocorreu no modelo de processo em questão, é escolhida uma cor inédita e atribuída a este vértice e a todos os posteriores vértices de mesmo tipo.
- Linha 21: o vértice é plotado em seus pontos X e Y, pintado com a cor escolhida nos passos anteriores.

À exemplo do algoritmo proposto para o modelo relacional, o algoritmo para a representação em bancos de dados orientados a grafos também possui complexidade quadrática, demonstrando em teoria sua viabilidade. Isto significa que é possível representar modelos de processos como grafos e, conseqüentemente, tratar os bancos de dados orien-

tados a grafos como uma opção para o armazenamento de modelos de processos.

Algoritmo 2: MAPEAMENTO EM BANCO DE DADOS BASEADO EM GRAFOS

Entrada: Lista de elementos do modelo de processo

```

1 início
2   para cada elemento faça
3     se elemento for um objeto de conexão então
4       aresta ← vazio
5       aresta[origem] ← elemento[id – origem]
6       aresta[destino] ← elemento[id – destino]
7       para cada ponto do objeto de conexão faça
8         Plotar ponto intermediário da aresta em suas coordenadas X e
          Y
9       fim
10      Plotar aresta
11     fim
12     senão
13       vertice ← vazio
14       se tipo do elemento já tiver ocorrido no modelo de processo então
15         vertice[cor] ← vertice_mesmo_tipo[cor]
16       fim
17       senão
18         cor ← cor_inedita_no_grafo
19         vertice[cor] ← cor
20       fim
21       Plotar vértice em seus pontos X e Y
22     fim
23   fim
24 fim

```

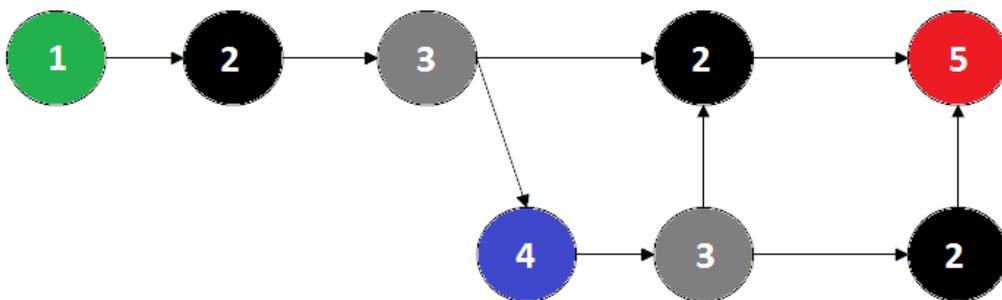
3.3.4 Exemplo de armazenamento

Para exemplificar a utilização do modelo conceitual proposto para bancos de dados orientados a grafos, foi utilizado o mesmo processo exibido na Figura 3.1. Nele, existem seis diferentes tipos de elementos gráficos para modelagem de processos, que são descri-

tos a seguir. Com o propósito de tornar ainda mais legível a identificação de cada tipo de categoria, foi adicionado um número que representa cada uma das cores no presente exemplo.

Com base nas diretivas definidas, a Figura 3.9 exibe o resultante do mapeamento do processo em questão para um grafo. As atividades (objetos de fluxo) são identificadas pela cor preta (número 2). Em geral, são associados a outros elementos e possuem suas características salvas nos atributos do vértice. Os operadores XOR (desvios, objetos de fluxo) são vértices identificados pela cor cinza (número 3), enquanto o subprocesso (objeto de fluxo) é um vértice identificado pela cor azul (número 4). Os eventos de início e fim (objetos de fluxo) são vértices identificados pelas cores verde (número 1) e vermelha (número 5), respectivamente. Por fim, as associações (objetos de conexão), diferentemente dos demais elementos que constam no presente exemplo, não se tratam de vértices, mas de arestas do grafo. Seu nome leva a identificação de que se trata de uma associação, enquanto seus atributos são responsáveis por armazenar as demais informações que eventualmente sejam necessárias.

Figura 3.9: Exemplo de grafo que representa um processo mapeado.



3.4 Considerações Finais

No presente Capítulo, foram descritos os dois modelos propostos o armazenamento de modelos de processos, um baseado em bancos de dados relacionais e outro focado em bancos de dados orientados a grafos. Conforme demonstrado, ambos possuem algoritmo de mapeamento com complexidade capaz de viabilizar computacionalmente suas execuções. Isso aponta para a possibilidade de representação dos modelos de processos em ambos os modelos de bancos de dados apresentados, ampliando a variedade de opções amplamente utilizadas nos trabalhos relacionados atuais – que, em geral, se utilizam apenas de modelos relacionais ou sequer se focam no armazenamento de processos.

O modelo relacional proposto foi pensado de maneira a permitir o gerenciamento de uma grande coleção de processos. Ao armazená-los de maneira padronizada em um banco de dados, é possível disponibilizar opções para que os usuários compartilhem seus processos de maneira simplificada, algo que não é absolutamente trivial no atual contexto de BPM. Isso permite que os processos sejam reutilizados entre diferentes companhias ou grupos de estudo, por exemplo, trazendo confiabilidade (uma vez que os processos já são consolidados) e otimização de tempo, uma vez que os processos não precisam ser criados do zero.

De maneira semelhante, o modelo não relacional proposto também foi desenvolvido de maneira a permitir o gerenciamento de grandes coleções de processos e o consequente compartilhamento deles entre os usuários. Adicionalmente, busca-se beneficiar das propriedades dos grafos na manipulação dos modelos de processos, algo que foi encontrado em apenas dois trabalhos relacionados: (JIN et al., 2013) e (BECKER; LAUE, 2012). Seu próprio armazenamento em banco de dados utiliza-se de bancos de dados orientados a grafos, técnica não encontrada durante o levantamento de trabalhos relacionados, que consta no Capítulo 2.2.

Ao apresentar estas alternativas de armazenamentos, abre-se possibilidade de abordar de maneira explícita um armazenamento em modelo relacional, sugerir um modelo não relacional que se diferencia dos utilizados nos trabalhos relacionados atuais e compará-los, analisando possíveis ganhos que trazem em relação um ao outro. Diversos trabalhos citados no Capítulo 2.2 traçam paralelos entre os bancos de dados relacionais e NoSQL, tais como (NAYAK; PORIYA; POOJARY, 2017), (KHAN; AHMED; SHAHZAD, 2017), (KUNDA; PHIRI, 2017) e (RAUT, 2017). Entretanto, nenhum deles coloca os conceitos destes bancos de dados no contexto do armazenamento de processos, levando em consideração suas peculiaridades e analisando o comportamento das operações a serem realizadas. Como limitação dos modelos apresentados, o fato de terem sido desenvolvidos de maneira não exaustiva, podendo eventualmente ser melhorados em termos de desempenho. Porém, como em modelagem de processos este nem sempre é o foco, preferiu-se fazer a análise baseada em outros critérios tão relevantes quanto, garantindo apenas a viabilidade mínima de execução em termos de desempenho computacional.

A Tabela 3.1 exibe um comparativo entre as propostas apresentadas e os conceitos encontrados nos trabalhos relacionados.

Tabela 3.1: Comparativo entre os principais trabalhos relacionados e os métodos propostos.

	Foco em armazenamento?	Usa banco de dados relacional?	Usa banco de dados baseado em grafos?	Propõe abordagens diferentes do modelo relacional para manipular os processos?	Busca solucionar o problema de lidar com grandes coleções de processos?
Proposto	Sim	Sim	Sim	Sim	Sim
(ROSA et al., 2011)	Sim	Sim	Não	Não	Sim
(LI et al., 2013)	Sim, porém atua mais no versionamento de processos	Não especificado	Não especificado	Não	Sim
(JIN et al., 2013)	Não	Sim	Não	Sim, trazendo um estudo sobre como encontrar fragmentos de processos como subgrafos	Sim
(BECKER; LAUE, 2012)	Não, tem foco em medir a semelhança entre processos	Não	Não	Sim, trazendo técnicas para mapear modelos de processos para grafos.	Não

4 AVALIAÇÃO EXPERIMENTAL

Neste Capítulo é descrita a avaliação experimental, a qual é dividida em duas diferentes abordagens. A primeira diz respeito a comparar os modelos de armazenamento propostos em termos de desempenho computacional. Para isso, é feito o mapeamento físico dos modelos de processo para bancos de dados relacionais e de grafos, sendo realizados experimentos que comparam a velocidade de processamento de ambos nas operações mais usuais nos modelos de processo. A segunda abordagem tem como objetivo detectar a familiaridade dos usuários com a utilização dos modelos de armazenamento propostos, visando descobrir se o modelo baseado em grafos traz ganhos, comparado ao relacional.

4.1 Experimentos de Mapeamento Físico dos Processos

O objetivo desta Seção é mapear um conjunto de modelos de processos para bancos de dados físicos e compará-los em termos de tempo de processamento nas operações mais utilizadas em armazenamento de BPM. Além do armazenamento de processos em si, outras operações que podem ser consideradas as principais são a reconstrução de processos e a busca por fragmento de processo. O armazenamento consiste em salvar todos os dados necessários para restaurar o modelo de processo original. A busca por fragmento permite encontrar ocorrências de um processo menor dentro de outros. A importância desta operação é identificar a existência de processos estruturalmente semelhantes dentro de um banco de dados, permitindo o reaproveitamento deles em diferentes contextos. A reconstrução de processos consiste em coletar todas as informações necessárias, dentre as previamente armazenadas no banco de dados, para obtenção do modelo de processo original.

É importante ressaltar que o tempo de processamento destas operações não necessariamente é o critério mais importante a ser levado em conta na decisão acerca de qual tipo de banco de dados deve ser utilizado. Muitas aplicações em BPM podem priorizar a correção da busca realizada, a facilidade com que o usuário identifica ou consulta os processos dentro da base de dados, por exemplo. Entretanto, medir o desempenho de processamento das operações é importante no sentido de verificar a viabilidade de execução de cada uma delas. Em suma, o presente experimento visa analisar a hipótese de viabilidade de execução das operações em um tempo razoável. Não busca aperfeiçoar de forma exhaustiva o tempo de processamento das operações, mas sim verificar se elas podem ser

executadas em tempo viável, sendo balanceadas com os demais aspectos importantes a serem levados em consideração na escolha de um tipo de banco de dados.

Cada modelo de processo avaliado é mapeado para os modelos lógicos propostos no Capítulo 3, tanto para o banco de dados relacional, quanto o banco de dados baseado em grafos. Logo após, é feito o mapeamento para o banco de dados físico e, por fim, são avaliadas as operações de armazenamento, busca de fragmentos e reconstrução de processos.

4.1.1 Cenário de Avaliação

Os experimentos foram executados em um Intel Core i5 3230M, com 8GB RAM e 1TB HD SATA III 7200 RPM de disco rígido. O sistema operacional utilizado é o Windows 10 Professional 64 bits.

Com relação à metodologia utilizada nos experimentos, criou-se um complemento ao software disponibilizado no trabalho (GASSEN et al., 2014), que permite a modelagem de processos. Tal software foi escolhido por cumprir a tarefa de modelagem de processamentos com os principais elementos do BPMN, além de possuir fácil acesso ao código-fonte, permitindo a realização das modificações necessárias para o acréscimo de funcionalidades necessárias. Tais funcionalidades adicionadas no presente experimento são a possibilidade de criação das consultas físicas nos bancos de dados, responsáveis por armazenar e buscar fragmentos de processos. O fluxo inclui a modelagem dos processos através desta ferramenta e a passagem de um arquivo XML contendo as informações necessárias para representar o processo a um webservice criado no presente trabalho. Este webservice tem como objetivo captar esta representação do processo no arquivo XML e tratá-la, gerando as consultas e executando-as nos respectivos bancos de dados. O armazenamento de processos foi realizado através da criação de um webservice para demonstrar a capacidade de utilização do modelo proposto em diferentes linguagens, ou seja, o armazenamento de processos pode ser realizado através de qualquer ferramenta de modelagem, bastando que este envie o arquivo XML conforme padrão estabelecido ao webservice.

Para evitar influências indevidas no tempo de processamento das operações, tais como aquelas geradas por frameworks utilizados no webservice, pelas aberturas de conexões aos bancos de dados ou outros fatores relacionados, o webservice foi construído de maneira a registrar em arquivos as consultas a serem realizadas. Tais consultas foram executadas posteriormente, de forma manual, diretamente no gerenciador de banco de dados

padrão dos modelos relacional e baseado em grafos. Além disso, é importante salientar que a obtenção de tempos apresentados é resultado da média de três execuções de cada operação, a fim de reduzir possíveis influências externas a própria consulta realizada.

4.1.2 Base de Dados

A escolha de modelos de processos como casos de teste foi feita de forma a diversificar os elementos de modelagem de processos disponíveis, a fim de avaliar possíveis impactos causados pela utilização dos mesmos. Um dos principais fatores levados em consideração foi a escolha de processos com diferentes tamanhos, com o intuito de avaliar a evolução da complexidade para o armazenamento de processos em cada tipo de banco de dados, à medida que os processos se tornam mais robustos. Entretanto, a realização dos experimentos teve como foco os modelos de processo que possuem no máximo cerca de 50 elementos, uma vez que as boas práticas de modelagem de processos recomendam decompor processos maiores do que este limite (MENDLING; REIJERS; AALST, 2010).

Foram utilizados modelos de processos da Sociedade Brasileira de Computação¹, a fim de avaliar o comportamento dos bancos de dados ao serem expostos a processos reais.

4.1.3 Mapeamento para o banco de dados físico MySQL

A presente subseção busca mapear o modelo lógico, proposto no Capítulo 3, para o modelo físico relacional. O objetivo é demonstrar a criação das tabelas, atributos e relacionamentos capazes de viabilizar a execução das operações dentro de um banco de dados relacional.

Para a realização de testes de desempenho em banco de dados relacional foi utilizado o MySQL 5.6.17, por ser um banco de dados de código aberto bastante popular², além de possuir desempenho e estabilidade satisfatórios para os testes realizados. Apesar de ser um banco de dados de código aberto, com base na GPL (*General Public License*), o MySQL sofre restrições caso o software que o utilize também não seja GPL, devendo ser adquirida uma licença comercial. O MySQL pode ser executado nas principais plataformas, como Linux, Windows, Solaris, FreeBSD e MacOS. Sua linguagem padrão para

¹<http://www.sbc.org.br>

²<http://db-engines.com/en/ranking/relational+dbms>

realização de consultas é o SQL.

Para a utilização do MySQL, foram desenvolvidas diversas interfaces gráficas que facilitam o planejamento e a gestão do banco de dados. Atualmente, o cliente oficial do MySQL é o MySQL Workbench³, distribuído de maneira gratuita pela própria Oracle, sua proprietária. Entre suas principais funcionalidades, está a oportunidade de criar modelos relacionais, que auxiliam no projeto e visualização do banco de dados. É possível desenhar as entidades, seus atributos e relacionamentos, permitindo que se tenha uma visão mais ampla dos elementos que compõe o banco de dados em que se está trabalhando. Após definir completamente a estrutura do banco de dados, o software possui recursos para a criação automática dos comandos necessários para a construção do banco de dados, através da exportação na linguagem SQL. A principal vantagem da utilização deste método é focar o trabalho na solução do problema, abstraindo questões relativas à sintaxe, que são tratadas pelo próprio MySQL Workbench. Desta maneira, para criar o modelo físico para o banco de dados MySQL, utilizou-se a própria ferramenta disponibilizada pelo MySQL Workbench, que gerou o código apresentado e comentado no Apêndice A. Percebe-se um modelo enxuto, que representa bem o modelo entidade relacionamento proposto, composto por apenas quatro tabelas.

4.1.4 Mapeamento para o banco de dados físico Neo4j

A presente subseção busca mapear o modelo lógico, proposto no Capítulo 3, para o modelo físico não-relacional. O objetivo é esclarecer a criação da estrutura capaz de viabilizar a execução das operações dentro de um banco de dados baseado em grafos.

A implementação dos testes realizados em banco de dados baseado em grafos foi realizada no Neo4j 2.2.3. Este banco de dados foi escolhido por possuir uma versão grátis, de código aberto, disponibilizar ferramentas interessantes para executar consultas e medir o desempenho de cada uma delas, além de ser um banco de dados bastante popular no mundo⁴. Sua linguagem padrão é o Cypher, capaz de expressar de maneira simples e eficiente os atributos e relacionamentos existentes em um grafo.

O Neo4j é um banco de dados baseado no conceito NoSQL e, ao contrário do MySQL, não utiliza os conceitos estruturados. Ao invés de armazenar as informações em tabelas, as estruturas são representadas de maneira mais dinâmica, como nos conceitos

³<https://www.mysql.com/products/workbench>

⁴<http://db-engines.com/en/ranking/graph+dbms>

matemáticos da Teoria dos Grafos. Por este motivo, antes de armazenar um grafo, não é preciso criar nenhuma estrutura prévia, uma vez que cada grafo possui seus próprios atributos. A modelagem física, em suma, é criada no momento do armazenamento do grafo.

4.1.5 Avaliação do Armazenamento de Processos

O objetivo deste experimento é avaliar o tempo de execução para armazenar modelos de processos. A proposta é analisar se os métodos descritos são capazes de armazenar todas as informações de maneira rápida e eficiente, comparando suas performances.

O armazenamento de processos em banco de dados relacional se inicia com a inserção dos dados de identificação do processo, que constam na tabela “processos”. Tomando como exemplo o subprocesso mostrado na Figura 4.1, teríamos a seguinte instrução sendo executada:

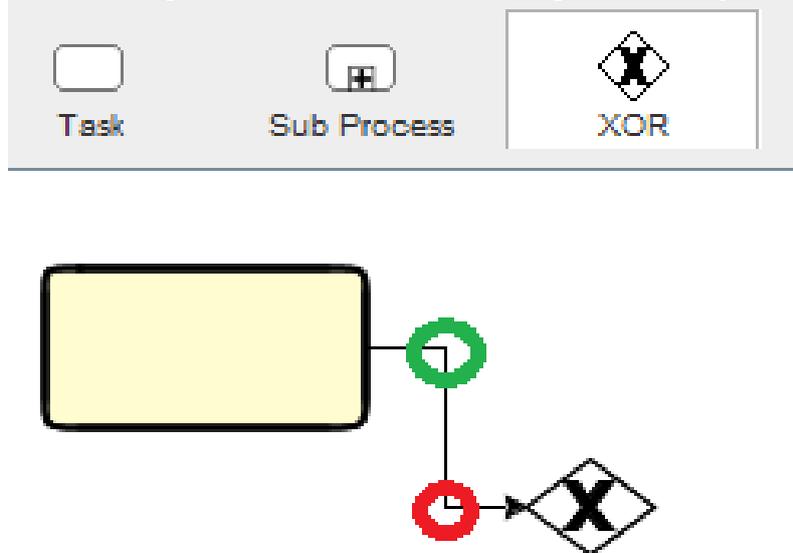
```
INSERT INTO processos (Pro_Nome, Pro_Xml)
VALUES ('Armazenamento Teste, '{"bpmnGraph":{"BPMNElement":
  [{"style":"Task","id":"2","parentID":"1","value":"null",
  "x":"30.0","y":"30.0","w":"100.0","h":"50.0"},{"style":
  "XOR","id":"4","parentID":"1","value":"null","x":"200.0",
  "y":"40.0","w":"40.0","h":"30.0"},{"style":"Edge","id":"5",
  "parentID":"1","value":"","from":"2","fromStyle":"Task",
  "to":"4","toStyle":"XOR","mxPoint":{"point":[{"x":"130.0",
  "y":"55.0"},{"x":"165.0","y":"55.0"},{"x":"200.0",
  "y":"55.0"}]}]}]}')
);
```

Para cada elemento existente, também é executada uma inserção, armazenando-o na tabela “elementos”. Utilizando o mesmo mostrado na Figura 4.1, para armazenar os elementos Task teríamos a seguinte instrução sendo executada:

```
INSERT INTO elementos (Ele_Id, Ele_Style, Ele_ParentId,
  Ele_Value, Ele_X, Ele_Y, Ele_H, Ele_W, Pro_Codigo)
VALUES (2, 'Task', 1, NULL, 30, 30, 50, 100, 1);
```

Os campos inseridos se referem, respectivamente, ao identificador do elemento dentro do modelo de processo, ao tipo de elemento, ao identificador do elemento pai, ao

Figura 4.1: Modelo de processo utilizado como exemplo na avaliação experimental.



seu valor (se for o caso), às coordenadas X e Y, altura e largura em que está desenhado no modelo de processo (para permitir a reconstrução gráfica do modelo de processo) e o identificador do processo ao qual faz parte.

O elemento de conexão representado pela seta na Figura 4.1, por sua vez, é armazenado na tabela “arestas”, através da seguinte instrução:

```
INSERT INTO arestas (Are_Id, Are_Style, Are_ParentId,
    Are_Value, Are_From, Are_To)
VALUES (5, 'Edge', 1, NULL, 2, 3);
```

Os campos inseridos se referem, respectivamente, ao identificador do objeto de conexão dentro do modelo de processo, ao tipo de objeto de conexão, ao identificador de seu elemento pai, ao seu valor (se for o caso), ao identificador do elemento de origem e ao identificador do elemento de destino.

Em casos onde a representação gráfica do objeto de conexão tem mudança de sentido, como o que ocorre nos pontos circulados em vermelho e verde na Figura 4.1, é preciso armazenar os dados relativos a estes pontos na tabela “arestas_pontos”. O objetivo disto é simplesmente conseguir reconstruir o modelo de processo com os elementos desenhados nas mesmas posições posteriormente. Para isso, para cada ponto, é executada uma instrução como a exemplificada a seguir:

```
INSERT INTO arestas_pontos (Are_Codigo, ArePon_X, ArePon_Y)
VALUES (5, 130, 55);
```

Os campos inseridos se referem, respectivamente, ao identificador da aresta a que o ponto pertence e às suas coordenadas X e Y onde ele se encontra.

O armazenamento de processos no banco de dados baseado em grafos é semelhante, sendo necessário executar uma instrução para cada vértice e aresta a ser criada. Tomando como exemplo o elemento “Task”, da Figura 4.1, seu armazenamento se dá com a execução da seguinte instrução:

```
CREATE (Task2:Task {graphName:'GrafoTeste', graphId:'1',
style:'Task', id:'2', parentID:'1', value:'null', x:'30.0',
y:'30.0', w:'100.0', h:'50.0'})
```

As informações armazenadas são as mesmas do banco relacional, ficando associadas ao vértice inserido. O armazenamento do elemento XOR da Figura 4.1 se dá de maneira análoga. A aresta a ser criada, por sua vez, é armazenada no banco de dados através da seguinte instrução:

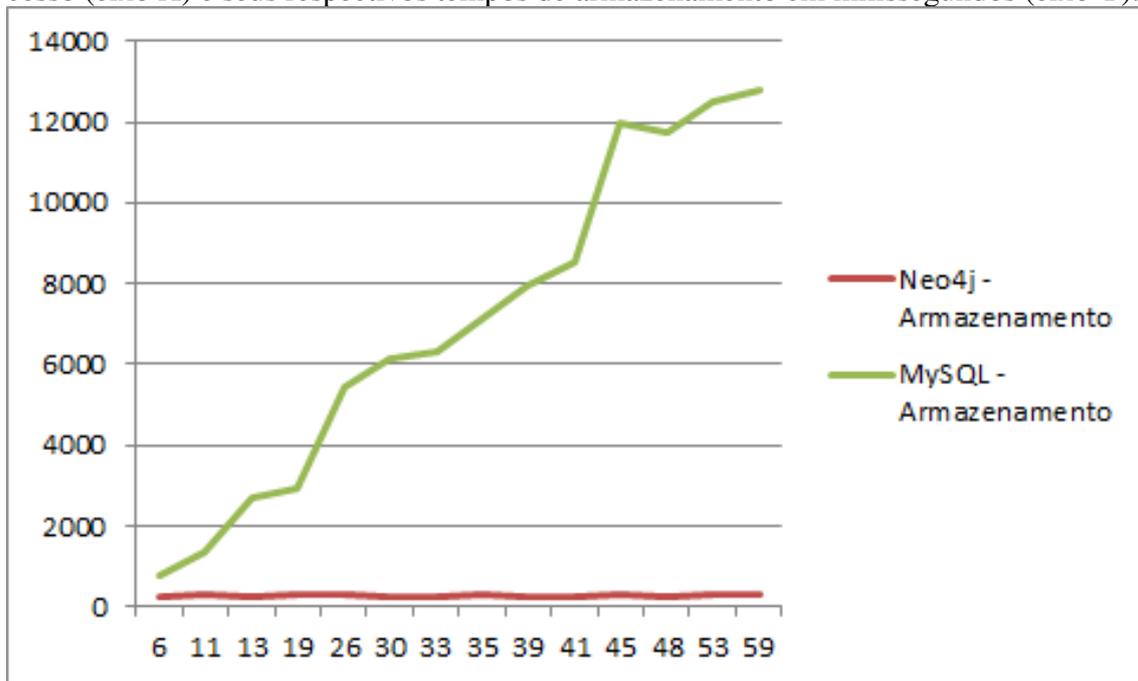
```
CREATE (Task2)-[:JOIN {graphName: 'GrafoTeste', graphId:'1',
style:'Edge', id:'5',parentID:'1', value:'', mxPointX1:'130.0',
mxPointY1:'55.0', mxPointX1:'155.0', mxPointY1:'55.0',
mxPointX1:'155.0', mxPointY1:'105.0', mxPointX1:'180.0',
mxPointY1:'105.0'}]->(XOR3);
```

A instrução cria uma aresta que faz a ligação entre os elementos identificador como “Task2” e “XOR3”. Além disso, todas as informações a respeito da aresta ficam associadas junto ao seu próprio armazenamento, inclusive os dados referentes às coordenadas de onde estão representadas.

Conforme mencionado, a realização de experimentos de armazenamento consistiu na inserção de uma série de processos nos bancos de dados relacional e não-relacional. Cada processo foi inserido 3 vezes, obtendo-se a média de tempo das operações como resultado final. A Figura 4.2 traça o comportamento identificado durante os testes, relacionando o número de elementos do modelo de processo (eixo X), com seus respectivos tempos de armazenamento em milissegundos (eixo Y).

Para os processos armazenados em banco de dados baseado em grafos, nota-se um valor constante no tempo de armazenamento dos modelos de processo, independente do número de elementos que existam neles, dentre os valores testados. Um modelo de processo com 6 elementos é armazenado em 268 ms, enquanto um de 48 elementos é armazenando em 264 ms, mostrando que o número de elementos não é determinante para

Figura 4.2: Gráfico representando relação entre número de elementos do modelo de processo (eixo X) e seus respectivos tempos de armazenamento em milissegundos (eixo Y).



o crescimento exponencial do tempo de processamento do armazenamento. A criação de elementos e arestas se dá em tempo bastante curto, que gira em torno de 220 a 315 ms. Tal resultado demonstra a viabilidade de armazenamento de processos, em um banco de dados baseado em grafos.

Para processos armazenados em bancos de dados relacionais, entretanto, nota-se um crescimento exponencial preocupante, de acordo com o número de elementos existentes no modelo de processo. Na base de dados testada, o menor tempo encontrado foi de 753 para o modelo de processo com 6 elementos, sendo o maior tempo de 12816 ms para o modelo de processo com 59 elementos – justamente o menor e o maior modelo de processo, respectivamente. A curva de tempo de processamento acompanha o crescimento do número de elementos, além de ser significativamente maior do que os tempos encontrados no modelo não-relacional. Credita-se a isso o relacionamento menos natural para o tratamento de grafos em banco relacional, além da necessidade de criação de mais consultas de inserção para armazenar dados relativos às arestas, em especial na tabela “arestas_pontos”. Entretanto, como não foram realizadas tentativas exaustivas para otimizar o tempo de processamento, acredita-se que ele possa ser significativamente melhorado e competir com o modelo não-relacional. Este, porém, não é o foco do presente trabalho, que pretende apenas verificar a viabilidade de armazenar modelos de processo em tempos razoáveis. De acordo com os parâmetros testados, considera-se que, apesar

dos tempos de processamento estarem aquém do desejado, este objetivo foi cumprido.

4.1.6 Avaliação da Busca por Fragmento de Processos

O objetivo deste experimento é avaliar o tempo de execução para buscar fragmentos de modelos de processos. A proposta é analisar se os métodos descritos são capazes de buscar subprocessos existentes dentro de processos maiores de maneira rápida e eficiente, comparando suas performances. A busca por fragmento de processo é importante na reutilização de processos durante a construção de novos modelos de processo.

A busca por fragmento de processo no MySQL tem por objetivo executar uma consulta que retorne o identificador dos modelos de processo que contenham o fragmento procurado. Tomando como exemplo o fragmento de processo mostrado na Figura 4.1, teríamos a seguinte instrução sendo executada:

```
SELECT DISTINCT(e1.Pro_Codigo) AS Pro_Codigo
FROM elementos e1, elementos e2, arestas a1
WHERE e1.Ele_Style = 'Task'
AND e2.Ele_Style = 'XOR'
AND e1.Ele_Codigo = a1.Are_From
AND e2.Ele_Codigo = a1.Are_To
```

A consulta realizada retorna o código identificador dos processos que contenham um elemento do tipo “Task” e outro do tipo “XOR” associados por um objeto de conexão.

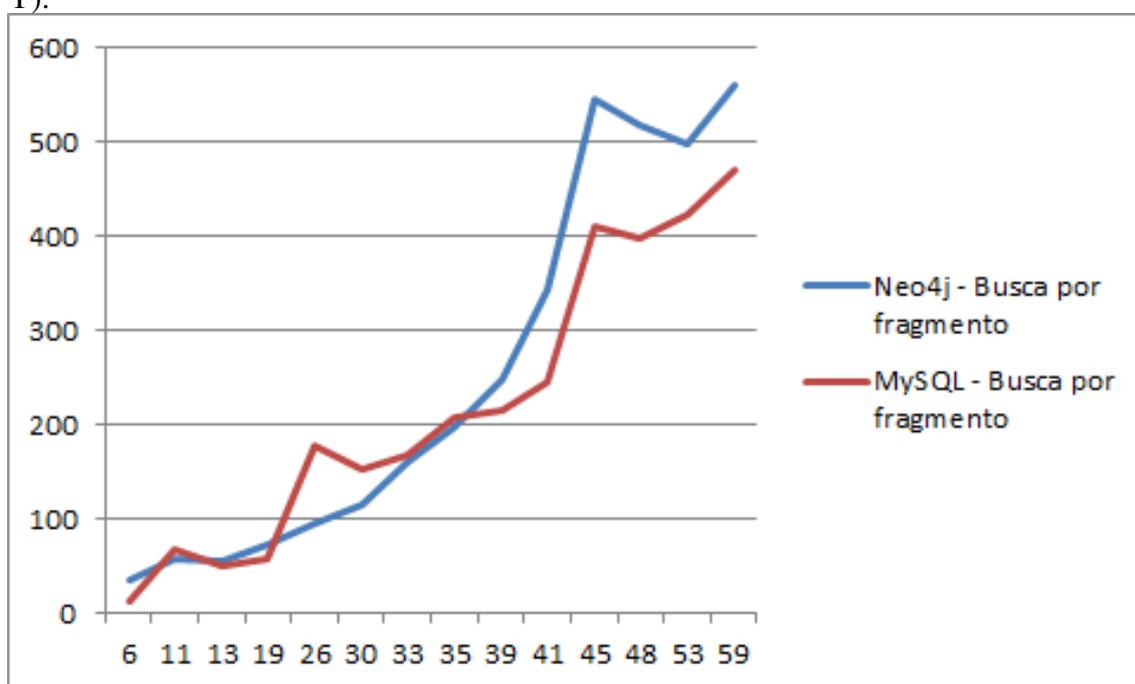
A busca pelo mesmo fragmento de processo no Neo4j, por sua vez, se dá através da execução da seguinte consulta:

```
MATCH (a:Task)-[:JOIN]->(b:XOR)
WITH DISTINCT a.graphId as graphId
RETURN graphId
```

A instrução “MATCH” é responsável por buscar um elemento do tipo “Task”, identificado pela letra “a”, que esteja associado por uma aresta ao elemento do tipo “XOR” identificado pela letra “b”. A instrução “WITH DISTINCT” é responsável por unificar eventuais identificadores de processo duplicados no retorno da consulta. A instrução “RETURN”, por sua vez, retorna os identificadores dos modelos de processo que contenham o fragmento buscado.

De maneira análoga aos experimentos de armazenamento de processos, a realização de experimentos de busca por fragmentos consistiu na busca de uma série de fragmentos de processos nos bancos de dados relacional e não-relacional. Cada fragmento foi buscado 3 vezes, obtendo-se a média de tempo das operações como resultado final. A Figura 4.3 traça o comportamento identificado durante os testes, relacionando o número de elementos do modelo de processo (eixo X), com seus respectivos tempos de busca por fragmentos de processos em milissegundos (eixo Y).

Figura 4.3: Gráfico representando relação entre número de elementos do fragmento de processo buscado (eixo X) e seus respectivos tempos de busca em milissegundos (eixo Y).



Ao contrário do fenômeno ocorrido no armazenamento de processos, nota-se um crescimento do tempo de processamento da busca de fragmento de processo no Neo4j, de acordo com o número de elementos existentes no fragmento. Em processos maiores, a partir de cerca de 40 elementos, este tempo é, inclusive, maior do que o próprio tempo de armazenamento equivalente. Todos os valores, entretanto, se mantiveram entre 36 e 560 ms, considerados viáveis para a execução das operações.

Na busca realizada no MySQL, apesar de os tempos também crescerem de acordo com a quantidade de elementos existentes no modelo de processo, notou-se valores bastante aceitáveis e completamente compatíveis com os encontrados no experimento com o Neo4j. O MySQL se mostrou mais rápido nos extremos, tendo sido mais eficiente para o menor e os maiores processos testados. Os tempos se mantiveram entre 14 e 471 ms. Desta maneira, é demonstrada não apenas a viabilidade da busca por fragmento no banco

relacional, mas também sua competitividade com o banco baseado em grafos, uma vez que ambos não apresentaram diferença significativa de tempo na execução das operações.

4.1.7 Avaliação da Reconstrução de Processos

O objetivo deste experimento é avaliar o tempo de execução para buscar todas as informações necessárias para reconstruir os modelos de processos de maneira rápida e eficiente, comparando suas performances.

A operação de reconstrução do processo no MySQL tem por objetivo retornar todos os elementos e arestas existentes em um modelo de processo. Tomando como exemplo o modelo de processo mostrado na Figura 4.1, para retornar os elementos existentes no modelo de processo, teríamos a seguinte consulta sendo executada:

```
SELECT * FROM elementos WHERE Pro_Codigo = {identificador}
```

A consulta realizada tem como parâmetro o código identificador do modelo de processo buscado. Retorna o conjunto de elementos que pertencem a ele. A busca de objetos de conexão se dá através da seguinte consulta:

```
SELECT * FROM arestas
WHERE Are_From IN (
    SELECT Ele_Codigo
    FROM elementos
    WHERE Pro_Codigo = {identificador})
OR Are_To IN (
    SELECT Ele_Codigo
    FROM elementos
    WHERE Pro_Codigo = {identificador})
```

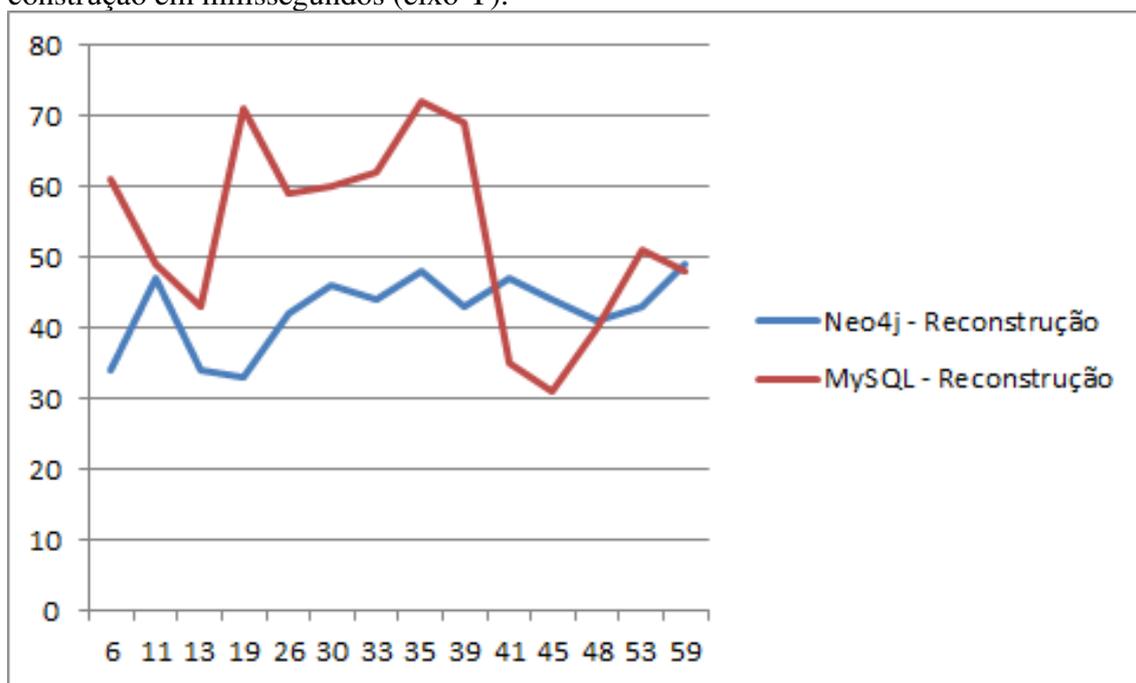
Analogamente à busca de elementos, a busca de arestas tem como parâmetro o identificador do modelo de processo buscado, retornando o conjunto de objetos de conexão existentes nele. A busca pelo mesmo fragmento de processo no Neo4j, por sua vez, se dá através da execução da seguinte consulta:

```
MATCH (n)
WHERE n.graphId = '{identificador}'
RETURN n
```

A consulta tem como parâmetro o identificador do modelo de processo buscado e retorna o grafo correspondente ao armazenamento do mesmo.

De maneira análoga aos experimentos anteriores, a realização de experimentos de reconstrução de processos consistiu na execução da operação em uma série de processos nos bancos de dados relacional e não-relacional. Cada modelo de processo foi reconstruído 3 vezes, obtendo-se a média de tempo das operações como resultado final. A Figura 4.4 traça o comportamento identificado durante os testes, relacionando o número de elementos do modelo de processo (eixo X), com seus respectivos tempos de busca das informações para reconstrução de processos em milissegundos (eixo Y).

Figura 4.4: Gráfico representando relação entre número de elementos do fragmento de processo buscado (eixo X) e seus respectivos tempos de busca das informações para reconstrução em milissegundos (eixo Y).



Por serem consultas extremamente simples e diretas, os tempos obtidos se mostraram pequenos e constantes, tanto no MySQL, quanto no Neo4j. Enquanto o Neo4j registrou valores entre 33 e 49 ms, o MySQL registrou tempos entre 31 e 72 ms. Além disso, não foi constatada relação entre o número de elementos e arestas dos processos com o tempo de processamento das operações. A diferença de desempenho de ambos os bancos de dados não foi significativa, sendo que todos os tempos apresentados se mostraram viáveis para a realização das operações com modelos de processos.

4.1.8 Análise Geral dos Resultados

A grande vantagem do modelo relacional com relação ao baseado em grafos é a popularidade, uma vez que se tratam de bancos de dados cujo conhecimento a respeito deles é bastante difundido, enquanto os bancos de dados orientados a grafos são menos utilizados. Entretanto, analisando os modelos propostos para armazenamento, dentro do contexto de modelagem de processos, nota-se um desempenho melhor do Neo4j sobre o MySQL. Os experimentos apresentados demonstram que os modelos relacionais podem gerar gargalos na medida em que os processos cresçam, uma vez que o relacionamento entre as entidades na representação de processos é muito menos natural do que o relacionamento existente no banco de dados baseado em grafos. Entretanto, é preciso salientar como limitação do trabalho a já citada falta de otimização exaustiva dos resultados. Como o objetivo é demonstrar a viabilidade de utilização dos modelos em um tempo razoável, não se buscou melhorar o desempenho do banco relacional a ponto de amenizar tais gargalos e obter tempos de processamento ideais.

Nas demais operações, relativas à busca por fragmentos e reconstrução de processos, tanto Neo4j, quanto MySQL se mostraram eficientes na execução das operações propostas. Além disso, ressalta-se também a equivalência entre eles, tendo ambos apresentado tempos de mesma ordem de grandeza, que não trazem diferenças significativas na execução das operações. Tratam-se, portanto, de duas opções absolutamente viáveis para trabalhar com modelos de processo. Neste sentido, a escolha do banco de dados a ser utilizado deve seguir outros critérios, tais como facilidade visual ou de realização de consultas para o usuário, por exemplo.

Outra limitação do trabalho diz respeito ao tamanho dos processos utilizados nos experimentos. Como no contexto de modelagem de processos é recomendado decompor processos maiores do que 50 elementos (MENDLING; REIJERS; AALST, 2010), buscou-se focar os estudos em processos compatíveis com esse porte. Não foi avaliado o comportamento dos modelos propostos ao serem defrontados com modelos de processo maiores, eventualmente criados pelos usuários. Por se tratarem de exceções e não serem usuais, priorizou-se manter a atenção em modelos de processo com perfis mais próximos dos que são amplamente utilizados.

4.2 Experimentos com Usuários

O objetivo desta Seção é apresentar os experimentos realizados com usuários, os quais tem o propósito de comparar os modelos de armazenamento propostos em termos de familiaridade com que os usuários lidam com eles. Para isso, cada usuário respondeu a um questionário, onde foi submetido a identificar modelos de processo dentre diferentes opções armazenadas em cada modelo. Buscou-se verificar a usabilidade dos modelos de armazenamento propostos tanto em termos visuais, quanto a capacidade de o usuário reconhecer uma consulta realizada na linguagem padrão de cada um dos bancos de dados. A seguir, são descritos os passos para a realização dos experimentos, bem como seus resultados.

4.2.1 Procedimento

O experimento consiste em um questionário⁵ (Apêndice B) com 28 perguntas no total, podendo ser realizado de forma presencial ou remota, uma vez que fica disponível em um site na Internet. Durante a resolução de todas as questões, é medido o tempo que o usuário leva para respondê-las, a fim de verificar o nível de hesitação tido por ele até escolher a resposta final. O público-alvo selecionado são pessoas que possuem certo conhecimento em bancos de dados relacionais, bancos de dados orientados a grafos ou em modelos de processos, pois acreditamos que este é o perfil de usuário com maior interesse nos modelos de armazenamento propostos. A exemplo dos experimentos de desempenho, foram utilizados os bancos MySQL e Neo4j para a criação de questões sobre bancos relacionais e não-relacionais, respectivamente. A modelagem de processos foi realizada com o auxílio da ferramenta criada no trabalho (GASSEN et al., 2014). Tal trabalho consiste em um estudo sobre as palavras empregadas para nomear elementos de um modelo de processo, utilizando ontologias para auxiliar na redução de problemas de ambiguidade causadas por elas. Para a realização dos experimentos com usuários em (GASSEN et al., 2014), foi construída uma ferramenta capaz de modelos processos utilizando os elementos básicos disponíveis em BPMN. Para o presente trabalho, utilizou-se essa ferramenta como base, expandindo suas funções de forma a permitir o armazenamento de modelos de processo nos bancos de dados MySQL e Neo4j. Também se acrescentou a funcionalidade de consulta à fragmentos de modelos de processo, capazes de retornar os modelos

⁵<http://mestrado.weblinker.com.br/rafael>

de processo armazenados nos bancos de dados que contenham o subprocesso buscado.

O questionário é dividido em 5 grupos de questões. O primeiro deles, composto por 6 perguntas, visa apresentar o trabalho e conhecer o perfil do usuário em questão. Para isso, são coletadas informações a respeito de seu nível de escolaridade, faixa etária, conhecimento em bancos de dados relacionais, conhecimento em bancos de dados orientados a grafos e conhecimentos em BPM. O segundo grupo tem por objetivo verificar a capacidade de identificação visual de um determinado modelo de processo, quando colocado lado a lado com um modelo armazenado em banco de dados baseado em grafos. Para isso, dado um grafo armazenado no banco de dados não-relacional, o usuário é desafiado a identificar qual o modelo de processo corresponde a ele, dentre 4 opções fornecidas. Este grupo de perguntas se inicia com um pequeno treinamento, a fim de familiarizar o usuário com os conceitos utilizados, bem como explicar os objetivos das questões ali contidas e fornecer uma explicação sobre como resolvê-las. O conteúdo encontrado no Apêndice B.3.1 mostra o treinamento realizado. O usuário é desafiado a realizar o experimento com dois modelos de processos, um de tamanho pequeno e outro médio. Após escolher a opção que considera correta, lhe é perguntado a respeito do quanto ele considera que ela é bem representada pelo grafo dado. Desta forma, o experimento avalia não apenas a correção da resposta fornecida, mas também o tempo necessário até que o usuário a escolha, bem como a sua percepção pessoal a respeito do modelo proposto.

O terceiro grupo de perguntas é semelhante ao segundo, porém voltado ao banco de dados relacional. Neste grupo, são fornecidos os dados essenciais para a construção do modelo de processo, contidos nas tabelas do banco de dados relacional. A partir deles, o usuário deve identificar, dentre 4 opções de modelos de processos, qual delas é representada pelos dados armazenados no banco relacional. À exemplo do grupo de questões anterior, este também possui um pequeno treinamento, bem como expõe o usuário a um modelo de processo pequeno e outro médio. Da mesma forma, são analisadas a correção da resposta, o tempo dispendido e a percepção do usuário a respeito da representação dada. O conteúdo encontrado no Apêndice B.4.1 apresenta o treinamento realizado neste grupo de questões.

O quarto grupo de questões, funciona da mesma forma dos anteriores, com treinamento prévio e análise de correção da resposta, tempo dispendido e percepção do usuário. Porém, seu objetivo é medir a capacidade de o usuário relacionar uma consulta realizada na linguagem Cypher - padrão do Neo4j - a um dos 4 modelos de processos dados. Para isso, é dada uma consulta responsável por retornar um modelo de processo que contém

certo fragmento buscado. O desafio do usuário é identificar, dentre as opções fornecidas, qual dos modelos de processo contém tal fragmento. O conteúdo encontrado no Apêndice B.5.1 apresenta o treinamento realizado neste grupo de questões.

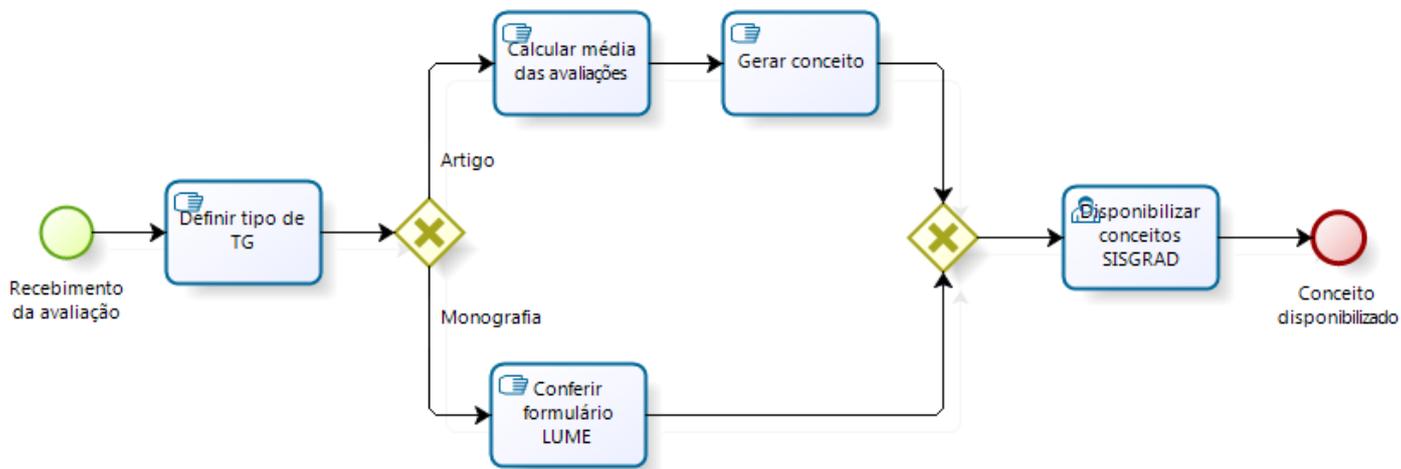
Por fim, o quinto e último grupo de questões é uma réplica do anterior, porém voltado ao modelo relacional. Visa analisar a capacidade de o usuário relacionar uma consulta realizada na linguagem SQL – padrão do MySQL – a um dos 4 modelos de processos dados. Da mesma forma que foi realizado no modelo não-relacional, é fornecido uma consulta que busca um fragmento de processo contido em um dos 4 modelos apresentados, sendo o usuário desafiado a descobrir qual deles possui tal fragmento. O conteúdo encontrado no Apêndice B.6.1 apresenta o treinamento realizado neste grupo de questões.

Entre cada grupo de questões, o usuário deve responder, ainda, um questionário com 10 perguntas de acordo com o método *System Usability Scale* (SUS) (BROOKE, 1996). Trata-se de uma escala numérica de usabilidade, que visa medir a satisfação do usuário ao interagir com um sistema.

Para a realização dos experimentos, foram utilizados dois modelos de processos reais, advindos da Sociedade Brasileira de Computação (SBC), um com tamanho pequeno e outro com tamanho médio. O primeiro modelo de processo versa sobre o recebimento de avaliações de trabalhos, geração e disponibilização de conceitos. O segundo processo, este um pouco maior, representa a entrega de correspondências para alunos e pesquisadores, por parte da secretária. Embora ambos os processos sejam da SBC, eles possuem domínios diferentes, uma vez que um deles versa sobre o gerenciamento de eventos, enquanto o outro é relativo ao cadastro de associados. Para a criação das questões e das alternativas que constam no questionário aplicado, estes modelos sofreram pequenas modificações de forma livre, a fim de identificar se os usuários são capazes de notar as diferenças entre os modelos de processo representados, por mais sensíveis que sejam. A Figura 4.5 exibe o primeiro modelo de processo, de tamanho menor, utilizado como base para o experimento. A Figura 4.6, por sua vez, exibe o modelo de processo relativo às entregas de correspondências.

Toda a documentação relativa aos experimentos consta no Apêndice B.

Figura 4.5: Processo de tamanho pequeno, utilizado como base para a criação de questões utilizadas no experimento.



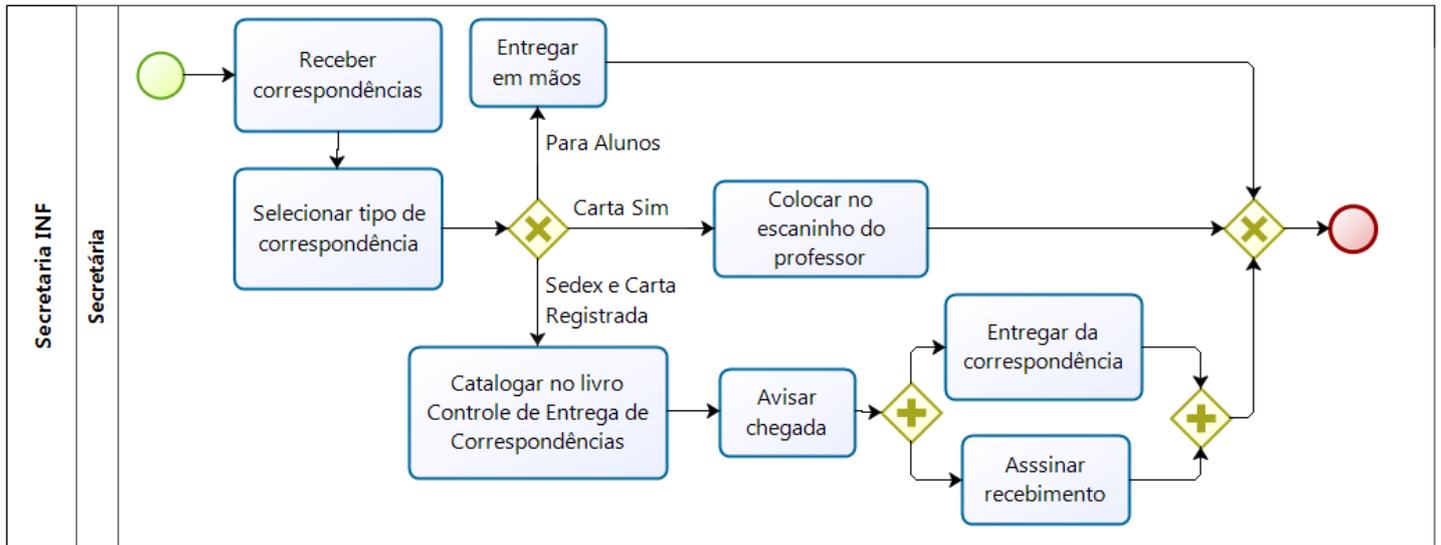
4.2.2 Participantes

Para participar do experimento, buscou-se pessoas com prévia experiência em bancos de dados relacionais, bancos de dados orientados a grafos ou BPM. O foco se deu nessas pessoas pois acredita-se que este seja o público-alvo, o perfil que possui grande potencial para utilização dos conceitos abordados pelo trabalho.

Sendo assim, de forma voluntária, participaram do experimento 13 pessoas, cujas idades e nível de escolaridade se concentraram em duas faixas principais: 38,5% tinham de 18 a 25 anos, enquanto 61,5% possuíam idade entre 26 e 30 anos. Com relação ao nível de escolaridade, 7,7% possuíam ensino médio completo, 38,4% eram alunos de graduação, 46,2% eram graduados e 7,7% eram mestres. A Figura 4.7 exibe a classificação dos participantes de acordo com a faixa etária e nível de escolaridade.

Com relação ao conhecimento sobre os assuntos abordados, 23,1% afirmou ter conhecimentos básicos sobre bancos de dados relacionais, 46,2% consideraram que possuem conhecimentos razoáveis e 30,7% acreditam ter um conhecimento aprofundado em bancos de dados relacionais. Considerando os bancos de dados orientados a grafos, 53,8% afirmaram já terem ouvido falar sobre o assunto, mas não terem conhecimentos básicos, enquanto 46,2% consideraram possuir conhecimentos básicos no assunto. Finalmente, em relação à BPM, 30,7% afirmaram já terem ouvido falar sobre o assunto mas não terem co-

Figura 4.6: Processo de tamanho médio, utilizado como base para a criação de questões utilizadas no experimento.

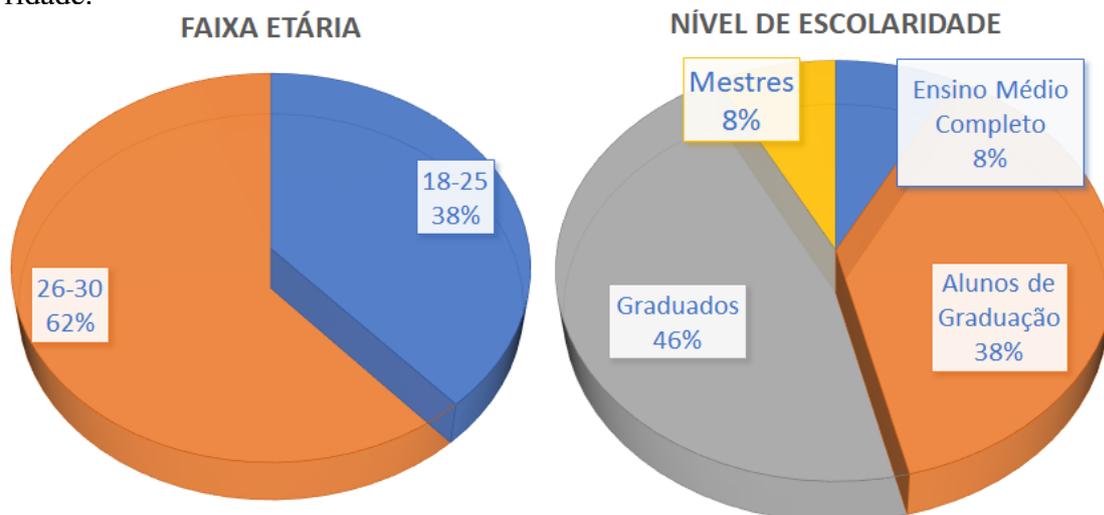


nhcimentos básicos, enquanto 38,5% afirmam ter conhecimentos básicos. Outros 15,4% afirmam possuir conhecimentos razoáveis e, por fim, mais 15,4% dos participantes afirmam possuir contato maior com BPM e conhecimentos mais sólidos sobre o assunto. A Figura 4.8 exibe a classificação dos participantes de acordo com os conhecimentos em bancos de dados relacionais, bancos de dados orientados a grafos e BPM.

4.2.3 Avaliação de Armazenamento em Banco de Dados Relacional

A avaliação do armazenamento em bancos de dados orientados a grafos se deu através da solução de duas questões, que constam na íntegra no Apêndice B.3. O enunciado de ambas é o seguinte: “A imagem abaixo representa um modelo de processo expresso dentro de determinado banco de dados relacional. Dentre as opções disponíveis, qual delas representa modelo de processo que contém o modelo mostrado?”. A seguir, foi apresentado um modelo de processo armazenado no banco de dados MySQL. A Figura 4.9 exibe o modelo de processo utilizado na primeira questão. Cinco alternativas foram dadas aos usuários, sendo que quatro delas continham a modelagem de processos em BPMN e a outra alternativa dava a opção de o usuário informar que não tinha conhecimento para responder a questão. Apenas uma das alternativas fornecidas corresponde ao

Figura 4.7: Classificação dos participantes de acordo com a faixa etária e nível de escolaridade.

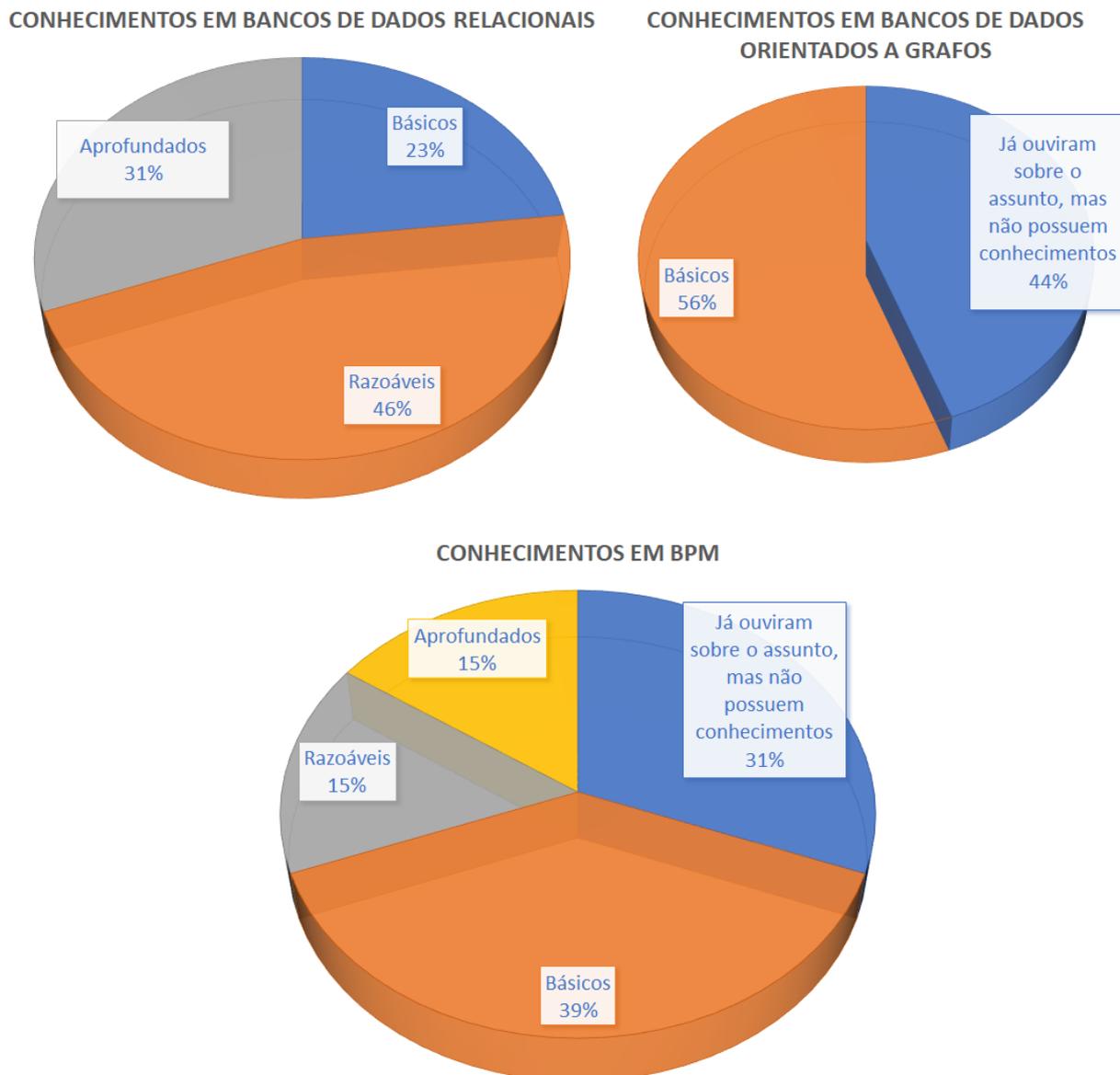


modelo de processo armazenado no banco de dados MySQL. As outras alternativas apresentavam pequenas alterações, realizadas a fim de avaliar a facilidade com que o usuário teria para identificá-las. A Figura 4.10 exibe a resposta correta da primeira questão, enquanto a Figura 4.11 exemplifica uma alternativa incorreta (os círculos vermelhos em destaque apontam a inversão da posição dos elementos *Task* e *Sub Process*, tornando a alternativa incorreta). Os resultados, em termos de performance e satisfação, estão listados a seguir.

4.2.3.1 Performance

A performance dos usuários na solução das questões foi medida tanto em termos de tempo para fornecerem suas respostas, quanto na própria correção delas. O tempo médio para solucionar a questão 1 foi de 206 segundos, enquanto a questão 2 teve 229 segundos de média para ser respondida. Com relação à precisão das respostas dadas, o aproveitamento foi de 92,3% acertos na questão 1 e 76,7% de acertos na questão 2. É possível que os erros encontrados estejam associados ao menor contato prévio do usuário com bancos de dados relacionais, uma vez que 60% dos erros cometidos foram feitos pela pequena parcela de usuários que afirmou não possuir bom conhecimento na área. A Figura 4.12 retrata a performance dos usuários na avaliação de armazenamento de modelos de processo em banco de dados relacional.

Figura 4.8: Classificação dos participantes de acordo com os conhecimentos em bancos de dados relacionais, bancos de dados orientados a grafos e BPM.



4.2.3.2 Satisfação

Para avaliar as técnicas na dimensão satisfação, foi solicitado o preenchimento de um questionário de acordo com o método SUS. No caso da avaliação do armazenamento em bancos relacionais, foi obtido um escore médio de 59,03 com um valor mínimo igual a 25 e máximo igual a 80. De acordo com a interpretação do método SUS, este resultado pode ser considerado ruim, estando na segunda pior faixa entre as cinco existentes.

Figura 4.9: Modelo de processo armazenado em banco de dados MySQL utilizado na questão 1 do experimento de armazenamento de processos em banco de dados relacional.

Tabela "Elementos"		Tabela "Arestas"	
Ele_Codigo	Ele_Style	Are_From	Are_To
10	Start Event	14	15
11	Task	10	11
12	Task	13	18
13	Task	16	13
14	Task	16	12
15	End Event	11	16
16	XOR	12	17
17	XOR	18	17
18	Sub Process	17	14

4.2.4 Avaliação de Armazenamento em Banco de Dados Orientado a Grafos

A avaliação do armazenamento em bancos de dados orientados a grafos se deu através da solução de duas questões, que constam na íntegra no Apêndice B.3. O enunciado de ambas é o seguinte: “A imagem abaixo representa um modelo de processo expresso dentro de determinado banco de dados não relacional. Dentre as opções disponíveis, qual delas representa modelo de processo que contém o modelo mostrado?”. A seguir, foi apresentado um modelo de processo armazenado no banco de dados Neo4j. A Figura 4.13 exibe o modelo de processo utilizado na primeira questão. Cinco alternativas foram dadas aos usuários, sendo que quatro delas continham a modelagem de processos em BPMN e a outra alternativa dava a opção do usuário informar que não tinha conhecimento para responder a questão. Apenas uma das alternativas fornecidas corresponde ao modelo de processo armazenado no banco de dados Neo4j. As outras alternativas apresentavam pequenas alterações, realizadas a fim de avaliar a facilidade com que o usuário teria para

Figura 4.10: Alternativa correta para a primeira questão do experimento de armazenamento de processos em banco de dados relacional.

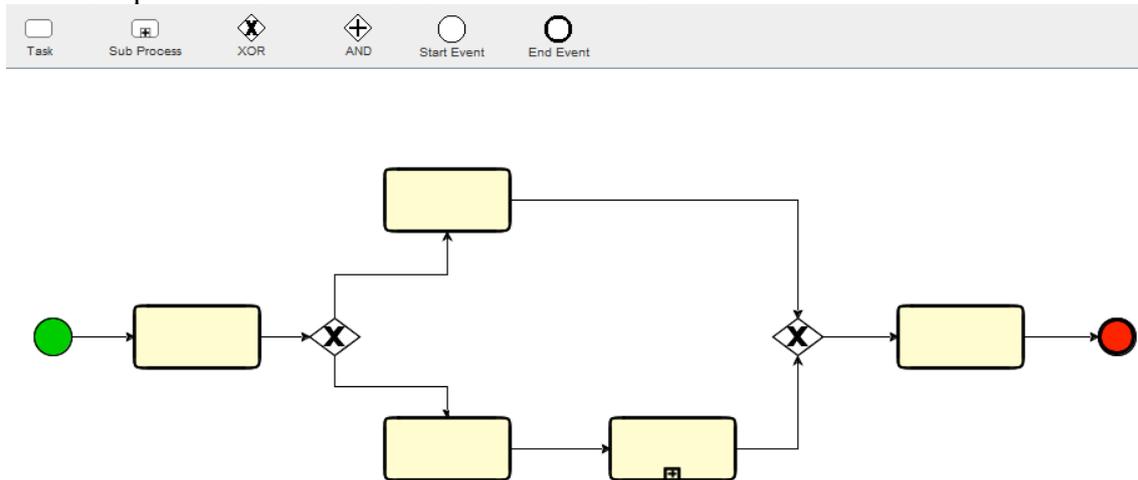
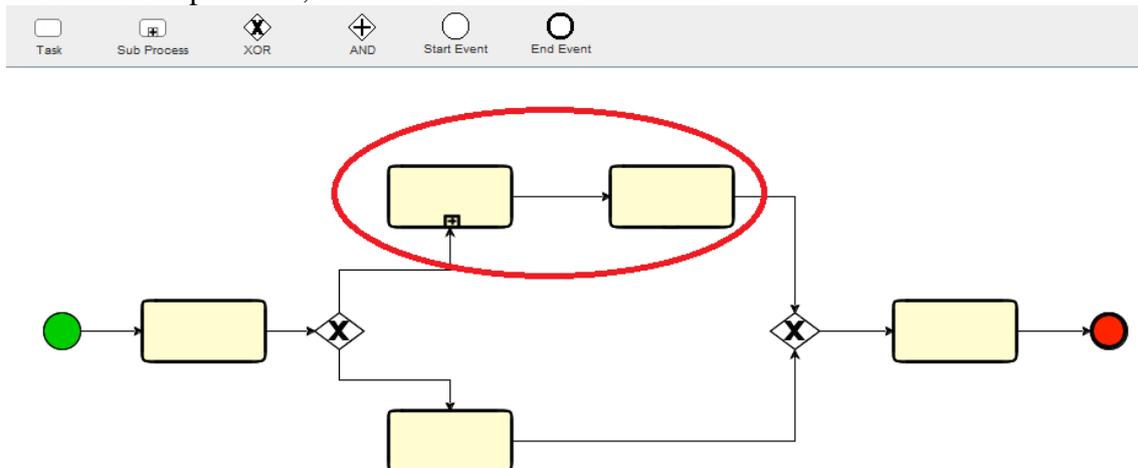


Figura 4.11: Exemplo de alternativa incorreta para a primeira questão do experimento de armazenamento de processos em banco de dados relacional. O círculo em vermelho destaca a inversão da posição dos elementos Task e Sub Process originalmente utilizados no modelo de processo, tornando a alternativa incorreta.

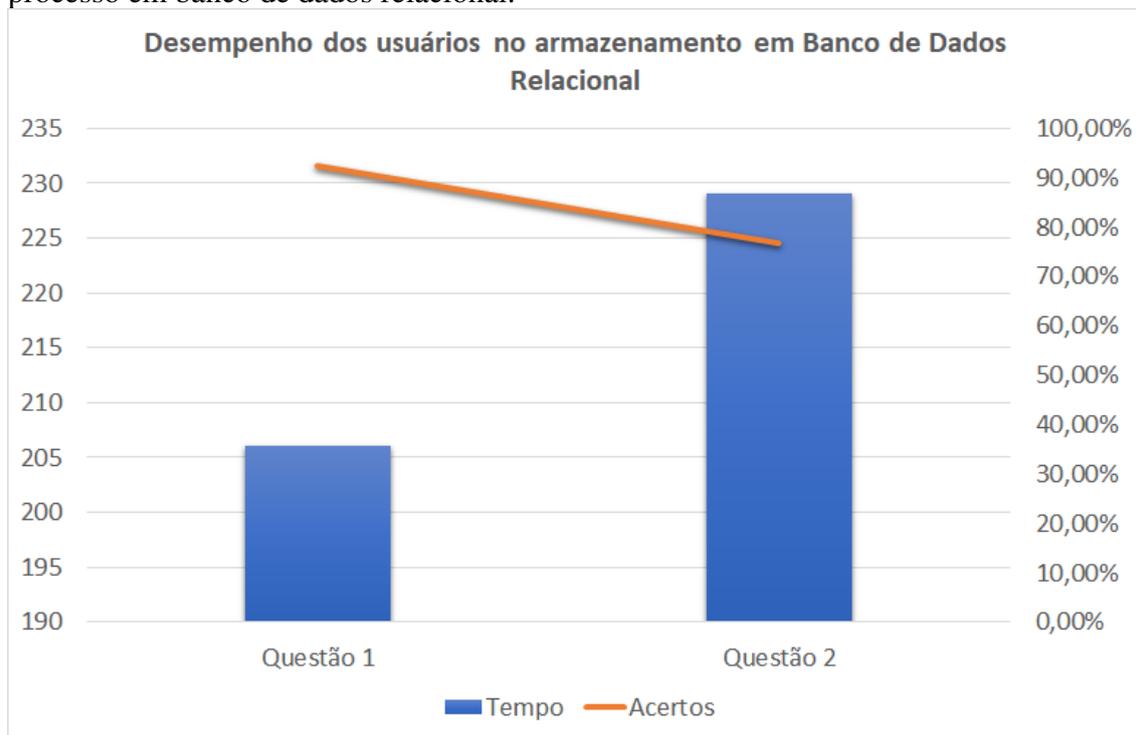


identificá-las. A Figura 4.14 exibe a resposta correta da primeira questão, enquanto a Figura 4.15 exemplifica uma alternativa incorreta (os círculos vermelhos em destaque apontam a substituição de elementos XOR por AND, tornando a alternativa incorreta). Os resultados, em termos de performance e satisfação, estão listados a seguir.

4.2.4.1 Performance

A performance dos usuários na solução das questões foi medida tanto em termos de tempo para fornecerem suas respostas, quanto na própria correção delas. O tempo médio para solucionar a questão 1 foi de 113 segundos, enquanto a questão 2 teve 126 segundos de média para ser respondida. Com relação à precisão das respostas dadas,

Figura 4.12: Performance dos usuários na avaliação de armazenamento de modelos de processo em banco de dados relacional.



notou-se um excelente aproveitamento dos usuários, com 100% de acertos na questão 1 e 84,6% de acertos na questão 2. É possível que os poucos erros encontrados estejam associados ao pequeno tempo dispendido sobre a questão, uma vez que os dois participantes envolvidos tiverem tempos menores do que a média – 61 e 72 segundos. A Figura 4.16 retrata a performance dos usuários na avaliação de armazenamento de modelos de processo em banco de dados orientado a grafos.

4.2.4.2 Satisfação

Para avaliar as técnicas na dimensão satisfação, foi solicitado o preenchimento de um questionário de acordo com o método SUS. No caso da avaliação do armazenamento em bancos de dados orientados a grafos, foi obtido um escore médio de 82,69 com um valor mínimo igual a 67,5 e máximo igual a 95. De acordo com a interpretação do método SUS, este resultado pode ser considerado muito bom. Notou-se um excelente retorno de satisfação por parte do grupo de usuários que afirmou possuir conhecimento prévio em BPM, onde o escore médio foi igual a 90,83, considerado excelente.

Figura 4.13: Modelo de processo armazenado em banco de dados Neo4j utilizado na questão 1 do experimento de armazenamento de processos em banco de dados orientado a grafos.

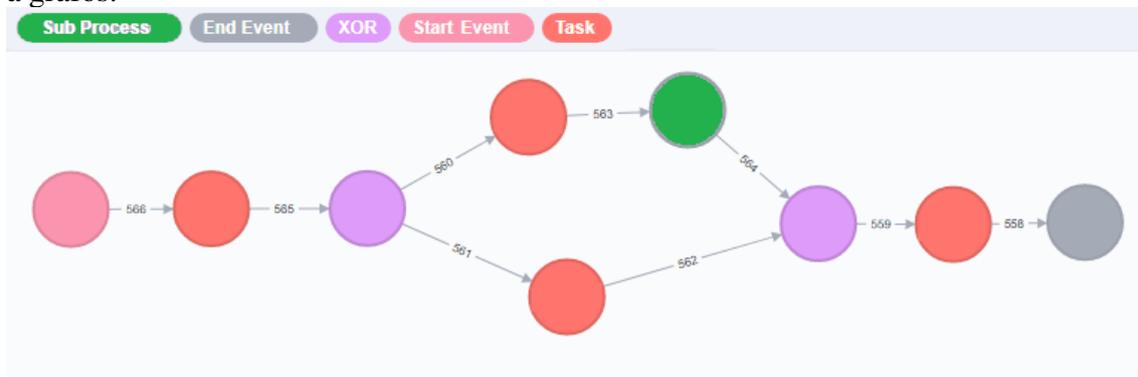
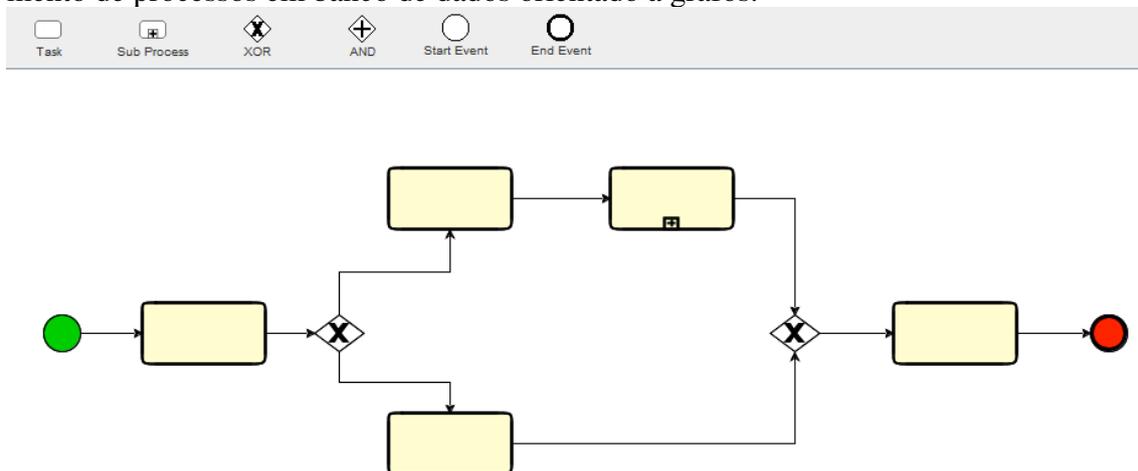


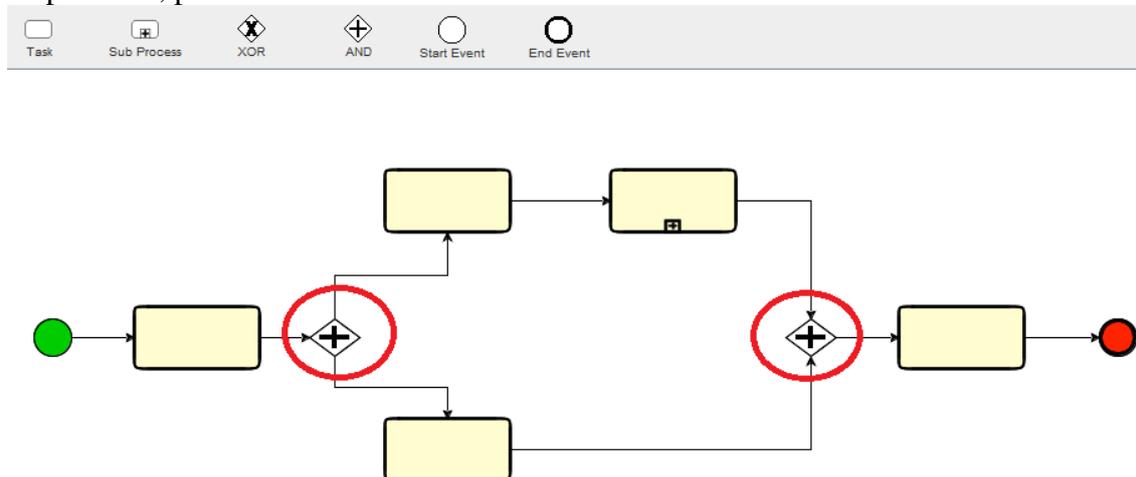
Figura 4.14: Alternativa correta para a primeira questão do experimento de armazenamento de processos em banco de dados orientado a grafos.



4.2.5 Avaliação de Consultas em Banco de Dados Relacional

A avaliação da consulta por fragmentos de modelos de processos em bancos de dados relacionais se deu através da solução de duas questões, que constam na íntegra no Apêndice B.6. O enunciado de ambas é o seguinte: “A imagem abaixo representa uma consulta realizada em determinada linguagem de banco de dados relacional, que retorna um modelo de processo que contém o fragmento buscado na consulta. Dentre as opções disponíveis, qual delas representa o modelo que contém o fragmento buscado pela consulta?”. A seguir, foi apresentada uma consulta de modelo de processo armazenado no banco de dados MySQL. A Figura 4.17 exibe a consulta de fragmento de modelo de processo utilizado na primeira questão. Cinco alternativas foram dadas aos usuários, sendo que quatro delas continham a modelagem de processos em BPMN e a outra alternativa dava a opção do usuário informar que não tinha conhecimento para responder a questão. Apenas uma das alternativas fornecidas continha o fragmento de modelo de processo

Figura 4.15: Exemplo de alternativa incorreta para a primeira questão do experimento de armazenamento de processos em banco de dados orientado a grafos. Os círculos em vermelho destacam a substituição dos elementos XOR originalmente utilizado no modelo de processo, por elementos AND.



buscado pela consulta realizada fornecida. A Figura 4.18 exibe a resposta correta da primeira questão (o círculo em vermelho destaca o fragmento buscado), enquanto a Figura 4.19 exemplifica uma alternativa incorreta, que não continha o fragmento de processo em questão. Os resultados, em termos de performance e satisfação, estão listados a seguir.

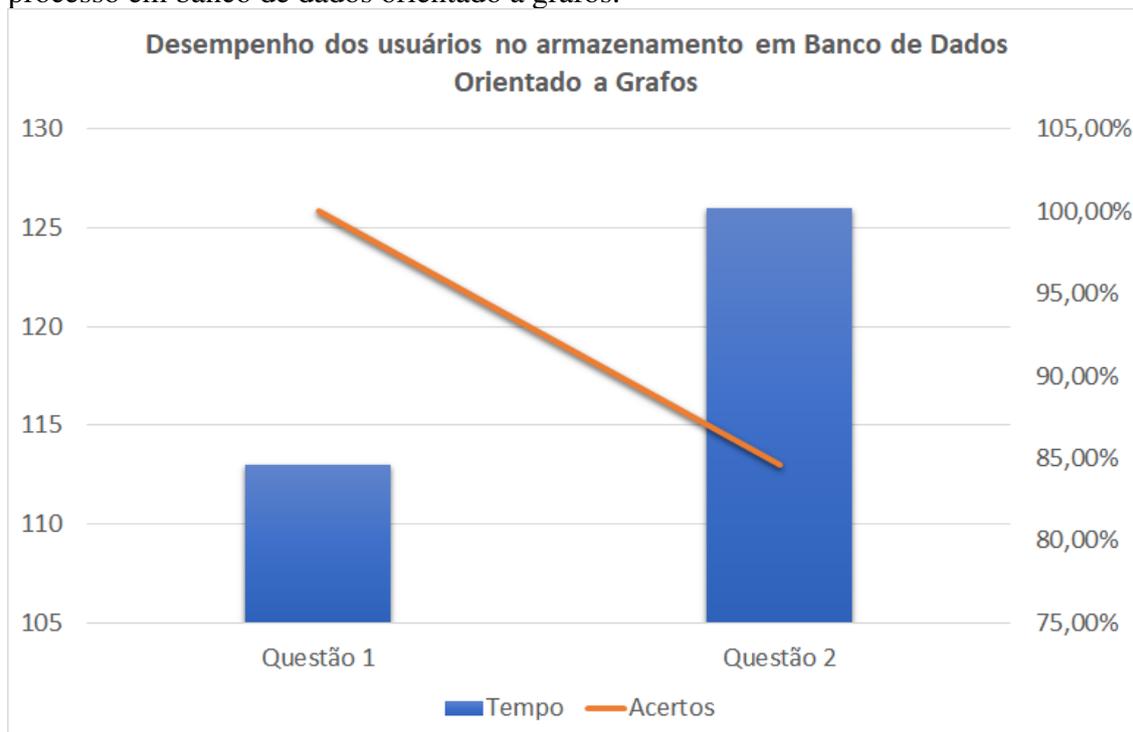
4.2.5.1 Performance

A performance dos usuários na solução das questões foi medida tanto em termos de tempo para fornecerem suas respostas, quanto na própria correção delas. O tempo médio para solucionar a questão 1 foi de 72 segundos, enquanto a questão 2 teve 176 segundos de média para ser respondida. Com relação à precisão das respostas dadas, o aproveitamento foi de 100% acertos na questão 1 e 84,6% de acertos na questão 2. Os erros cometidos se concentram nos usuários que possuem pouca experiência na área, uma vez que 100% dos erros cometidos foram por participantes que afirmaram não possuir sólidos conhecimentos em bancos de dados relacionais. A Figura 4.20 retrata a performance dos usuários na avaliação de consultas de modelos de processo em banco de dados relacional.

4.2.5.2 Satisfação

Para avaliar as técnicas na dimensão satisfação, foi solicitado o preenchimento de um questionário de acordo com o método SUS. No caso da avaliação da consulta por

Figura 4.16: Performance dos usuários na avaliação de armazenamento de modelos de processo em banco de dados orientado a grafos.



fragmentos em bancos relacionais, foi obtido um escore médio de 73,46 com um valor mínimo igual a 35 e máximo igual a 92,5. De acordo com a interpretação do método SUS, este resultado pode ser considerado bom, estando na terceira faixa entre as cinco existentes.

4.2.6 Avaliação de Consultas em Banco de Dados Orientado a Grafos

A avaliação da consulta por fragmentos em bancos de dados orientados a grafos se deu através da solução de duas questões, que constam no Apêndice B.5. O enunciado de ambas é o seguinte: “A imagem abaixo representa uma consulta realizada em determinada linguagem de banco de dados não relacional, que retorna um modelo de processo que contém o fragmento buscado na consulta. Dentre as opções disponíveis, qual delas representa o modelo que contém o fragmento buscado pela consulta?”. A seguir, foi apresentada uma consulta de modelo de processo armazenado no banco de dados Neo4j. A Figura 4.21 exibe a consulta de fragmento de modelo de processo utilizado na primeira questão. Cinco alternativas foram dadas aos usuários, sendo que quatro delas continham a modelagem de processos em BPMN e a outra alternativa dava a opção do usuário informar que não tinha conhecimento para responder a questão. Apenas uma das alternativas

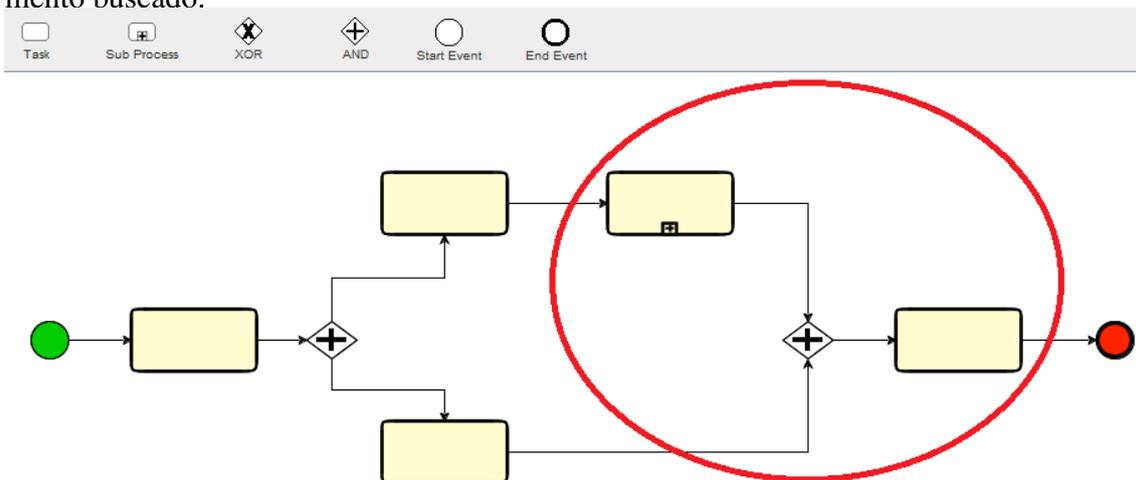
Figura 4.17: Consulta de fragmento de modelo de processo armazenado em banco de dados MySQL utilizado na questão 1 do experimento de consulta de processos em banco de dados relacional.

```

1 SELECT DISTINCT(e1.Pro_Codigo) AS Pro_Codigo
2 FROM elementos e1, elementos e2, elementos e3, arestas a1,
   arestas a2
3 WHERE 1
4 AND e1.Ele_Style = 'SubProcess'
5 AND e2.Ele_Style = 'AND'
6 AND e3.Ele_Style = 'Task'
7 AND e1.Ele_Codigo = a1.Are_From
8 AND e2.Ele_Codigo = a1.Are_To
9 AND e2.Ele_Codigo = a2.Are_From
10 AND e3.Ele_Codigo = a2.Are_To

```

Figura 4.18: Alternativa correta para a primeira questão do experimento de consulta de modelos de processos em banco de dados relacional. O círculo vermelho destaca o fragmento buscado.

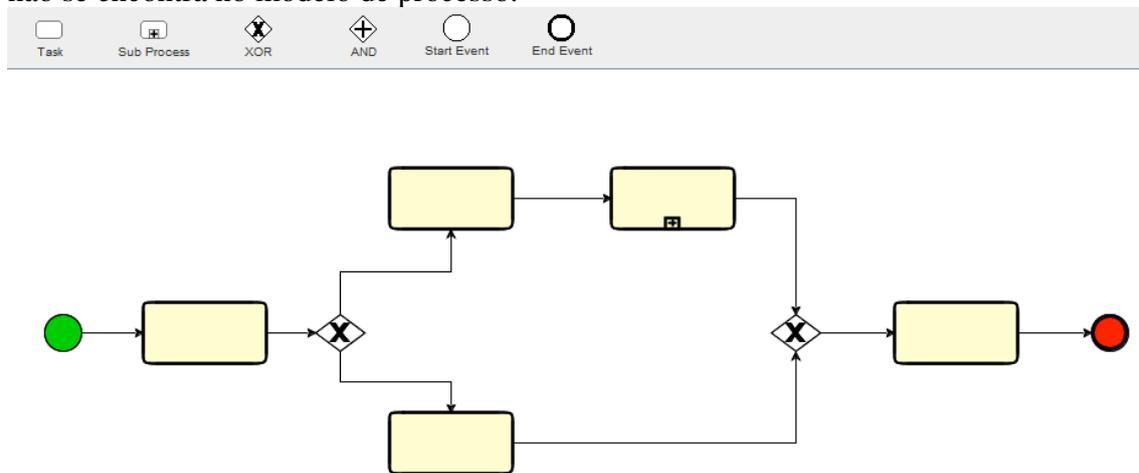


fornechas continha o fragmento de modelo de processo buscado pela consulta realizada fornecida. A Figura 4.22 exibe a resposta correta da primeira questão (o círculo em vermelho destaca o fragmento buscado), enquanto a Figura 4.23 exemplifica uma alternativa incorreta, que não continha o fragmento de processo em questão. Os resultados, em termos de performance e satisfação, estão listados a seguir.

4.2.6.1 Performance

A performance dos usuários na solução das questões foi medida tanto em termos de tempo para fornecerem suas respostas, quanto na própria correção delas. O tempo médio para solucionar a questão 1 foi de 77 segundos, enquanto a questão 2 teve 95 segundos de média para ser respondida. Com relação à precisão das respostas dadas,

Figura 4.19: Exemplo de alternativa incorreta para a primeira questão do experimento de consulta de modelos de processos em banco de dados relacional. O fragmento destacado não se encontra no modelo de processo.



o aproveitamento foi de 92,3% de acertos na questão 1 e 92,3% de acertos na questão 2. Não foi notado nenhum dado que indique alguma tendência significativa relativa aos erros cometidos pelos usuários. A Figura 4.24 retrata a performance dos usuários na avaliação de consultas de modelos de processo em banco de dados orientado a grafos.

4.2.6.2 Satisfação

Para avaliar as técnicas na dimensão satisfação, foi solicitado o preenchimento de um questionário de acordo com o método SUS. No caso da avaliação do armazenamento em bancos de dados orientados a grafos, foi obtido um escore médio de 77,69 com um valor mínimo igual a 50 e máximo igual a 90. De acordo com a interpretação do método SUS, este resultado pode ser considerado bom, estando na terceira faixa entre as cinco existentes.

4.3 Análise Geral dos Resultados

Os experimentos demonstraram um bom número de acertos dos usuários para ambos os modelos de bancos de dados, seja o relacional ou o orientado a grafos. Este fator leva a crer na boa capacidade de representação dos métodos propostos, bem como de recuperação dos dados armazenados, ao menos dentro do grupo de usuários pré-estabelecido como o público-alvo do presente trabalho. Pessoas com conhecimentos mais sólidos em bancos de dados relacionais não tiveram grandes dificuldades em fornecerem as respostas

Figura 4.20: Performance dos usuários na avaliação de consultas de modelos de processo em banco de dados relacional.

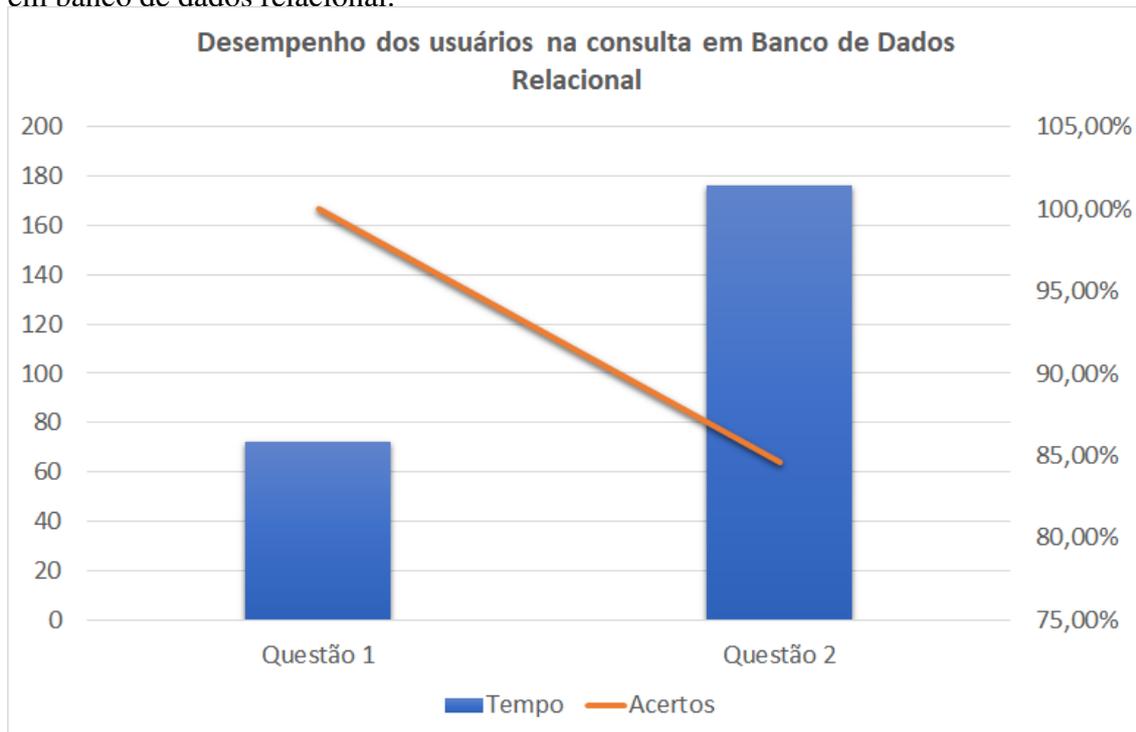


Figura 4.21: Consulta de fragmento de modelo de processo armazenado em banco de dados Neo4j utilizado na questão 1 do experimento de consulta de processos em banco de dados orientado a grafos.

```

1 MATCH (a:XOR) -[:JOIN]->(b:SubProcess),
2 (b:SubProcess) -[:JOIN]->(c:Task)
3 WITH DISTINCT a.graphId as graphId RETURN graphId

```

corretas. No caso do método de armazenamento em bancos de dados orientados a grafos, nem mesmo os usuários com menor experiência com bancos de dados tiveram grandes problemas em solucionar as questões.

O tempo médio dispendido em cada questão teve uma diferença menos significativa nos experimentos de consulta de fragmentos de modelos de processo, se compararmos os métodos em bancos de dados relacional e NoSQL. Para identificar os modelos de processo armazenados, entretanto, os usuários precisaram de um menor tempo no método que utiliza bancos de dados orientados a grafos. Acredita-se que a representação visual dos dados armazenados, por ser próxima ao modelo de processo original, tem papel fundamental na identificação mais imediata das informações armazenadas por parte do usuário. As tabelas 4.1 e 4.2 exibem um comparativo entre os métodos propostos ao serem expostos nos experimentos de armazenamento e consulta de fragmentos de modelos de processos, respectivamente.

Figura 4.22: Alternativa correta para a primeira questão do experimento de consulta de modelos de processos em banco de dados orientados a grafos. O círculo vermelho destaca o fragmento buscado.

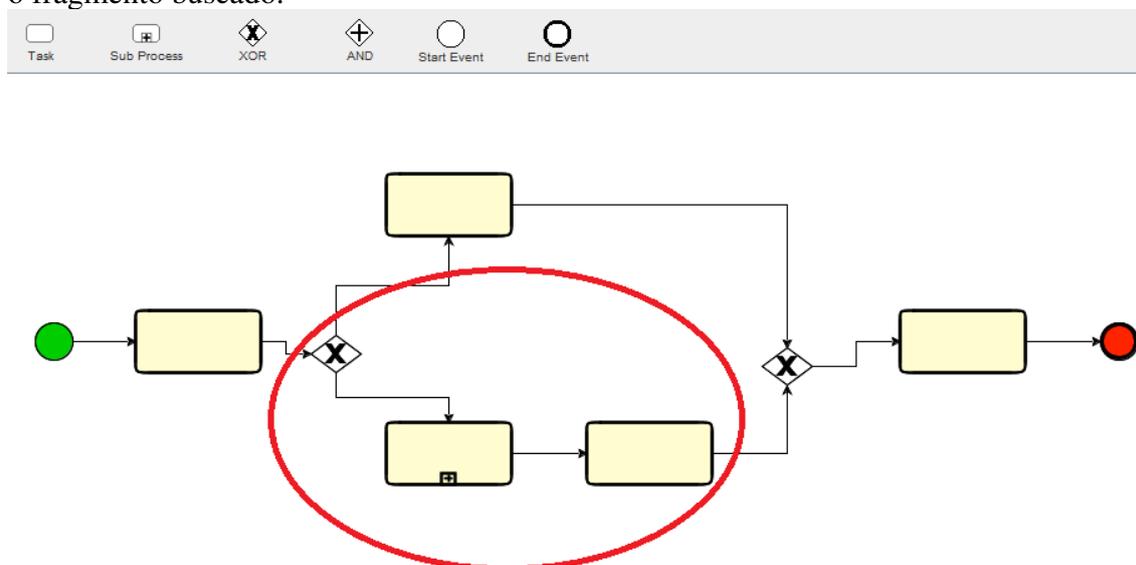


Tabela 4.1: Comparativo entre o modelo relacional e o orientado a grafos nos experimentos de armazenamento de modelos de processos.

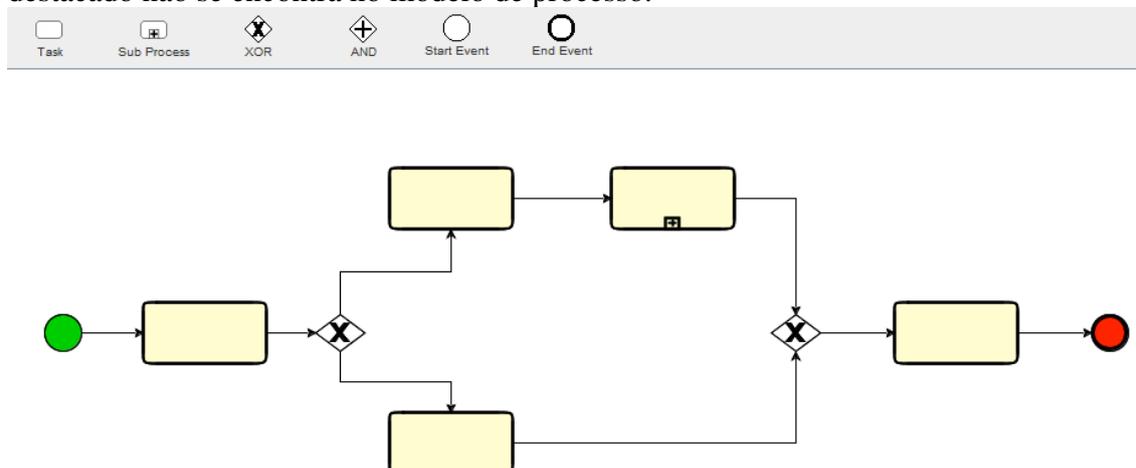
	Questão 1 - Tempo médio (s)	Questão 2 - Tempo médio (s)	Questão 1 - Acertos (%)	Questão 2 - Acertos (%)
Relacional	206	229	92,3	76,7
Grafos	113	126	100	84,6

Tabela 4.2: Comparativo entre o modelo relacional e o orientado a grafos nos experimentos de consultas de fragmentos de modelos de processos.

	Questão 1 - Tempo médio (s)	Questão 2 - Tempo médio (s)	Questão 1 - Acertos (%)	Questão 2 - Acertos (%)
Relacional	72	176	100	84,6
Grafos	77	95	92,3	92,3

Com relação à satisfação dos usuários com cada método, notou-se larga vantagem para o banco de dados orientado a grafos no que se refere ao armazenamento. Analisando o formulário SUS, itens como a facilidade de uso, a menor complexidade do método e a impressão de que as pessoas serão capazes de aprender o método rapidamente se mostraram determinantes para que a proposta de armazenamento em bancos de dados orientados a grafos se sobressaísse ao banco de dados relacional (escores médios 82,69 e 59,03, respectivamente). Considerando a realização de consultas, o resultado foi muito mais próximo entre ambos os métodos: enquanto o modelo relacional obteve um escore médio de 73,46, o modelo não relacional obteve 77,69. A favor do método não-relacional, os usuários consideraram a facilidade de aprendizagem, que teve 4,3 como média de pontuação, contra 3,8 do método relacional. Por outro lado, a maior experiência dos usuários

Figura 4.23: Exemplo de alternativa incorreta para a primeira questão do experimento de consulta de modelos de processos em banco de dados orientado a grafos. O fragmento destacado não se encontra no modelo de processo.



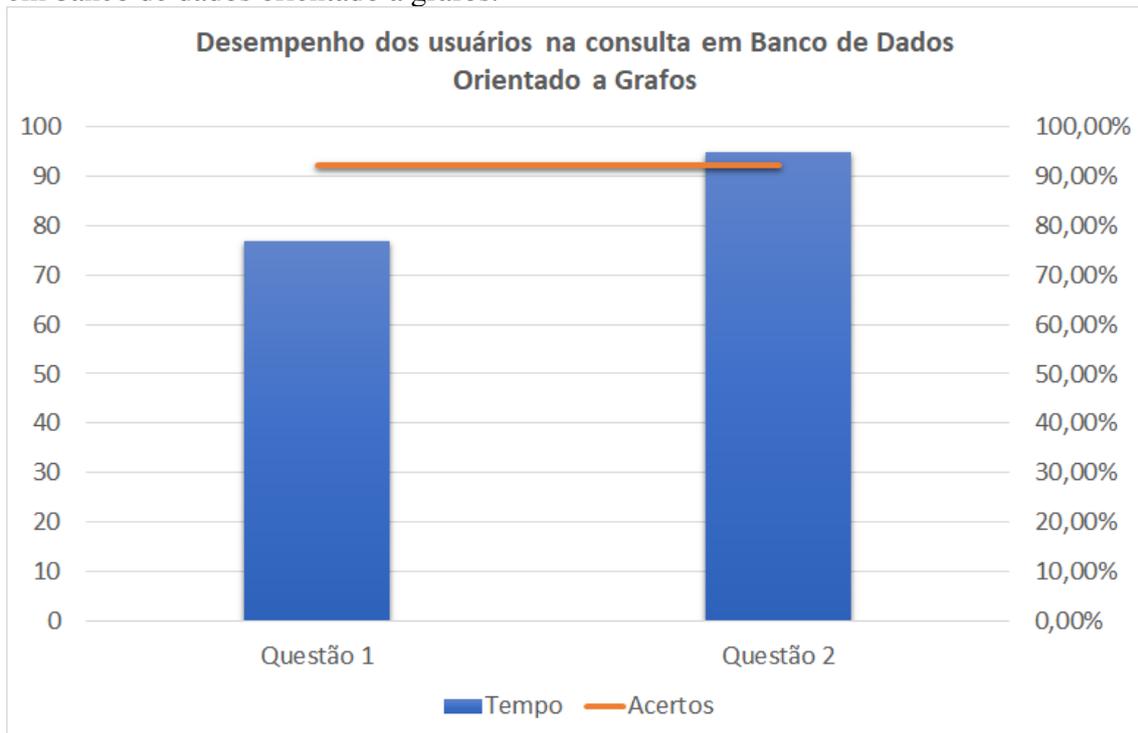
com o modelo relacional trouxe maior equivalência entre os métodos no resultado final: enquanto os usuários deram uma pontuação média de 1,6 para a necessidade de aprender conceitos novos no modelo relacional, foi dada uma pontuação de 2,3 para o modelo não-relacional. A Tabela 4.3 mostra a comparação dos resultados SUS para os experimentos realizados. É interessante notar também a boa recepção ao método por parte dos usuários com boa experiência em BPM, onde foram unânimes em preferir o modelo não-relacional de armazenamento. Nas consultas, 75% deles afirmaram preferir o método de armazenamento em bancos de dados orientados a grafos, enquanto apenas 25% consideram os métodos equivalentes.

Tabela 4.3: Comparação de resultados do método SUS para os experimentos de satisfação realizados.

	Escore mé- dio	Resultado	Escore má- ximo	Escore mí- nimo
Armazenamento Relacio- nal	59,03	Ruim	80	25
Armazenamento Grafos	82,69	Muito bom	95	67,5
Consulta Relacional	73,46	Bom	92,5	35
Consulta Grafos	77,69	Bom	90	50

Diante de tais dados, é possível compreender a relevância e a importância de considerar os diferentes métodos de armazenamento de modelos de processo. Apesar da grande maioria dos trabalhos atuais utilizarem bancos de dados relacionais, a presente pesquisa aponta a capacidade dos bancos de dados orientados a grafos em atuar de maneira competitiva. Armazenar modelos de processos em bancos de dados NoSQL pode trazer ganhos em termos de performance, escalabilidade e flexibilidade, abrindo um ca-

Figura 4.24: Performance dos usuários na avaliação de consultas de modelos de processo em banco de dados orientado a grafos.



minho interessante para gerenciar grandes coleções de modelos de processo em nuvem, por exemplo. Porém, pensando além destas capacidades objetivas, os experimentos realizados demonstraram satisfação e maior familiaridade dos usuários com o armazenamento em bancos de dados orientados a grafos. Utilizá-los pode tornar a área de BPM mais compreensível e interessante mesmo a usuários leigos, assim trazendo, talvez, ainda mais popularidade ao assunto.

5 CONCLUSÃO

Neste trabalho foi introduzido o problema do armazenamento de processos, assunto com potencial para construir soluções que atendam interesses tanto acadêmicos quanto empresariais. Uma pesquisa bibliográfica sobre os conceitos relacionados ao gerenciamento de processos de negócio, incluindo seu armazenamento em repositórios tradicionais, foi realizada. Também foram estudadas as participações de diferentes tipos de bancos de dados, em especial os relacionais e os orientados a grafos, no atual cenário de armazenamento de processos. Alguns dos artigos pesquisados se destacaram e tiveram uma breve explicação acerca de seus focos principais e suas respectivas contribuições para a presente pesquisa.

Na sequência deste levantamento bibliográfico, foi apresentada uma proposta para mapeamento de processos para bancos de dados relacionais e orientados a grafos. A seguir, foram realizados testes comparativos entre ambas as tecnologias e os métodos utilizados para manipular os dados em cada uma delas. As conclusões dos experimentos, bem como seus métodos de realização, foram mostradas, apontando certa vantagem para o modelo de banco de dados orientado a grafos. Seu desempenho é comparável ao banco de dados relacional, muitas vezes sendo superior, além de possuir capacidade de fornecer uma representação à qual o usuário possui uma familiaridade maior. Além dessas qualidades encontradas, ainda existe a vantagem inerente aos bancos de dados NoSQL, que é a facilidade em trabalhar com sistemas distribuídos.

Ao elaborar o presente estudo, notou-se certa discrepância entre a atenção dada para a modelagem de processos e para o seu devido modelo de armazenamento. Apesar do crescente reconhecimento da importância da área de BPM, seja no meio acadêmico ou empresarial, ainda são raras as soluções capazes de lidar com grandes coleções de processos de maneira eficiente e confiável. Não existem muitos estudos que comparem o comportamento dos processos ao serem armazenados de diferentes formas e em diversos tipos de bancos de dados. Esta situação torna-se ainda mais intensa se considerarmos o armazenamento em nuvem, onde o material resultante de pesquisas relacionadas ao tema é extremamente escasso.

Acredita-se que as observações presentes neste trabalho, em especial o apontamento de lacunas importantes que ainda não recebem a devida atenção, possam contribuir para a definição de uma linha principal que aponte métodos para que o problema seja solucionado. A constante evolução da área de BPM deve vir acompanhada de boas solu-

ções para o armazenamento de grandes coleções de processos. É necessário, portanto, um amadurecimento nos estudos e propostas apresentadas nesse sentido. Definir uma arquitetura capaz de suportar o gerenciamento de grandes coleções de processos é um desafio de imprescindível solução para que sejam atingidas a eficiência e confiabilidade adequadas no compartilhamento e otimização de processos.

REFERÊNCIAS

- ASAY, M. **NoSQL keeps rising, but relational databases still dominate big data**. 2016. Disponível em: <<http://www.techrepublic.com/article/nosql-keeps-rising-but-relational-databases-still-dominate-big-data/>>. Acesso em: setembro 2017. Available from Internet: <<http://www.techrepublic.com/article/nosql-keeps-rising-but-relational-databases-still-dominate-big-data/>>.
- BECKER, M.; LAUE, R. A comparative survey of business process similarity measures. **Computers in Industry**, v. 63, n. 2, p. 148–167, 2012. Available from Internet: <<http://dblp.uni-trier.de/db/journals/cii/cii63.html#BeckerL12>>.
- BREWER, E. A. Towards robust distributed systems. In: **Symposium on Principles of Distributed Computing (PODC)**. [s.n.], 2000. Available from Internet: <<http://www.cs.berkeley.edu/~brewer/cs262b-2004/PODC-keynote.pdf>>.
- BROOKE, J. **"SUS-A quick and dirty usability scale." Usability evaluation in industry**. CRC Press, 1996. Available from Internet: <<https://www.crcpress.com/product/isbn/9780748404605>>.
- DUMAS, M. et al. **Fundamentals of Business Process Management**. [S.l.]: Springer, 2013. I-XXVII, 1-399 p.
- FAUVET, M. et al. Managing Process Model Collections with APROMORE. In: MAGLIO, P. et al. (Ed.). **Proceedings of Service-Oriented Computing (ICSOC 2010)**. [S.l.]: Springer-Verlag, Berlin, 2010. (Lecture Notes in Computer Science, v. 6470), p. 699–701.
- GASSEN, J. B. et al. Business process modeling: Vocabulary problem and requirements specification. In: GOSSETT, K.; MCNELLY, B. J.; JONES, D. (Ed.). **SIGDOC**. ACM, 2014. p. 2:1–2:10. Available from Internet: <<http://dblp.uni-trier.de/db/conf/sigdoc/sigdoc2014.html#GassenMTO14>>.
- HECHT, R.; JABLONSKI, S. Nosql evaluation. In: **International Conference on Cloud and Service Computing**. [s.n.], 2011. p. 336–41. Available from Internet: <<https://scholar.google.de/scholar.bib?q=info:m5MMgURVKQAJ:scholar.google.com/&output=citation&scisig=AAGBfm0AAAAVNiYUG2ZAQokdZ8jGYiDglOgMwkJADa&scisf=4&hl=en>>.
- INSIDER, B. **Here is the Swedish startup behind the Paradise Papers — which made all hell break loose**. [S.l.], 2017. Available from Internet: <<http://nordic.businessinsider.com/the-swedish-startup-behind-the-paradise-leaks-2017-11>>.
- IPROCESS. **Um guia para iniciar estudos em BPMN (I): Atividades e sequência**. 2012. Disponível em: <<http://blog.iprocess.com.br/2012/11/um-guia-para-iniciar-estudos-em-bpmn-i-atividades-e-sequencia/>>. Acesso em: dezembro 2014. Available from Internet: <<http://blog.iprocess.com.br/2012/11/um-guia-para-iniciar-estudos-em-bpmn-i-atividades-e-sequencia/>>.
- JIN, T. et al. Efficient querying of large process model repositories. **Computers in Industry**, v. 64, n. 1, p. 41–49, 2013. Available from Internet: <<http://dblp.uni-trier.de/db/journals/cii/cii64.html#JinORHW13>>.

KHAN, W.; AHMED, E.; SHAHZAD, W. Predictive performance comparison analysis of relational & nosql graph databases. **International Journal of Advanced Computer Science and Applications**, v. 8, n. 5, 2017. Available from Internet: <<https://pdfs.semanticscholar.org/325b/be2c92072b16645a5cded3b9eda3cb97660b.pdf>>.

KITCHENHAM, B. et al. Systematic literature reviews in software engineering – a systematic literature review. **Information and Software Technology**, v. 51, n. 1, p. 7 – 15, 2009. ISSN 0950-5849. Special Section - Most Cited Articles in 2002 and Regular Research Papers. Available from Internet: <<http://www.sciencedirect.com/science/article/pii/S0950584908001390>>.

KUNDA, D.; PHIRI, H. A comparative study of nosql and relational database. **ZAMBIA INFORMATION COMMUNICATION TECHNOLOGY (ICT) JOURNAL**, v. 1, p. 1–4, 2017. Available from Internet: <<http://ictjournal.icict.org.zm/index.php/zictjournal/article/download/8/3>>.

LI, J. et al. Process model storage solutions: Proposition and evaluation. In: SONG, M.; WYNN, M. T.; LIU, J. (Ed.). **AP-BPM**. Springer, 2013. (Lecture Notes in Business Information Processing, v. 159), p. 56–66. Available from Internet: <<http://dblp.uni-trier.de/db/conf/apbpm/apbpm2013.html#LiW0Y13>>.

MAGDALENO, A. M.; WERNER, C. M. L.; ARAUJO, R. M. de. Reconciling software development models: A quasi-systematic review. **Journal of Systems and Software**, v. 85, n. 2, p. 351–369, 2012. Available from Internet: <<http://dblp.uni-trier.de/db/journals/jss/jss85.html#MagdalenoWA12>>.

MARKETS; MARKETS. **Business Process Management Market by Solution, by Service, by Deployment Type (Cloud, On-Premises), by Organization Size (SMES, Enterprises), by Business Function, by Industry Vertical, by Region (NA, Europe, APAC, MEA, LA) - Global Forecast to 2021**. 2016. Disponível em: <<http://www.marketsandmarkets.com/Market-Reports/business-process-management-market-157890056.html>>. Acesso em: setembro 2017. Available from Internet: <<http://www.marketsandmarkets.com/Market-Reports/business-process-management-market-157890056.html>>.

MENDLING, J.; REIJERS, H.; AALST, W. Seven Process Modeling Guidelines (7PMG). **Information and Software Technology**, v. 52, n. 2, p. 127–136, 2010.

NAYAK, A.; PORIYA, A.; POOJARY, D. Type of nosql databases and its comparison with relational databases. **International Journal of Applied Information Systems**, v. 5, n. 4, 2017. Available from Internet: <<http://research.ijais.org/volume5/number4/ijais12-450888.pdf>>.

NEO4J, I. **Cypher Introduction**. [S.l.], 2007. Available from Internet: <<http://neo4j.com/docs/developer-manual/current/cypher/#cypher-intro>>.

NEO4J, I. **openCypher**. [S.l.], 2015. Available from Internet: <<http://www.opencypher.org>>.

OMG. **Business Process Model and Notation**. 1997. Disponível em: <<http://www.bpmn.org/>>. Acesso em: dezembro 2014. Available from Internet: <<http://www.bpmn.org/>>.

OMG. **Documents Associated With Business Process Model And Notation (BPMN) Version 2.0**. 2011. Disponível em: <<http://www.omg.org/spec/BPMN/2.0/>>. Acesso em: dezembro 2014. Available from Internet: <<http://www.omg.org/spec/BPMN/2.0/>>.

PERSON, L. **NoSQL Market by Type (Key-Value Store, Document Databases, Column Based Stores and Graph Database) and Application (Data Storage, Metadata Store, Cache Memory, Distributed Data Depository, e-Commerce, Mobile Apps, Web Applications, Data Analytics and Social Networking) - Global Opportunity Analysis and Industry Forecast, 2013 - 2020**. 2015. Disponível em: <<https://www.alliedmarketresearch.com/NoSQL-market>>. Acesso em: setembro 2017. Available from Internet: <<https://www.alliedmarketresearch.com/NoSQL-market>>.

RAUT, A. B. Nosql database and its comparison with rdbms. **International Journal of Computational Intelligence Research**, v. 13, n. 7, p. 1645–1651, 2017. Available from Internet: <http://www.ripublication.com/ijcir17/ijcirv13n7_08.pdf>.

RESEARCH, O. **Business Process Management Market Global Analysis**. 2017. Disponível em: <<http://www.reuters.com/brandfeatures/venture-capital/article?id=4462>>. Acesso em: setembro 2017. Available from Internet: <<http://www.reuters.com/brandfeatures/venture-capital/article?id=4462>>.

ROSA, M. et al. APROMORE: An Advanced Process Model Repository. **Expert Systems With Applications**, v. 38, n. 6, p. 7029–7040, 2011.

SGANDERLA, K. **Um guia para iniciar estudos em BPMN (V): Subprocessos**. 2012. Disponível em: <<http://blog.iprocess.com.br/2012/12/um-guia-para-iniciar-estudos-em-bpmn-v-subprocessos/>>. Acesso em: setembro 2017. Available from Internet: <<http://blog.iprocess.com.br/2012/12/um-guia-para-iniciar-estudos-em-bpmn-v-subprocessos/>>.

YAN, Z.; DIJKMAN, R. M.; GREFEN, P. W. P. J. Business process model repositories - framework and survey. **Information and Software Technology**, v. 54, n. 4, p. 380–395, 2012. Available from Internet: <<http://dblp.uni-trier.de/db/journals/infosof/infosof54.html#YanDG12>>.

**APÊNDICE A — SCRIPT EM LINGUAGEM SQL PARA MAPEAMENTO DE
MODELOS DE PROCESSO PARA O BANCO DE DADOS MYSQL**

```
--  
-- Estrutura da tabela `arestas`  
--  
  
CREATE TABLE `arestas` (  
  `Are_Codigo` int(11) NOT NULL,  
  `Are_Id` int(11) NOT NULL,  
  `Are_Style` varchar(45) NOT NULL,  
  `Are_ParentId` int(11) NOT NULL,  
  `Are_Value` varchar(45) NOT NULL,  
  `Are_From` int(11) NOT NULL,  
  `Are_To` int(11) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;  
  
-----  
  
--  
-- Estrutura da tabela `arestas_pontos`  
--  
  
CREATE TABLE `arestas_pontos` (  
  `ArePon_Codigo` int(11) NOT NULL,  
  `ArePon_X` int(11) NOT NULL,  
  `ArePon_Y` int(11) NOT NULL,  
  `Are_Codigo` int(11) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;  
  
-----  
  
--
```

```
-- Estrutura da tabela `elementos`
--

CREATE TABLE `elementos` (
  `Ele_Codigo` int(11) NOT NULL,
  `Ele_Id` int(11) NOT NULL,
  `Ele_Style` varchar(45) NOT NULL,
  `Ele_ParentId` int(11) NOT NULL,
  `Ele_Value` varchar(45) NOT NULL,
  `Ele_X` int(11) NOT NULL,
  `Ele_Y` int(11) NOT NULL,
  `Ele_H` int(11) NOT NULL,
  `Ele_W` int(11) NOT NULL,
  `Pro_Codigo` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-----

--
-- Estrutura da tabela `processos`
--

CREATE TABLE `processos` (
  `Pro_Codigo` int(11) NOT NULL,
  `Pro_Nome` varchar(45) NOT NULL,
  `Pro_Xml` text NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Indexes for dumped tables
--

--
-- Indexes for table `arestas`
```

```
--  
ALTER TABLE `arestas`  
  ADD PRIMARY KEY (`Are_Codigo`),  
  ADD KEY `fk_arestas_elementos1_idx` (`Are_From`),  
  ADD KEY `fk_arestas_elementos2_idx` (`Are_To`);  
  
--  
-- Indexes for table `arestas_pontos`  
--  
ALTER TABLE `arestas_pontos`  
  ADD PRIMARY KEY (`ArePon_Codigo`),  
  ADD KEY `fk_arestas_pontos_arestas1_idx` (`Are_Codigo`);  
  
--  
-- Indexes for table `elementos`  
--  
ALTER TABLE `elementos`  
  ADD PRIMARY KEY (`Ele_Codigo`),  
  ADD KEY `fk_elementos_processos_idx` (`Pro_Codigo`);  
  
--  
-- Indexes for table `processos`  
--  
ALTER TABLE `processos`  
  ADD PRIMARY KEY (`Pro_Codigo`);  
  
--  
-- AUTO_INCREMENT for dumped tables  
--  
--  
-- AUTO_INCREMENT for table `arestas`  
--  
ALTER TABLE `arestas`
```

```
    MODIFY `Are_Codigo` int(11) NOT NULL AUTO_INCREMENT;
--
-- AUTO_INCREMENT for table `arestas_pontos`
--
ALTER TABLE `arestas_pontos`
    MODIFY `ArePon_Codigo` int(11) NOT NULL AUTO_INCREMENT;
--
-- AUTO_INCREMENT for table `elementos`
--
ALTER TABLE `elementos`
    MODIFY `Ele_Codigo` int(11) NOT NULL AUTO_INCREMENT;
--
-- AUTO_INCREMENT for table `processos`
--
ALTER TABLE `processos`
    MODIFY `Pro_Codigo` int(11) NOT NULL AUTO_INCREMENT;
--
-- Constraints for dumped tables
--
--
-- Limitadores para a tabela `arestas`
--
ALTER TABLE `arestas`
    ADD CONSTRAINT `fk_arestas_elementos1` FOREIGN KEY (`Are_From`)
    REFERENCES `elementos` (`Ele_Codigo`)
    ON DELETE NO ACTION ON UPDATE NO ACTION,
    ADD CONSTRAINT `fk_arestas_elementos2` FOREIGN KEY (`Are_To`)
    REFERENCES `elementos` (`Ele_Codigo`)
    ON DELETE NO ACTION ON UPDATE NO ACTION;
--
-- Limitadores para a tabela `arestas_pontos`
--
```

```
ALTER TABLE `arestas_pontos`  
  ADD CONSTRAINT `fk_arestas_pontos_arestas1`  
  FOREIGN KEY (`Are_Codigo`) REFERENCES `arestas` (`Are_Codigo`)  
  ON DELETE NO ACTION ON UPDATE NO ACTION;  
  
--  
-- Limitadores para a tabela `elementos`  
--  
ALTER TABLE `elementos`  
  ADD CONSTRAINT `fk_elementos_processos`  
  FOREIGN KEY (`Pro_Codigo`) REFERENCES `processos` (`Pro_Codigo`)  
  ON DELETE NO ACTION ON UPDATE NO ACTION;
```

APÊNDICE B — DOCUMENTOS USADOS NAS AVALIAÇÕES DE USUÁRIOS

B.1 Perfil do Usuário

Qual sua faixa etária?

- 15-18 18-25 26-30 31-40 Mais de 40

Qual seu nível de escolaridade?

- Ensino Médio Completo
 Aluno de graduação
 Ensino Superior Completo
 Mestre
 Doutor

Como você avalia seu conhecimento em bancos de dados relacionais?

- Nunca ouvi falar sobre o assunto
 Já ouvi falar sobre o assunto, mas não tenho conhecimentos básicos
 Tenho conhecimentos básicos
 Já tive contato e tenho conhecimentos razoáveis
 Tenho ou tive contato mais aprofundado e tenho bom conhecimento

Como você avalia seu conhecimento em bancos de dados orientados a grafos?

- Nunca ouvi falar sobre o assunto
 Já ouvi falar sobre o assunto, mas não tenho conhecimentos básicos
 Tenho conhecimentos básicos
 Já tive contato e tenho conhecimentos razoáveis
 Tenho ou tive contato mais aprofundado e tenho bom conhecimento

Como você avalia seu conhecimento em BPM (Business Process Model)?

- Nunca ouvi falar sobre o assunto
 Já ouvi falar sobre o assunto, mas não tenho conhecimentos básicos
 Tenho conhecimentos básicos
 Já tive contato e tenho conhecimentos razoáveis
 Tenho ou tive contato mais aprofundado e tenho bom conhecimento

B.2 Questionário SUS - System Usability Scale

Foi utilizado para todos os grupos de questões analisados. São realizadas afirmações que devem ser respondidas usando uma escala Likert as seguintes opções: “Discordo totalmente”, “Discordo”, “Neutro”, “Concordo” e “Concordo totalmente”. As questões realizadas são as listadas a seguir.

- 1- Eu acho que gostaria de usar esse método com frequência.
- 2- Eu acho o método desnecessariamente complexo.
- 3- Eu achei o método fácil de usar.
- 4- Eu acho que precisaria de ajuda de uma pessoa com conhecimentos técnicos para usar o método.
- 5- Eu acho que os modelos de processo são bem representados pelo método.
- 6- Eu acho que o método apresenta muita incoerência.
- 7- Eu imagino que as pessoas aprenderão como usar esse método rapidamente.
- 8- Eu achei o método atrapalhado de usar.
- 9- Eu me senti confiante ao usar o método.
- 10 - Eu precisei aprender várias coisas novas antes de conseguir usar o método.

B.3 Avaliação de Armazenamento de Modelo de Processo em Banco Orientado a Grafos

B.3.1 Treinamento Realizado

Um modelo de processo é uma coleção de atividades estruturadas ou tarefas que produzem um produto ou serviço específico. Sua representação se dá da seguinte maneira:

A área localizada no topo da Figura B.1 (com fundo cinza) mostra os elementos de um modelo de processo que podem ser representados no caso em específico. Tais elementos se diferenciam por cores e formatos diferentes. Abaixo da área em vermelho, temos representado determinado modelo de processo genérico.

O presente trabalho propõe um método de armazenamento de processos em bancos de dados NÃO-RELACIONAIS. O armazenamento do modelo de processo descrito acima pode ser representado conforme a Figura B.2, mostrada abaixo.

A área localizada no topo da Figura B.1 (com fundo azul) contém o nome dos elementos representados no modelo armazenado no banco não relacional. Estes elementos

Figura B.1: Representação de um modelo de processo.

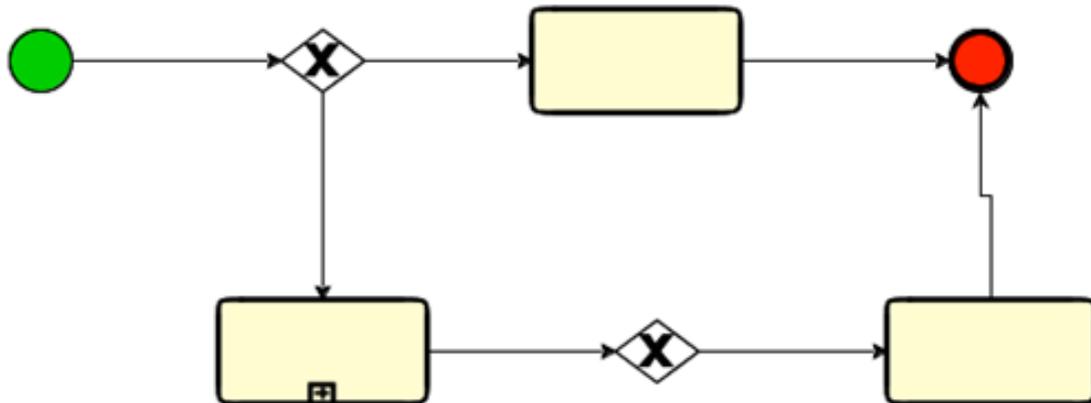
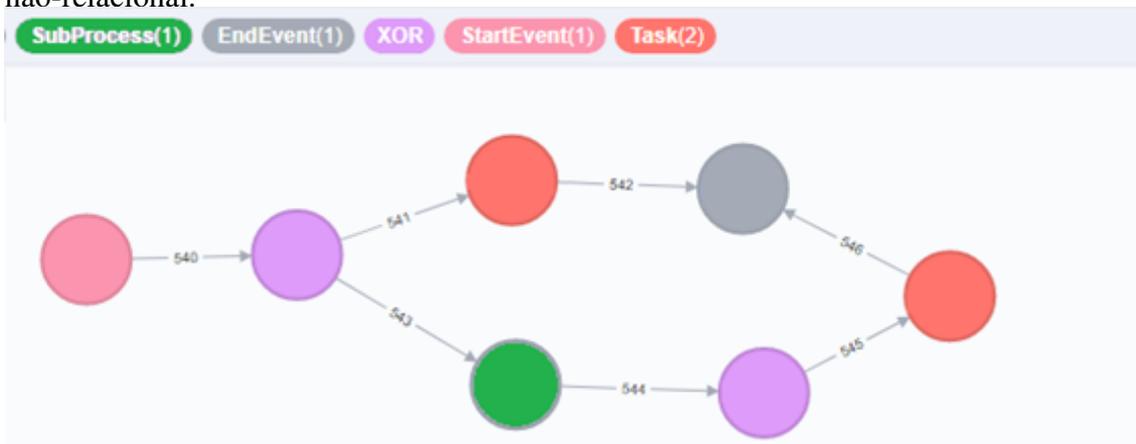


Figura B.2: Representação de um modelo de processo armazenado em banco de dados não-relacional.



possuem os mesmos nomes dos elementos do modelo de processo original, representado na Figura B.1. Além disso, sua legenda, circulada em vermelho, identifica os elementos armazenados através de suas respectivas cores.

Nos dois próximos exercícios, você deverá identificar qual o modelo de processo dentre as 4 opções fornecidas é representado pelo modelo armazenado no banco de dados não relacional que estará em questão. Para isso, você deverá compará-los tanto em termos de suas arestas (ligações entre os elementos), quanto ao tipo de elemento que participa da ligação. Tomando como exemplo o modelo armazenado demonstrado na Figura B.2, começamos analisando a primeira ligação entre os elementos, circulada em vermelho:

Ela indica a ligação de um elemento do tipo “StartEvent” (rosa) com um elemento do tipo “XOR” (roxo). Analisando a mesma ligação da Figura abaixo, notamos não se tratar de um mesmo modelo de processo, pois liga um elemento do tipo “StartEvent” com um elemento do tipo “Task”.

Figura B.3:

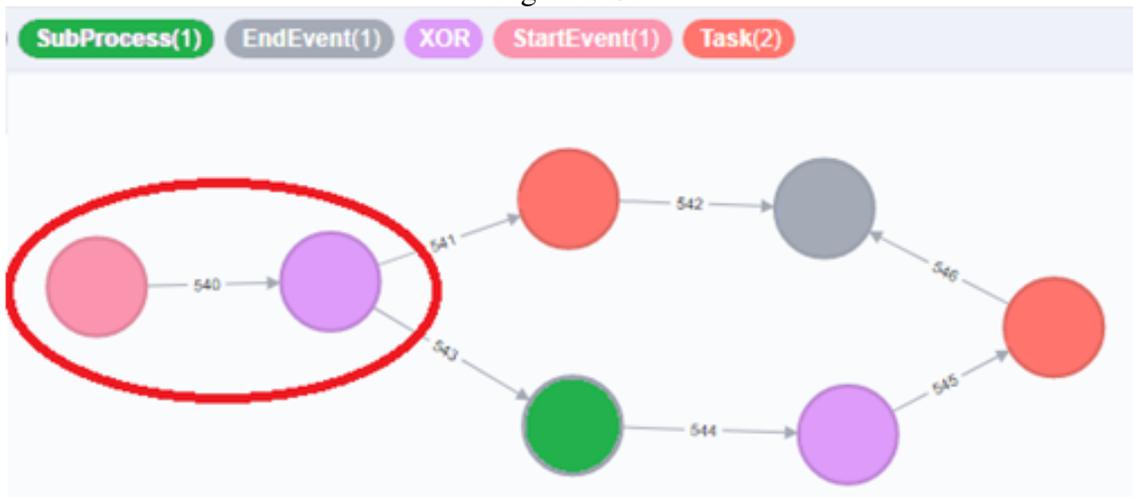
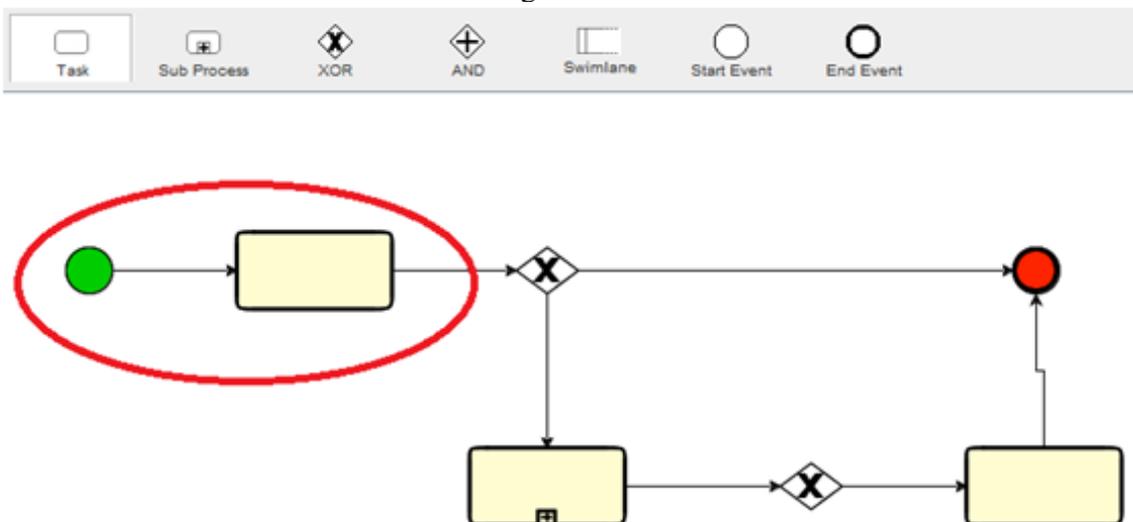


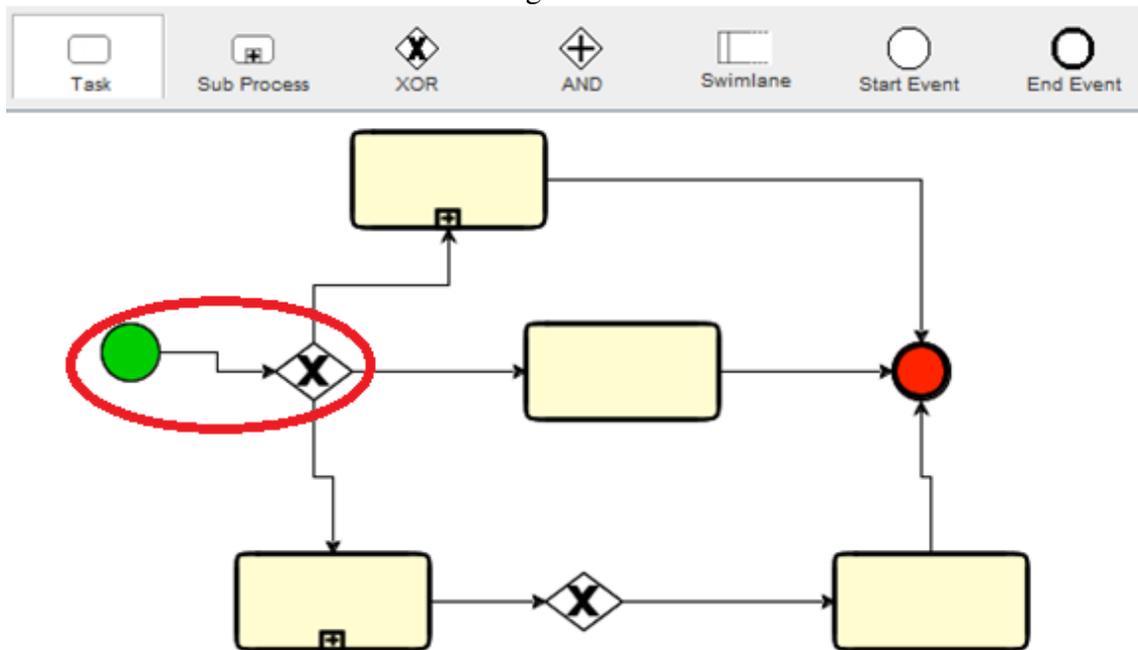
Figura B.4:



Por outro lado, a Figura B.5 mostra um modelo de processo em que a primeira ligação de elementos é idêntica à do modelo armazenado no modelo não-relacional. No círculo vermelho, vemos a ligação de um elemento do tipo “StartEvent” a um elemento do tipo “XOR”. Seguindo a análise, entretanto, podemos observar que este elemento “XOR” está ligado a outros 3 elementos, enquanto no modelo armazenado o “XOR” se liga a apenas outros 2 elementos. Logo, podemos descartar o modelo apresentado na Figura B.5 como sendo o mesmo modelo armazenado no banco de dados não relacional, da Figura B.2.

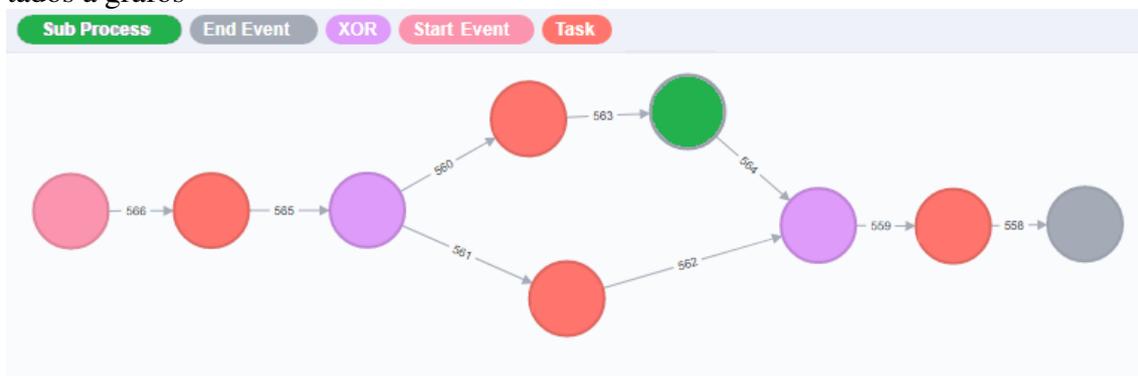
Fazendo essa sucessiva análise de elementos e ligações, percebemos que o modelo mostrado na Figura B.2 é exatamente o mesmo da Figura B.1. Ou seja, esta é a resposta que você deveria fornecer caso este exemplo fosse colocado como questão e são estes conceitos que devem ser utilizados para responder as 2 perguntas a seguir.

Figura B.5:



B.3.2 Questão 1

Figura B.6: Questão 1 do experimento sobre armazenamento em bancos de dados orientados a grafos



B.3.3 Questão 2

B.4 Avaliação de Armazenamento de Modelo de Processo em Banco Relacional

B.4.1 Treinamento Realizado

Um modelo de processo é uma coleção de atividades estruturadas ou tarefas que produzem um produto ou serviço específico. Sua representação se dá da seguinte maneira:

Figura B.7: Primeira opção da questão 1 do experimento sobre armazenamento em bancos de dados orientados a grafos

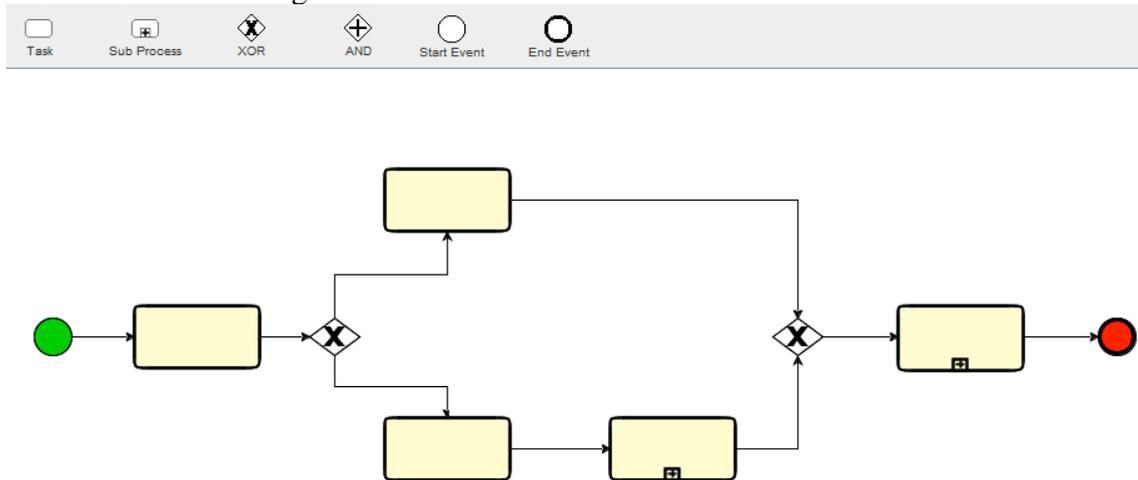
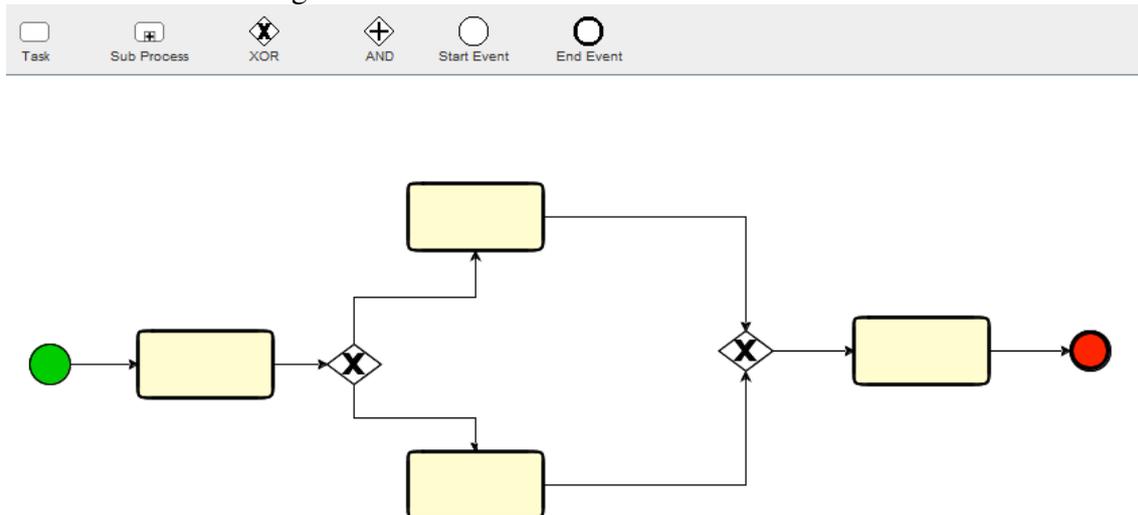


Figura B.8: Segunda opção da questão 1 do experimento sobre armazenamento em bancos de dados orientados a grafos



A área localizada no topo da Figura B.16 (com fundo cinza) mostra os elementos de um modelo de processo que podem ser representados no caso em específico. Tais elementos se diferenciam por cores e formatos diferentes. Abaixo da área em vermelho, temos representado determinado modelo de processo genérico.

O presente trabalho propõe um método de armazenamento de processos em bancos de dados RELACIONAIS. O armazenamento do modelo de processo descrito acima pode ser representado conforme a Figura B.17, mostrada abaixo.

A tabela “elementos” da Figura B.17 contém o nome dos elementos representados no modelo armazenado no banco relacional. Estes elementos possuem os mesmos nomes dos elementos do modelo de processo original, representado na Figura B.16. A tabela

Figura B.9: Terceira opção da questão 1 do experimento sobre armazenamento em bancos de dados orientados a grafos

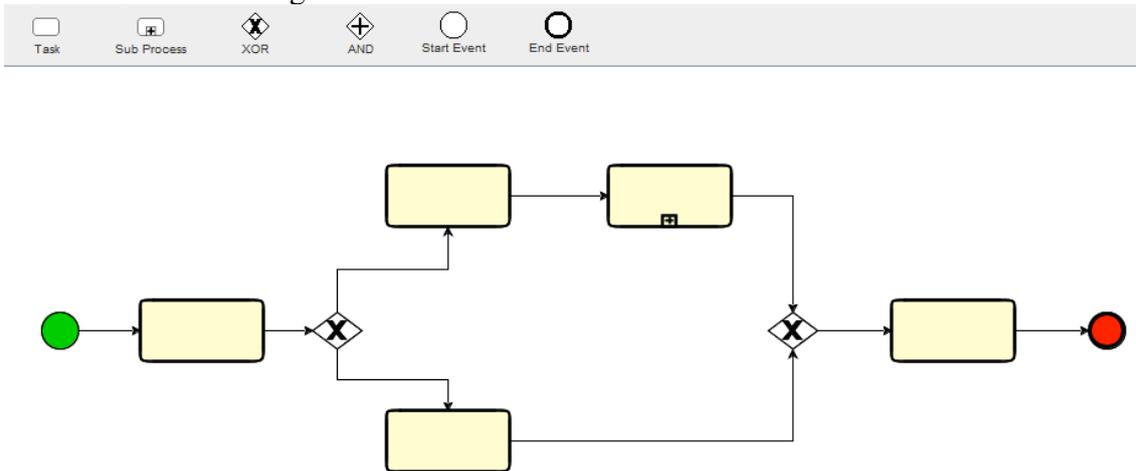
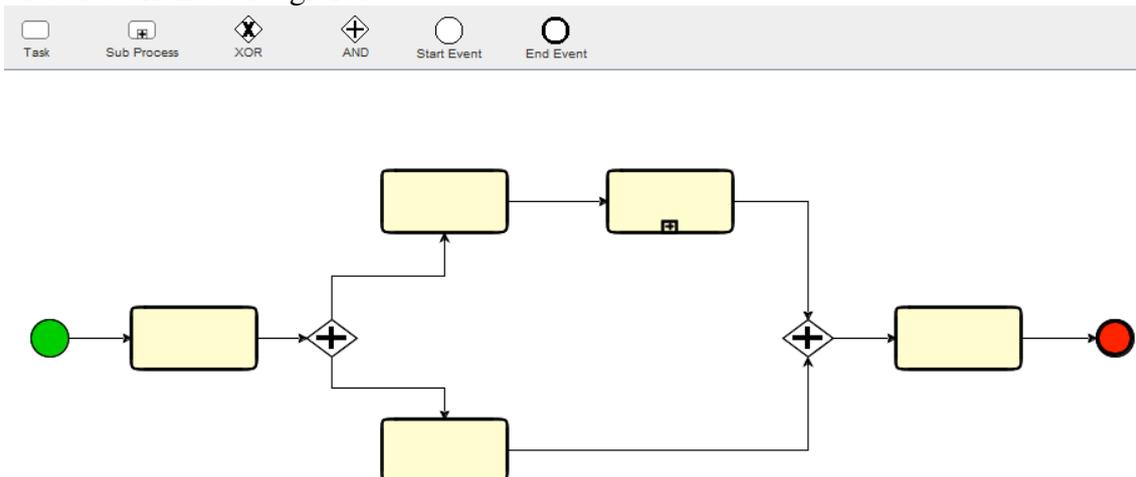


Figura B.10: Quarta opção da questão 1 do experimento sobre armazenamento em bancos de dados orientados a grafos



“arestas” representa a ligação entre os elementos. A coluna “Are_From” indica de onde a aresta parte, enquanto a coluna “Are_To” indica o destino dela.

Nos dois próximos exercícios, você deverá identificar qual o modelo de processo dentre as 4 opções fornecidas é representado pelo modelo armazenado no banco de dados relacional que estará em questão. Para isso, você deverá compará-los tanto em termos de suas arestas (ligações entre os elementos), quanto ao tipo de elemento que participa da ligação. Tomando como exemplo o modelo armazenado demonstrado na Figura B.17, começamos analisando a primeira ligação entre os elementos, circulada em vermelho:

Ela indica a ligação de um elemento do tipo “StartEvent” (Ele_Codigo número 1) com um elemento do tipo “XOR” (Ele_Codigo número 2). Analisando a mesma ligação da Figura abaixo, notamos não se tratar de um mesmo modelo de processo, pois liga um elemento do tipo “StartEvent” com um elemento do tipo “Task”.

Figura B.11: Questão 2 do experimento sobre armazenamento em bancos de dados orientados a grafos

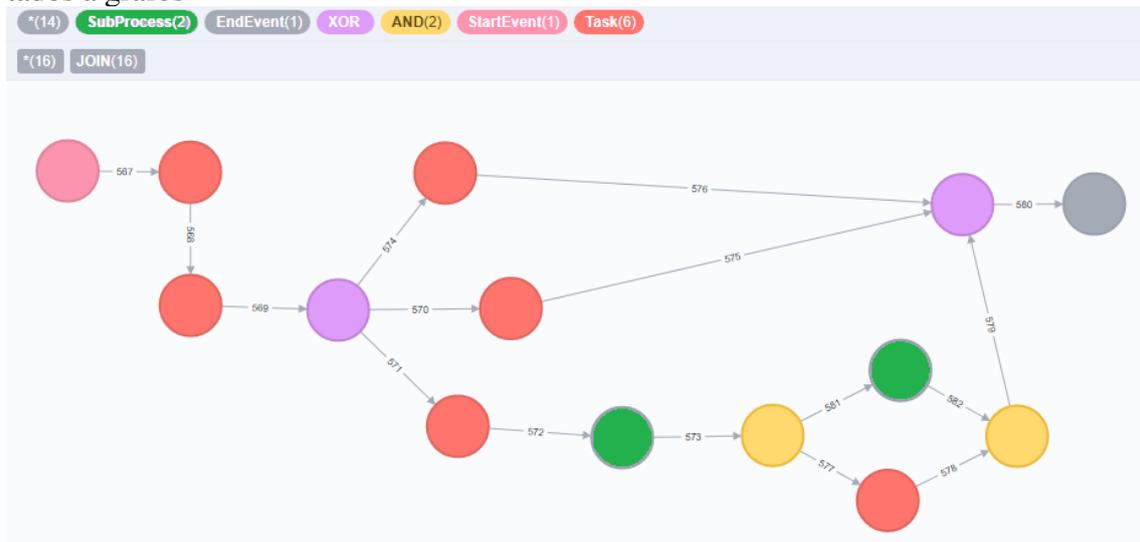
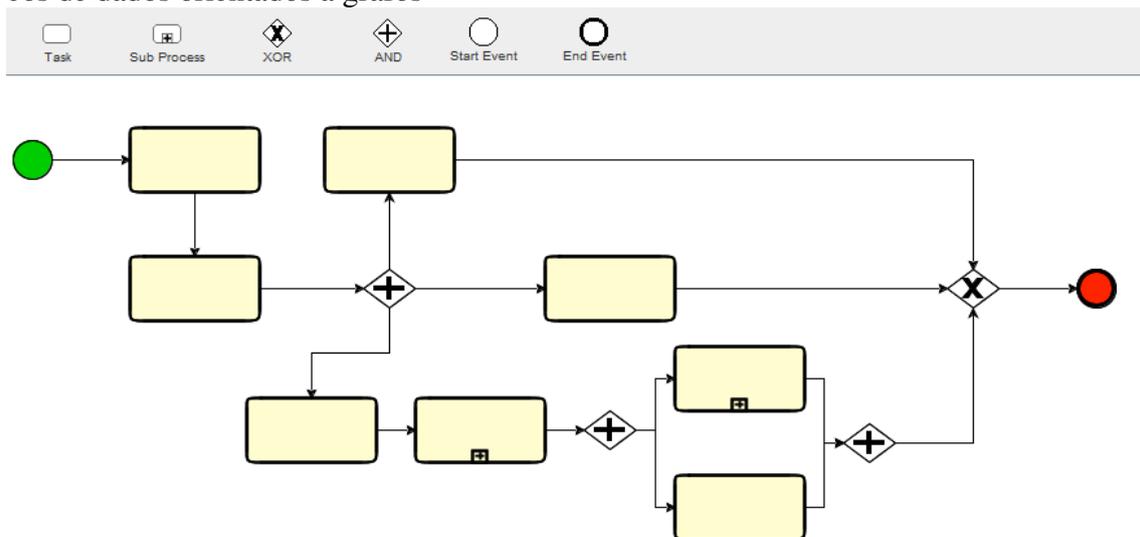


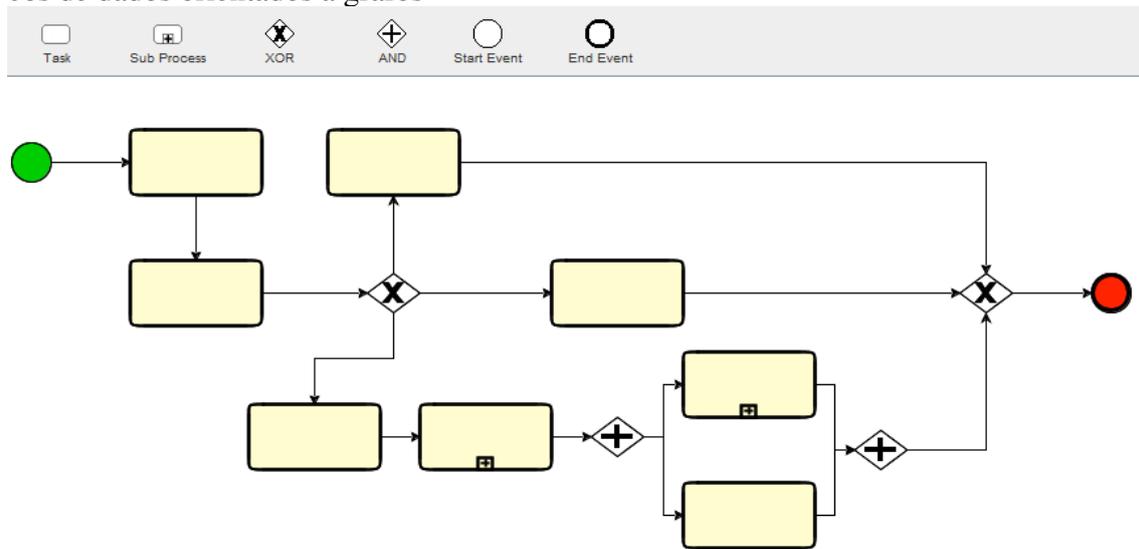
Figura B.12: Primeira opção da questão 2 do experimento sobre armazenamento em bancos de dados orientados a grafos



Por outro lado, a Figura B.20 mostra um modelo de processo em que a primeira ligação de elementos é idêntica à do modelo armazenado no modelo relacional. Inicialmente, vemos a ligação de um elemento do tipo “StartEvent” a um elemento do tipo “XOR”. Seguindo a análise, entretanto, podemos observar que este elemento “XOR” está ligado a outros 3 elementos, enquanto no modelo armazenado o “XOR” se liga a apenas outros 2 elementos, como demonstrado na Figura B.21. Logo, podemos descartar o modelo apresentado na Figura B.20 como sendo o mesmo modelo armazenado no banco de dados relacional, da Figura B.17.

Fazendo essa sucessiva análise de elementos e ligações, percebemos que o modelo mostrado na Figura B.17 é exatamente o mesmo da Figura B.16. Ou seja, esta é a resposta

Figura B.13: Segunda opção da questão 2 do experimento sobre armazenamento em bancos de dados orientados a grafos



que você deveria fornecer caso este exemplo fosse colocado como questão e são estes conceitos que devem ser utilizados para responder as 2 perguntas a seguir.

B.4.2 Questão 1

B.4.3 Questão 2

B.5 Consulta de Modelo de Processo em Banco de Dados Orientado a Grafos

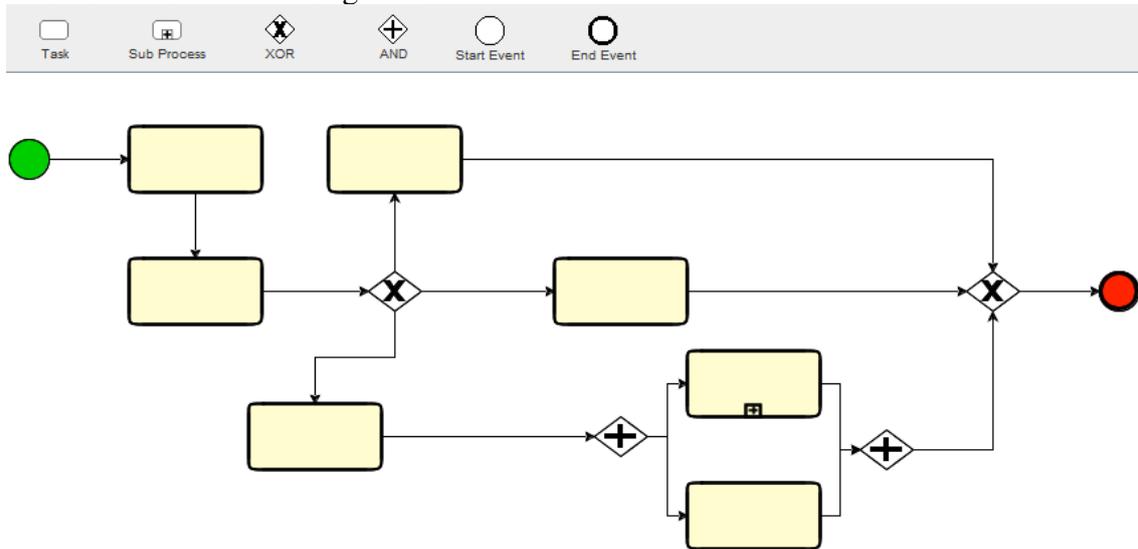
B.5.1 Treinamento Realizado

Um modelo de processo é uma coleção de atividades estruturadas ou tarefas que produzem um produto ou serviço específico. Sua representação se dá da seguinte maneira:

A área localizada no topo da Figura B.32 (com fundo cinza) mostra os elementos de um modelo de processo que podem ser representados no caso em específico. Tais elementos se diferenciam por cores e formatos diferentes. Abaixo da área em vermelho, temos representado determinado modelo de processo genérico.

O presente trabalho propõe um método de consulta de fragmentos de processos em bancos de dados NÃO-RELACIONAIS. Fragmentos são pedaços de processos contidos dentro de um processo maior. Um exemplo de fragmento de processo está descrito na Figura B.33. Nele, podemos ver a ligação entre um elemento “XOR”, uma “Task” e um

Figura B.14: Terceira opção da questão 2 do experimento sobre armazenamento em bancos de dados orientados a grafos

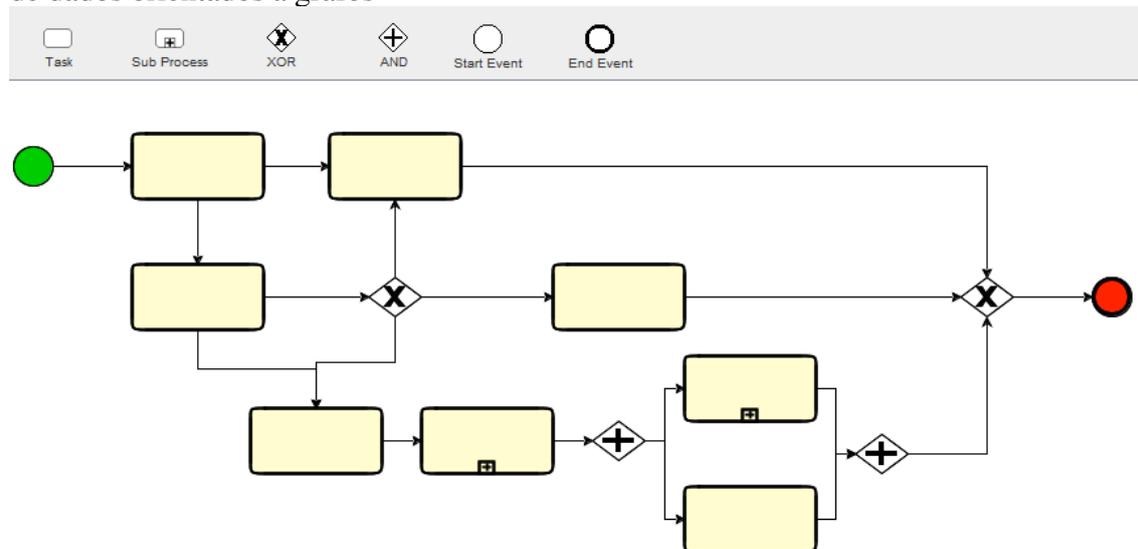


“End-Event”. Trata-se de um fragmento do modelo de processo mostrado na Figura B.32, pois está contido nele, conforme mostra o círculo vermelho da Figura B.34.

A consulta mostrada na Figura B.35 representa o fragmento de processo destacado anteriormente. As linhas relevantes para o experimento são as que vem antes do texto “WITH”, grifado em azul (neste caso, linha 3). Cada linha representa a ligação entre os dois elementos. Cada elemento, por sua vez, recebe um identificador único, representado por uma letra do alfabeto. No caso retratado, na Figura B.35, o elemento “a” é do tipo “XOR”, enquanto o elemento “b” é do tipo “Task” e o elemento “c” é do tipo “EndEvent”. A linha 1, na qual pode ser visto “(a:XOR)-[:JOIN]->(b:Task)” indica a existência de uma aresta com origem em um elemento identificado por “c” do tipo “EndEvent”. Esse cenário é exatamente equivalente ao fragmento de processo indicado na Figura B.33.

Nos dois próximos exercícios, você deverá identificar qual o modelo de processo dentre as 4 opções fornecidas contém o fragmento de processo que estará em questão. Para isso, você deverá compará-los tanto em termos de suas arestas (ligações entre os elementos), quanto ao tipo de elemento que participa da ligação.

Figura B.15: Quarta opção da questão 2 do experimento sobre armazenamento em bancos de dados orientados a grafos



B.5.2 Questão 1

B.5.3 Questão 2

B.6 Consulta de Modelo de Processo em Banco de Dados Relacional

B.6.1 Treinamento Realizado

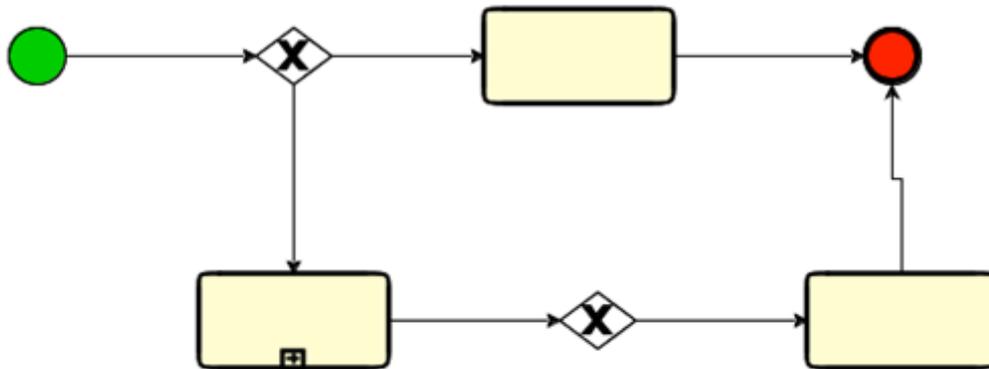
Um modelo de processo é uma coleção de atividades estruturadas ou tarefas que produzem um produto ou serviço específico. Sua representação se dá da seguinte maneira:

A área localizada no topo da Figura B.46 (com fundo cinza) mostra os elementos de um modelo de processo que podem ser representados no caso em específico. Tais elementos se diferenciam por cores e formatos diferentes. Abaixo da área em cinza, temos representado determinado modelo de processo genérico.

O presente trabalho propõe um método de consulta de fragmentos de processos em bancos de dados RELACIONAIS. Fragmentos são pedaços de processos contidos dentro de um processo maior. Um exemplo de fragmento de processo está descrito na Figura B.47. Nele, podemos ver a ligação entre um elemento “XOR”, uma “Task” e um “End-Event”. Trata-se de um fragmento do modelo de processo mostrado na Figura B.46, pois está contido nele, conforme mostra o círculo vermelho da Figura B.48.

A consulta mostrada na Figura B.49 representa o fragmento de processo destacado

Figura B.16: Representação de um modelo de processo.



anteriormente. As linhas relevantes para o experimento são as que vem antes do texto “AND”, grifado em roxo (neste caso, linha 4). Cada elemento recebe um identificador único, representado por uma letra do alfabeto (“e” para elementos e “a” para arestas) e um número. As linhas que contém “Ele_Style” são responsáveis por definir o tipo de cada elemento. No exemplo da Figura B.49, a linha 4, na qual se observa “AND e1.Ele_Style = 'XOR'” indica que o elemento identificado por “e1” possui o tipo “XOR”.

Por outro lado, as demais linhas servem para associar a origem ou o destino de uma aresta a determinado elemento. No caso da Figura B.49, vemos na linha 7, onde se pode observar “AND e1.Ele_Codigo = a1.Are_From” que o elemento “e1” (como já visto, do tipo “XOR”), possui uma aresta “a1” com origem nele. O destino da aresta “a1” é visto na linha 8, onde ela está associada ao elemento “e2” (como já visto, do tipo “Task”) através do “Are_To”. Estas duas linhas indicam a ligação entre um “XOR” e um “Task”. De forma semelhante, as linhas 9 e 10 indicam a origem de uma aresta “a2” do elemento “e2” (“Task”) ao elemento “e3” (“EndEvent”). Este cenário é exatamente o do fragmento buscado na Figura B.47.

Nos dois próximos exercícios, você deverá identificar qual o modelo de processo dentre as 4 opções fornecidas contém o fragmento de processo que estará em questão. Para isso, você deverá compará-los tanto em termos de suas arestas (ligações entre os elementos), quanto ao tipo de elemento que participa da ligação.

Figura B.17: Representação de um modelo de processo armazenado em banco de dados relacional.

Tabela "elementos"

Ele_Codigo	Ele_Style
1	Start-Event
2	XOR
3	Task
4	End-Event
5	XOR
6	Sub-Process
7	Task

Tabela "arestas"

Are_From	Are_To
1	2
2	3
3	4
2	6
6	5
5	7
7	4

B.6.2 Questão 1

B.6.3 Questão 2

Figura B.18:

Tabela "elementos"

Ele_Codigo	Ele_Style
1	Start-Event
2	XOR
3	Task
4	End-Event
5	XOR
6	Sub-Process
7	Task

Tabela "arestas"

Are_From	Are_To
1	2
2	3
3	4
2	6
6	5
5	7
7	4

Figura B.19:

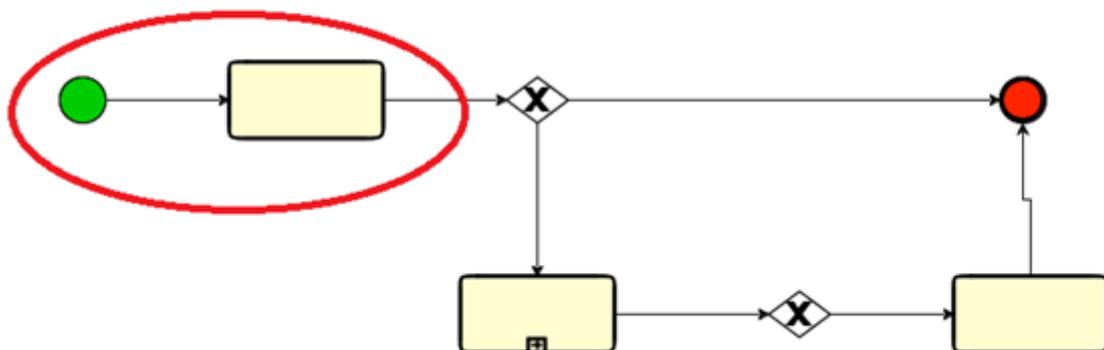


Figura B.20:

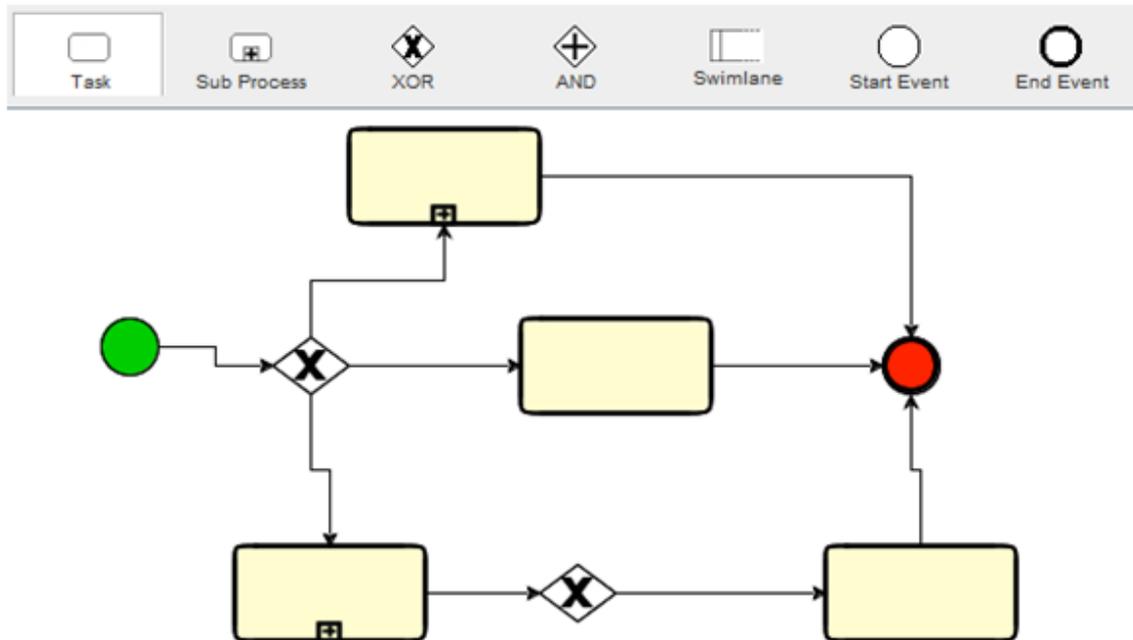


Figura B.21:

Tabela "elementos"

Ele_Codigo	Ele_Style
1	Start-Event
2	XOR
3	Task
4	End-Event
5	XOR
6	Sub-Process
7	Task

Tabela "arestas"

Are_From	Are_To
1	2
2	3
3	4
2	6
6	5
5	7
7	4

Figura B.22: Questão 1 do experimento sobre armazenamento em bancos de dados relacionais

Tabela "Elementos"		Tabela "Arestas"	
Ele_Codigo	Ele_Style	Are_From	Are_To
10	Start Event	14	15
11	Task	10	11
12	Task	13	18
13	Task	16	13
14	Task	16	12
15	End Event	11	16
16	XOR	12	17
17	XOR	18	17
18	Sub Process	17	14

Figura B.23: Primeira opção da questão 1 do experimento sobre armazenamento em bancos de dados relacionais

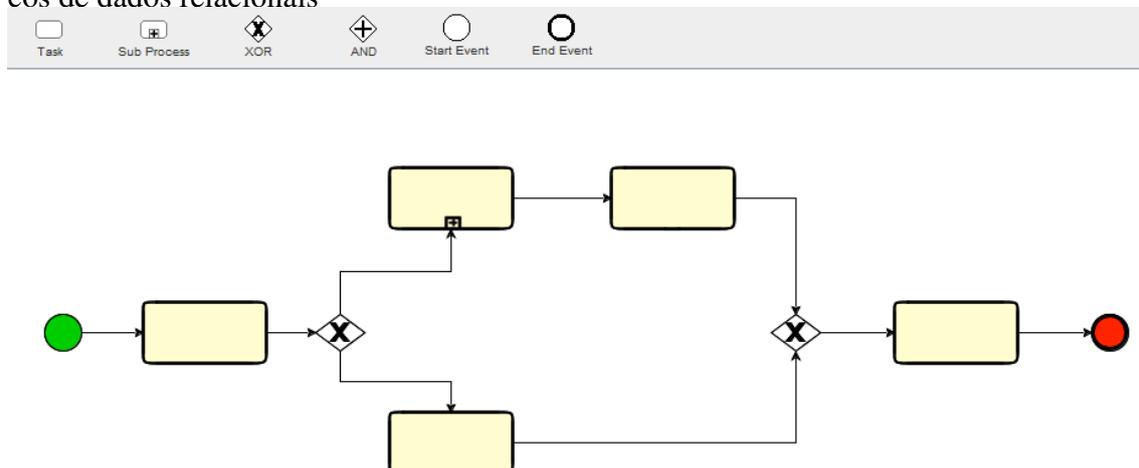


Figura B.24: Segunda opção da questão 1 do experimento sobre armazenamento em bancos de dados relacionais

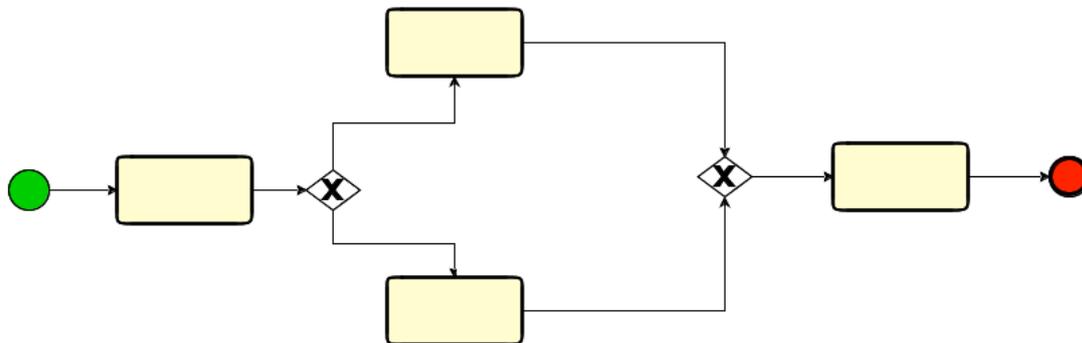


Figura B.25: Terceira opção da questão 1 do experimento sobre armazenamento em bancos de dados relacionais

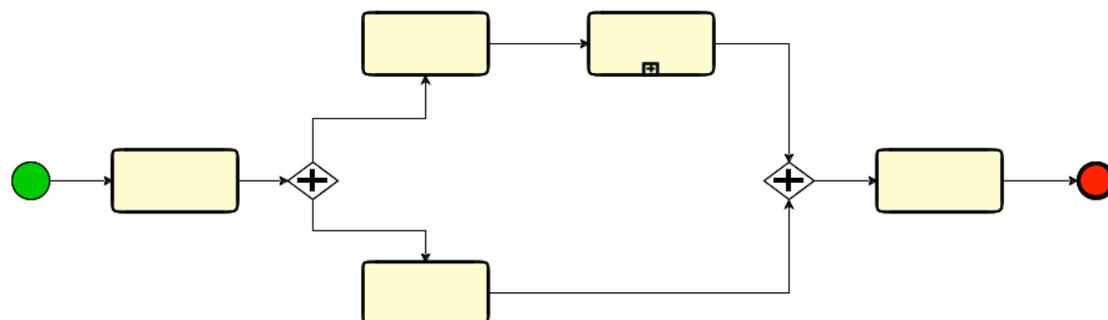


Figura B.26: Quarta opção da questão 1 do experimento sobre armazenamento em bancos de dados relacionais

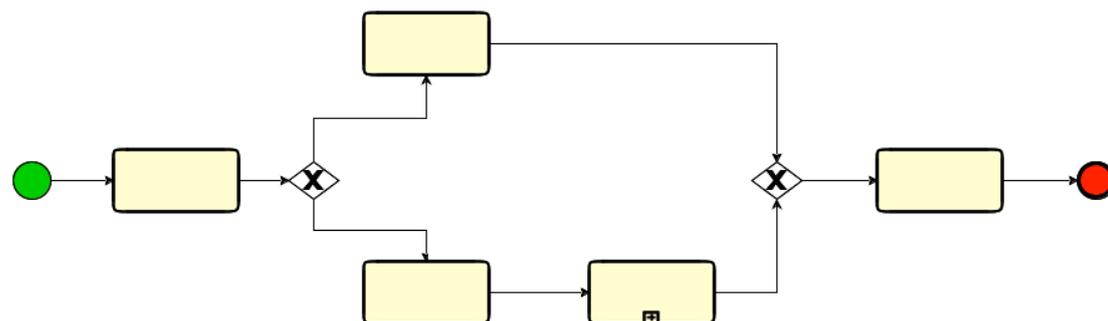


Figura B.27: Questão 2 do experimento sobre armazenamento em bancos de dados relacionais

Tabela "Elementos"		Tabela "Arestas"	
Ele_Codigo	Ele_Style	Are_From	Are_To
46	Start Event	46	47
47	Task	47	49
48	Task	48	52
49	Task	49	50
50	XOR	49	54
51	Task	50	48
52	XOR	50	54
53	End Event	50	51
54	Task	51	52
55	Sub Process	52	53
56	AND	54	55
57	AND	55	56
58	Task	56	58
59	Sub Process	56	59
		57	52
		58	57
		59	57

Figura B.30: Terceira opção da questão 2 do experimento sobre armazenamento em bancos de dados relacionais

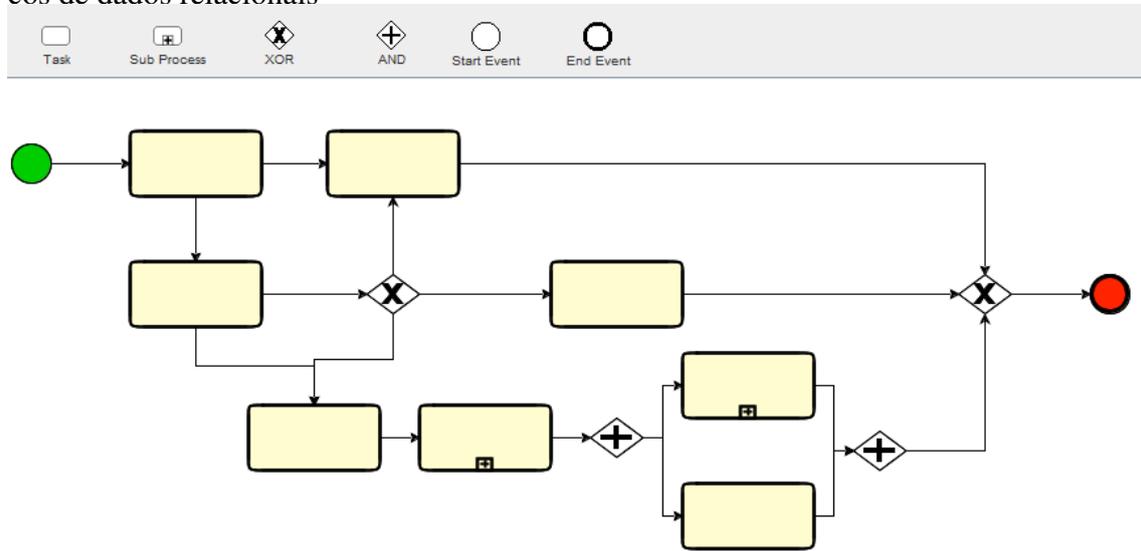


Figura B.31: Quarta opção da questão 2 do experimento sobre armazenamento em bancos de dados relacionais

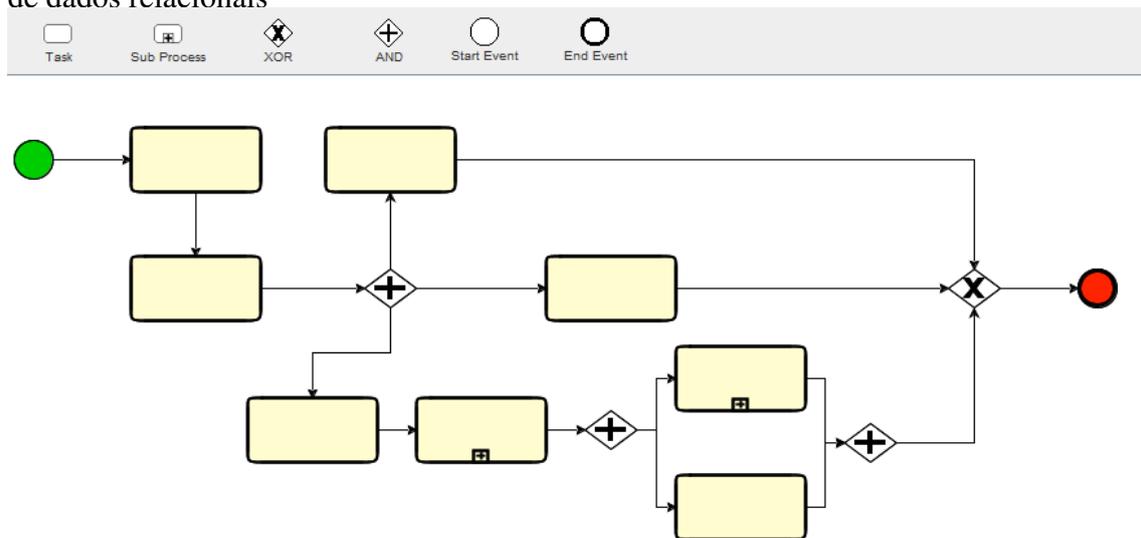


Figura B.32: Representação de um modelo de processo.

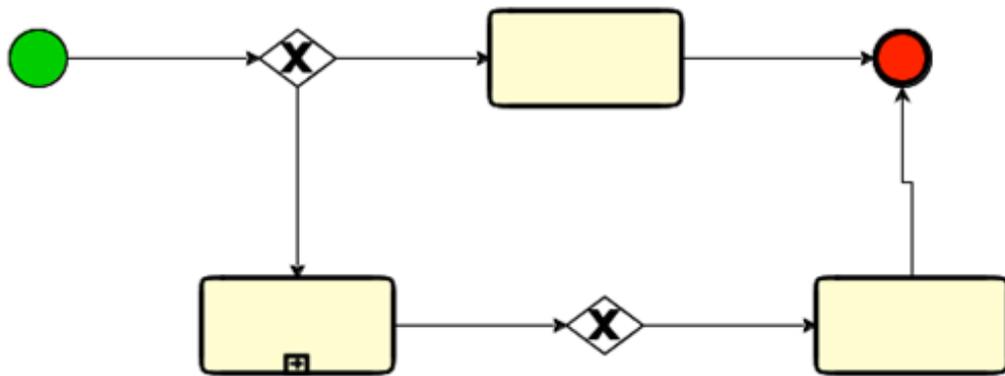


Figura B.33:

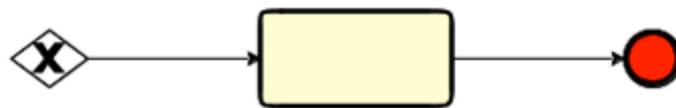


Figura B.34:

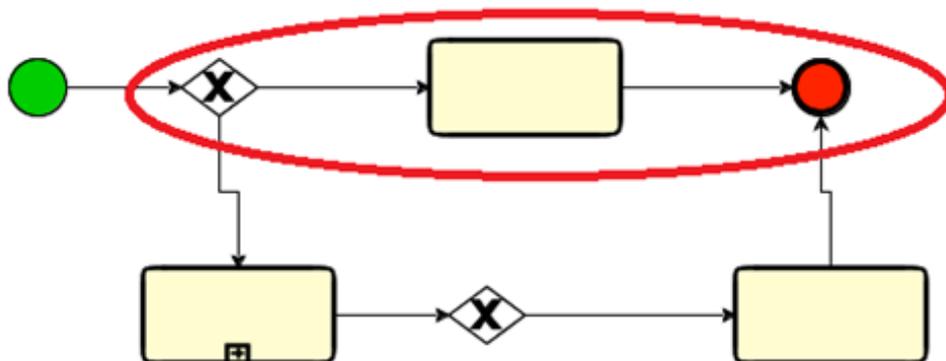


Figura B.35:

```

1 MATCH (a:XOR) -[:JOIN] -> (b:Task) ,
2 (b:Task) -[:JOIN] -> (c:EndEvent)
3 WITH DISTINCT a.graphId as graphId
4 RETURN graphId

```

Figura B.36: Questão 1 do experimento sobre consulta por fragmentos em bancos de dados orientados a grafos

```

1 MATCH (a:XOR) -[:JOIN] -> (b:SubProcess) ,
2 (b:SubProcess) -[:JOIN] -> (c:Task)
3 WITH DISTINCT a.graphId as graphId RETURN graphId

```

Figura B.37: Primeira opção da questão 1 do experimento sobre consulta por fragmentos em bancos de dados orientados a grafos

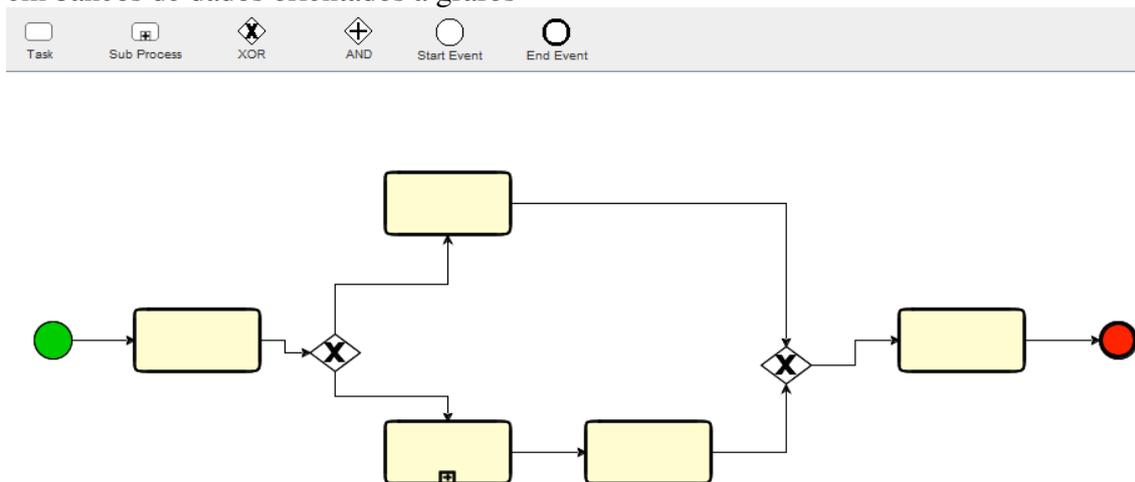


Figura B.38: Segunda opção da questão 1 do experimento sobre consulta por fragmentos em bancos de dados orientados a grafos

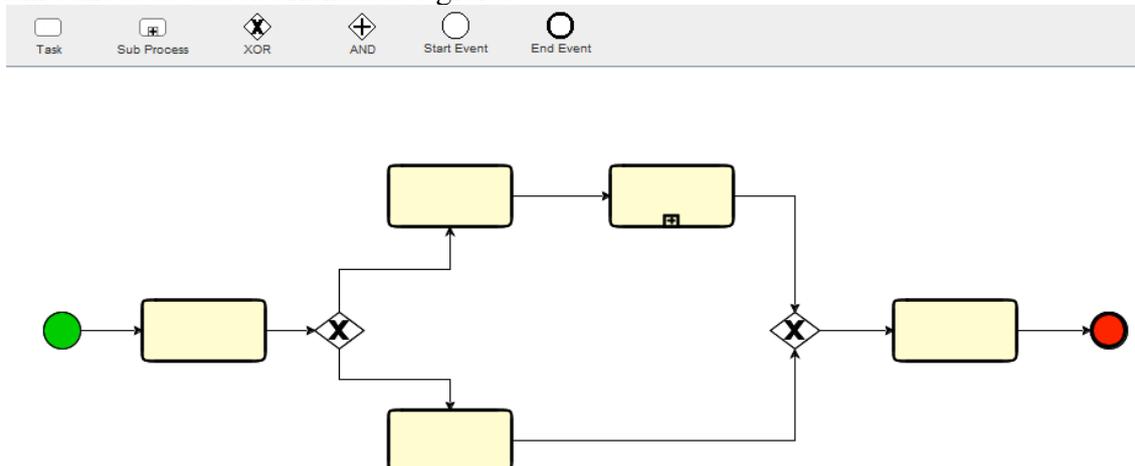


Figura B.39: Terceira opção da questão 1 do experimento sobre consulta por fragmentos em bancos de dados orientados a grafos

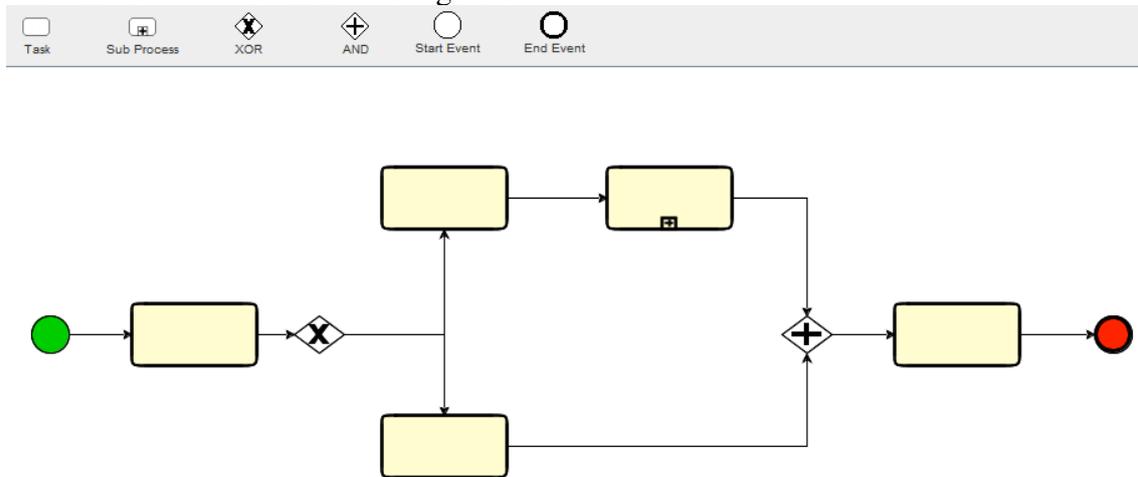


Figura B.40: Quarta opção da questão 1 do experimento sobre consulta por fragmentos em bancos de dados orientados a grafos

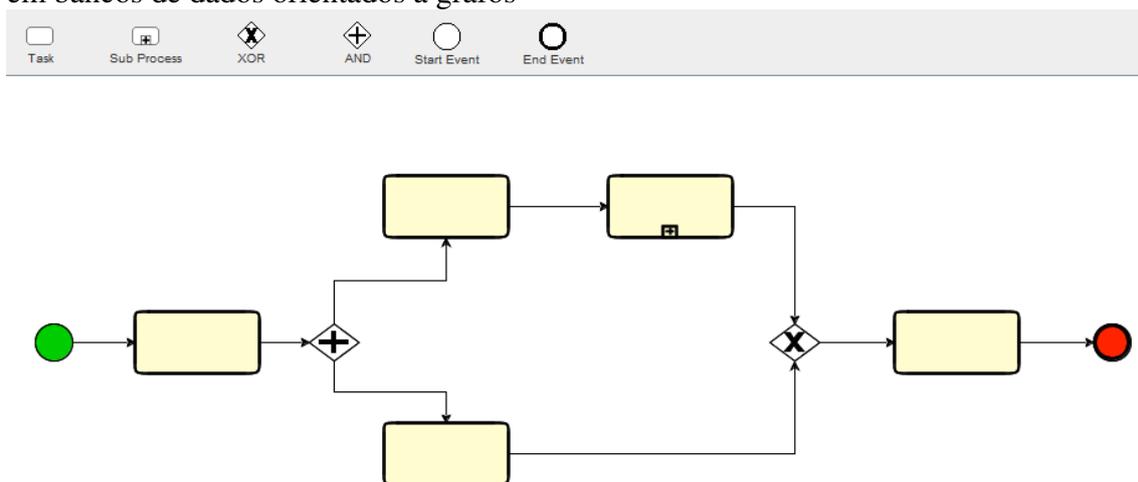


Figura B.41: Questão 2 do experimento sobre consulta por fragmentos em bancos de dados orientados a grafos

```

1 MATCH (a:Task) -[:JOIN]->(b:SubProcess),
2 (b:SubProcess) -[:JOIN]->(c:AND),
3 (c:AND) -[:JOIN]->(d:Task),
4 (c:AND) -[:JOIN]->(e:SubProcess)
5 WITH DISTINCT a.graphId as graphId
6 RETURN graphId

```

Figura B.42: Primeira opção da questão 2 do experimento sobre consulta por fragmentos em bancos de dados orientados a grafos

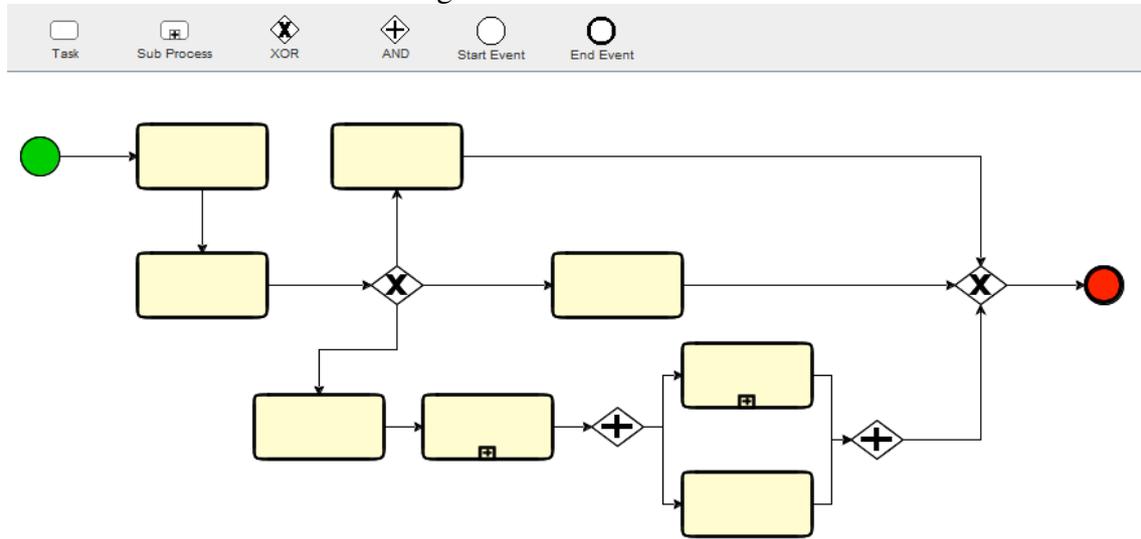


Figura B.43: Segunda opção da questão 2 do experimento sobre consulta por fragmentos em bancos de dados orientados a grafos

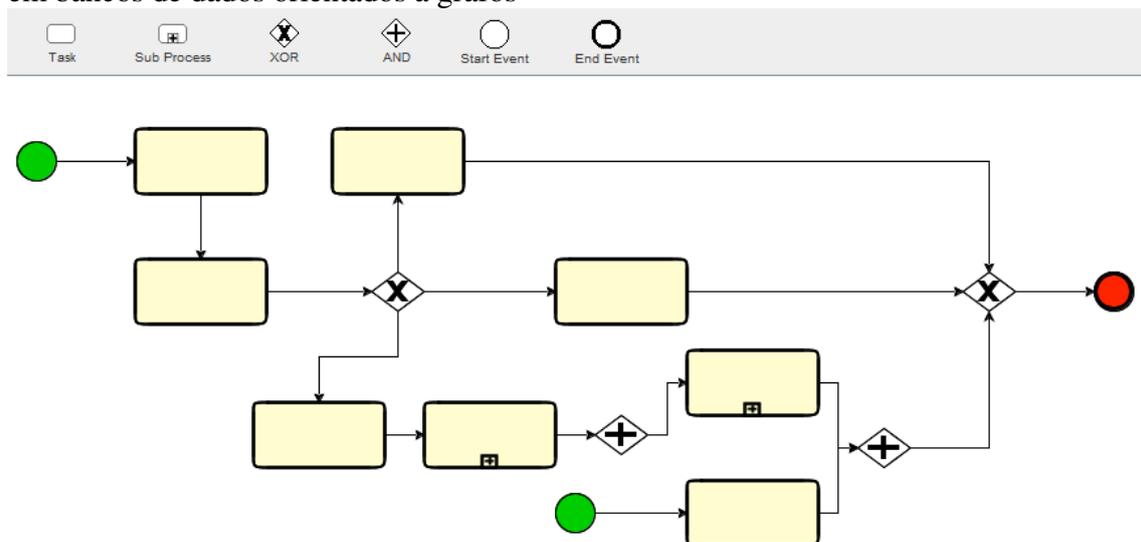


Figura B.44: Terceira opção da questão 2 do experimento sobre consulta por fragmentos em bancos de dados orientados a grafos

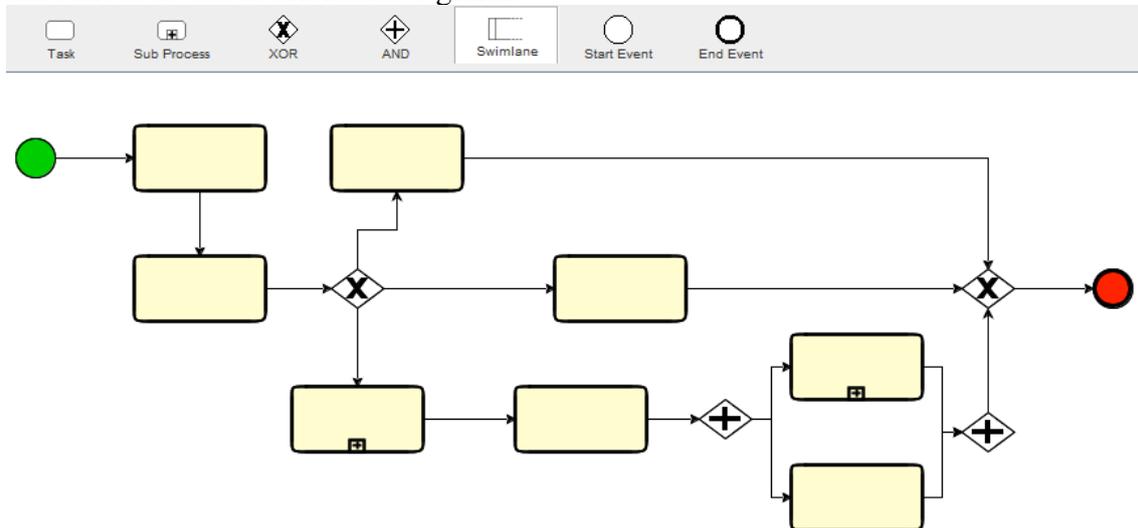


Figura B.45: Quarta opção da questão 2 do experimento sobre consulta por fragmentos em bancos de dados orientados a grafos

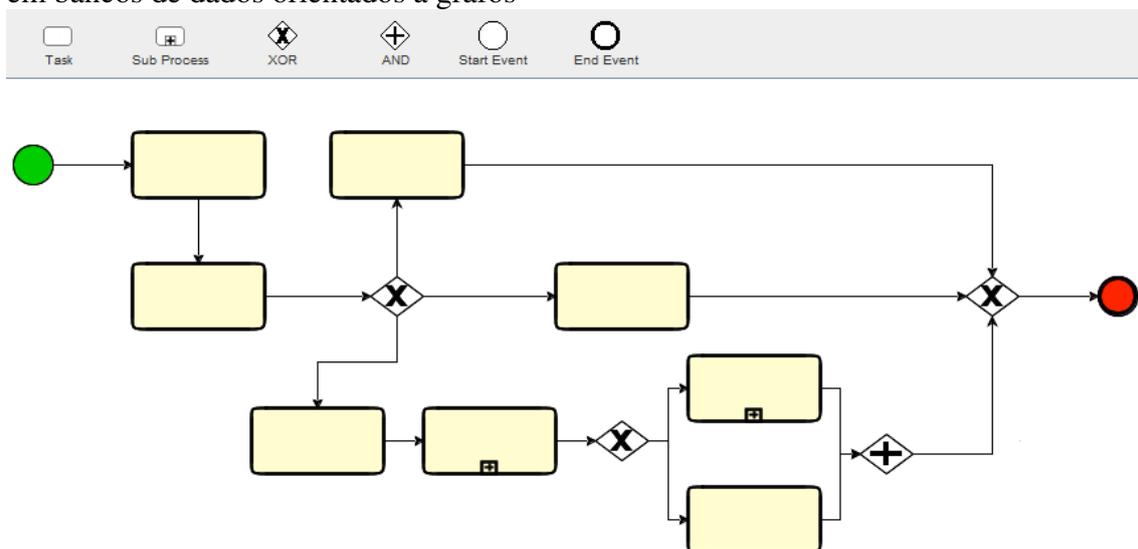


Figura B.46: Representação de um modelo de processo.

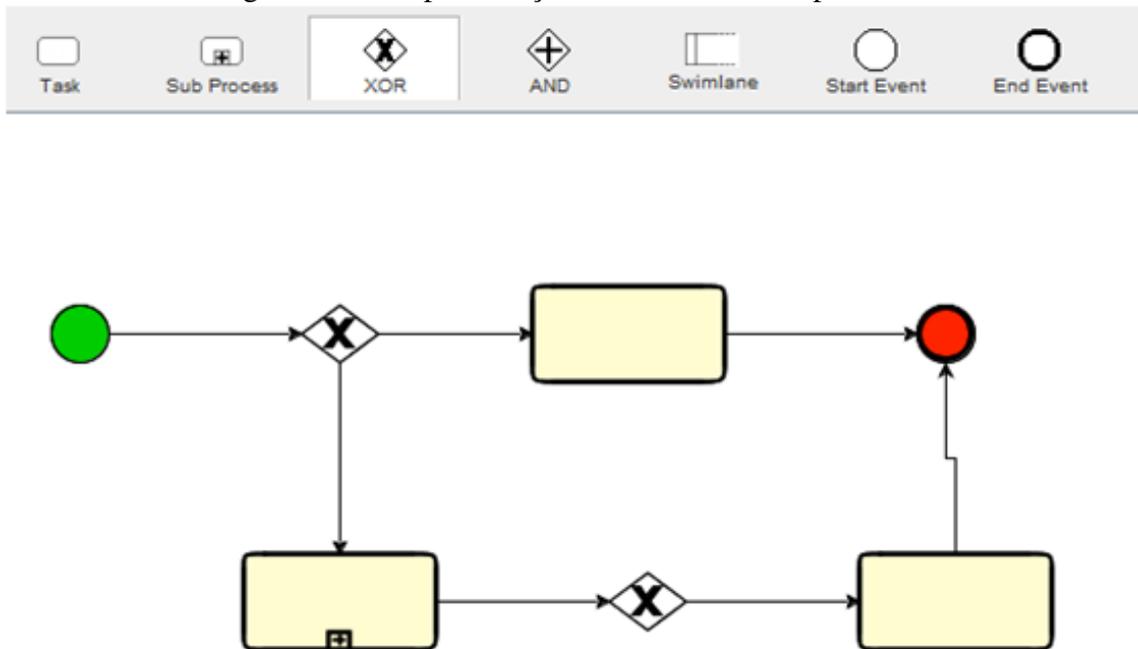


Figura B.47:

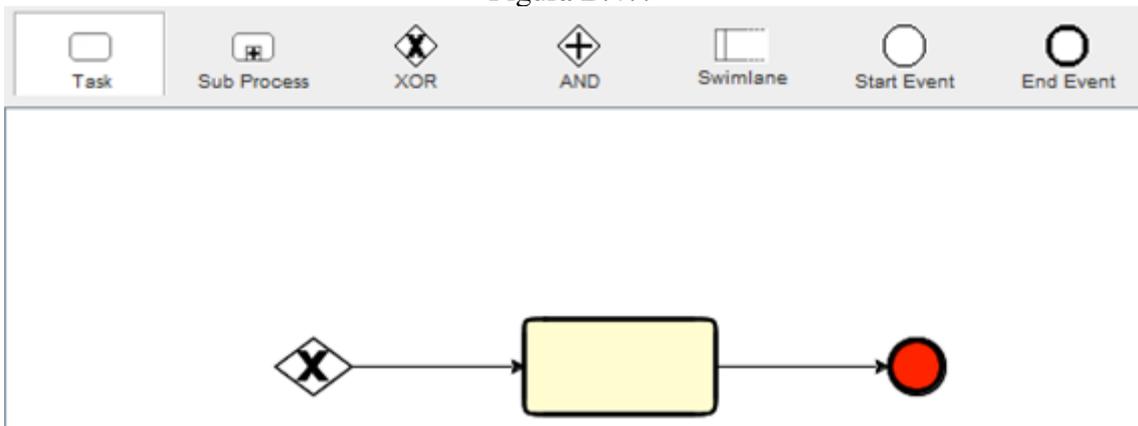


Figura B.48:

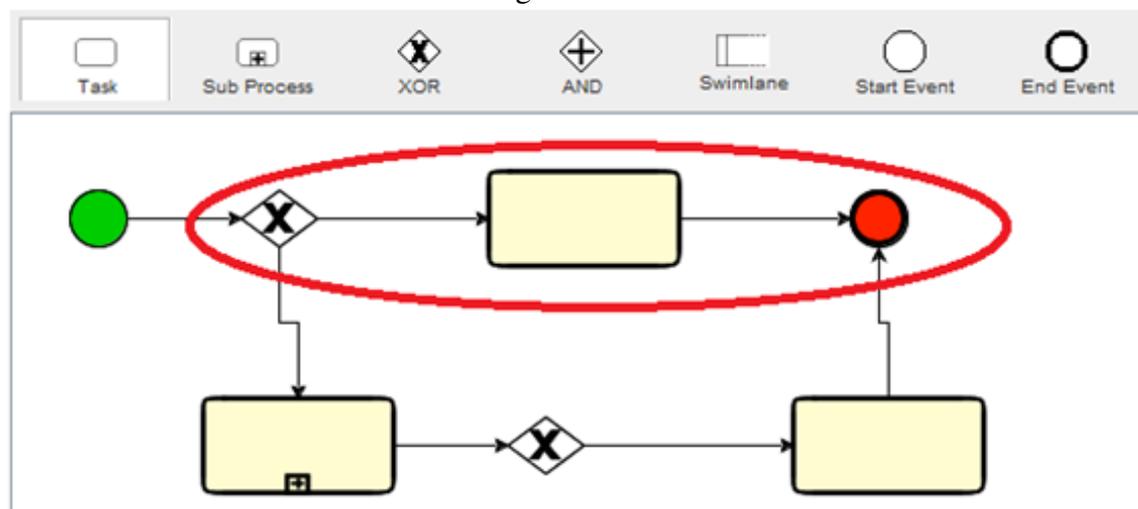


Figura B.49:

```

1 SELECT DISTINCT(e1.Pro_Codigo) AS Pro_Codigo
2 FROM elementos e1, elementos e2, elementos e3, arestas a1, arestas a2
3 WHERE 1
4 AND e1.Ele_Style = 'XOR'
5 AND e2.Ele_Style = 'Task'
6 AND e3.Ele_Style = 'End-Event'
7 AND e1.Ele_Codigo = a1.Are_From
8 AND e2.Ele_Codigo = a1.Are_To
9 AND e2.Ele_Codigo = a2.Are_From
10 AND e3.Ele_Codigo = a2.Are_To

```

Figura B.50: Questão 1 do experimento sobre consulta por fragmentos em bancos de dados relacionais

```

1 SELECT DISTINCT(e1.Pro_Codigo) AS Pro_Codigo
2 FROM elementos e1, elementos e2, elementos e3, arestas a1,
   arestas a2
3 WHERE 1
4 AND e1.Ele_Style = 'SubProcess'
5 AND e2.Ele_Style = 'AND'
6 AND e3.Ele_Style = 'Task'
7 AND e1.Ele_Codigo = a1.Are_From
8 AND e2.Ele_Codigo = a1.Are_To
9 AND e2.Ele_Codigo = a2.Are_From
10 AND e3.Ele_Codigo = a2.Are_To

```

Figura B.51: Primeira opção da questão 1 do experimento sobre consulta por fragmentos em bancos de dados relacionais

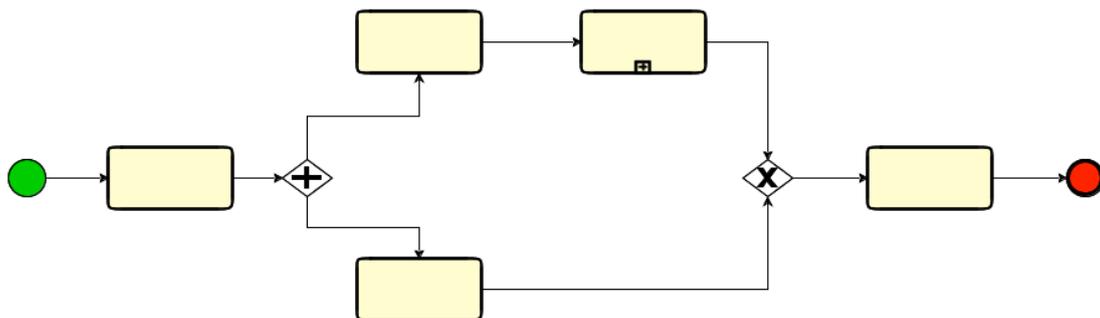


Figura B.52: Segunda opção da questão 1 do experimento sobre consulta por fragmentos em bancos de dados relacionais

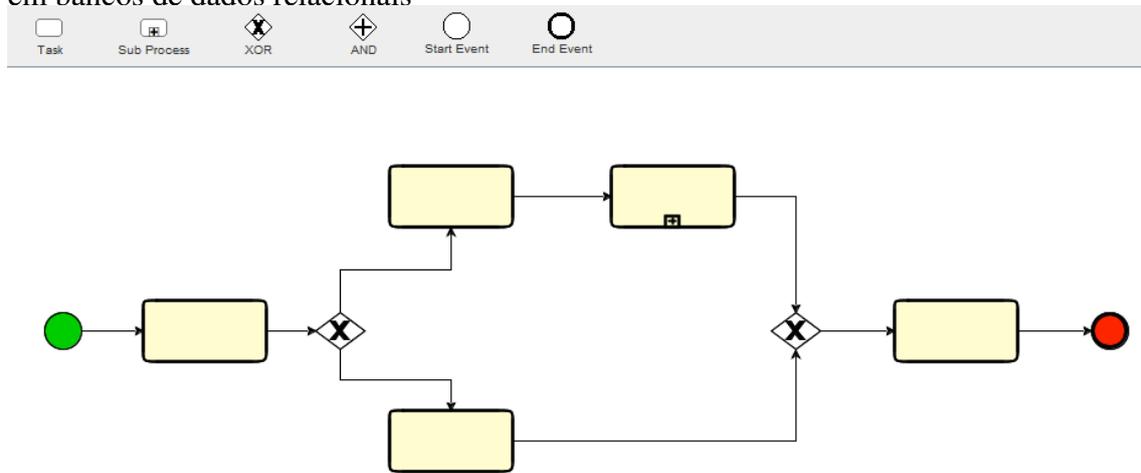


Figura B.53: Terceira opção da questão 1 do experimento sobre consulta por fragmentos em bancos de dados relacionais

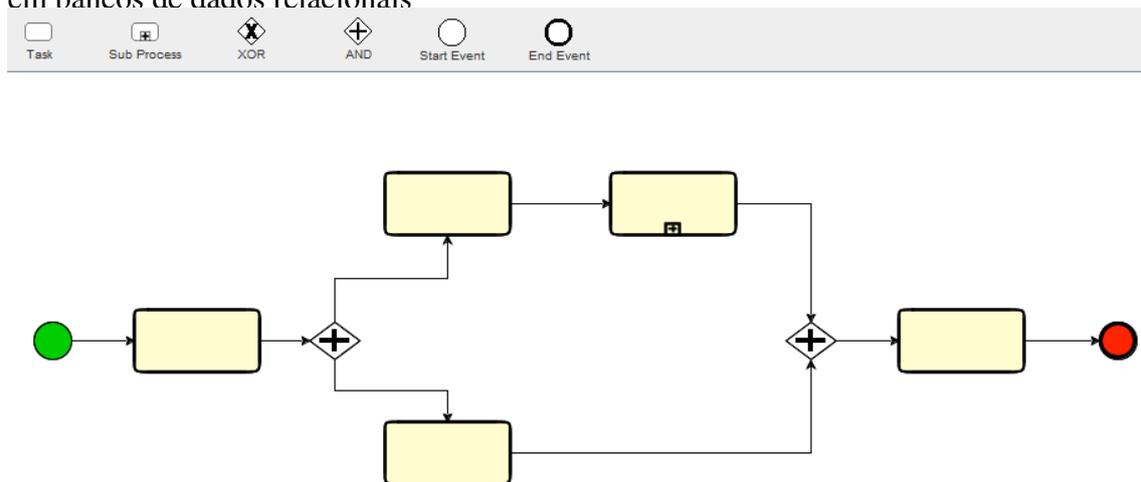


Figura B.54: Quarta opção da questão 1 do experimento sobre consulta por fragmentos em bancos de dados relacionais

