



MESTRADO EM SISTEMAS E COMPUTAÇÃO

HENDERSON ACOSTA BRAGANÇA

**XBRL PROCESSOR: UMA FERRAMENTA PARA GERAÇÃO DE
INSTÂNCIAS XBRL**

Salvador
2023

HENDERSON ACOSTA BRAGANÇA

**XBRL PROCESSOR: UMA FERRAMENTA PARA GERAÇÃO DE
INSTÂNCIAS XBRL**

Dissertação apresentada ao Mestrado em Sistemas e
Computação da Universidade Salvador – UNIFACS, como
requisito parcial para obtenção do grau de Mestre.

Orientador: Prof. Dr. Paulo Caetano da Silva
Coorientador: Prof. Daniel José Díaz.

Salvador
2023

Ficha Catalográfica elaborada pelo Sistema de Bibliotecas da Universidade Salvador - UNIFACS.

Bragança, Henderson Acosta

XBRL PROCESSOR: uma ferramenta para geração de instâncias XBRL/
Henderson Acosta Bragança. - Salvador: UNIFACS, 2023.

74 f.: il.

Dissertação apresentada ao Programa de Pós-Graduação, Mestrado em Sistemas e
Computação da Universidade Salvador - UNIFACS, como requisito parcial para
obtenção do título de Mestre.

Orientador Prof. Dr. Paulo Caetano da Silva.

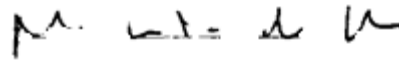
Coorientador: Prof. Daniel José.

1. Ciência da Computação. 2. XBRL. 3. XML. 4. Integração de dados. 5.
mapeamento de dados. I. Silva, Paulo Caetano da, orient. II. Díaz, Daniel José, co-
orient. III. Título.

CDD:004

TERMO DE APROVAÇÃO**HENDERSON ACOSTA BRAGANÇA****XBRL PROCESSOR: UMA FERRAMENTA PARA GERAÇÃO DE INSTÂNCIAS
XBRL**

Dissertação aprovada como requisito parcial para obtenção do grau de Mestre em Sistemas e Computação, Universidade Salvador - UNIFACS, pela seguinte banca examinadora:

**Paulo Caetano da Silva**

Doutor em Ciência da Computação pela Universidade Federal de Pernambuco, Brasil
Universidade Salvador - UNIFACS

**Daniel José Diaz**

Contador pela Universidade Nacional de Rosário
Professor da Facultad de Ciencias Económicas y Estadística
Universidade Nacional do Rosário - UNR

**Edson Luiz Riccio**

Doutor em Administração pela Universidade de São Paulo, Brasil
Universidade de São Paulo - USP

Joberto Sérgio Barbosa Martins

Doutor Université Pierre et Marie Curie, França
Universidade Salvador - UNIFACS



Documento assinado digitalmente

JOBERTO SERGIO BARBOSA MARTINS

Data: 29/05/2023 07:54:33-0300

Verifique em <https://validar.iti.gov.br>

Salvador, 26 de maio de 2023

“A filosofia da qualidade as perguntas, a tecnologia da qualidade as respostas.”

Autor desconhecido

AGRADECIMENTOS

Agradeço inicialmente ao autor e consumidor da nossa fé (Hebreus 12:2), ao Deus de Abraão, Isaque e Jacó (Êxodo 3:6) por permitir a realização deste sonho.

A minha amada esposa Mariana Bragança, a minha filha Manuela Bragança e filho Pedro Bragança pelo amor, conselhos, incentivos e suporte, me mantendo firme no propósito. Aos meus pais Árisson Bragança e Selma Bragança que me guiaram desde os meus primeiros passos para que não tivesse medo de atingir grandes objetivos. Aos familiares que sempre estiveram na torcida e disponíveis para ajudar.

Ao grande amigo Silas Pinho Ladislau que além de incentivar, contribuiu para o desenvolvimento desta Dissertação. Deus em sua infinita sabedoria nos disse em Provérbios 18:24, que em nossa jornada encontraremos amigos mais próximos que um irmão e sou feliz por ter encontrado um irmão na fé e na vida. Agradeço pelas incontáveis horas de trabalho juntos, criando as ideias e escrevendo os códigos.

Ao mestre, amigo e incentivador Professor Dr. Paulo Caetano minha eterna gratidão pelas aulas, pelo incentivo em me candidatar ao Mestrado e pela orientação durante toda a pós-graduação. Certamente sou uma pessoa melhor após todos os ensinamentos que levarei por toda a vida. Ao senhor, querido Professor Paulo, meu profundo carinho e gratidão por acreditar em mim.

Agradeço ao Coordenador do Mestrado Dr. Joberto Martins pela grande ajuda nos momentos críticos que a caminhada nos reservou, bem como a sua equipe, na pessoa da querida Josiane Melo.

Bendirei o Senhor o tempo todo! Os meus lábios sempre o louvarão. Minha alma se gloriará no Senhor; ouçam os oprimidos e se alegrem. Proclamem a grandeza do Senhor comigo; juntos exaltemos o seu nome. (Salmos 34:1-3)

RESUMO

É evidente o crescimento da utilização da tecnologia *eXtensible Business Reporting Language* (XBRL) no contexto de reportes financeiros na internet, seja por suas vantagens e benefícios ou por imposições governamentais, no entanto, os dados a serem transportados por essa linguagem estão, em sua maioria, armazenados em estruturas definidas como banco de dados, alguns relacionais outros *Not Only SQL* (NoSQL). A necessidade de integração da tecnologia XBRL com outras tecnologias de armazenamento de dados vem crescendo continuamente, sendo necessário pesquisas com o objetivo de buscar uma solução para o mapeamento de dados entre estes ambientes. As possíveis dificuldades em integrar XBRL com outras tecnologias, banco de dados relacionais ou NoSQL, arquivos CSV, JSON, necessitam ser mapeadas e superadas. A geração de documentos XBRL a partir de banco de dados pode ser custosa, uma vez que não existe alternativa nativa que exportem do sistema gerenciador de banco de dados, os dados em XBRL. Para isso, é necessário sistemas de terceiros e específicos para gerar os documentos XBRL, o que adiciona complexidade, uma vez que esses documentos não se conectam ao sistema gerenciador de banco de dados. Estas dificuldades acarretam problemas de desempenho e armazenamento e em casos de grande volume de dados, como é o de entrega de dados para órgãos governamentais, a complexidade aumenta. Destarte, se faz imprescindível o estudo sobre técnicas e métodos que permitam inferir uma solução para realizar essa integração e/ou mapeamento, preferencialmente de forma genérica, que abarque a estrutura de dados XBRL e os principais modelos de dados utilizados atualmente, i.e. SGBD's relacionais, NoSQL, JSON ou arquivos CSV. Esta dissertação apresenta uma solução de *Extract, Transform and Load* (ETL), voltados para a extração de dados em fontes diversas e a geração de instâncias XBRL, intitulada XBRL Processor. A ferramenta contempla diferentes tipos de fonte de dados, recupera informações de instâncias XBRL modelo e gera instância XBRL com os dados carregados da fonte de dados e as informações fornecidas na instância modelo. Adicionalmente, foram implementadas as parametrizações para atender a entrega da Matriz de Saldos Contábeis para o Sistema de Informações Contábeis e Financeiras da Secretaria do Tesouro Nacional do Brasil e realizado um estudo de caso, de modo a validar a ferramenta XBRL Processor.

Palavras-chave: XBRL; XML; integração de dados; mapeamento de dados.

ABSTRACT

It is evident the growth of the use of eXtensible Business Reporting Language (XBRL) technology in the context of financial reporting on the internet, either for its advantages and benefits or for government impositions, however, the data to be transported by this language are mostly stored in structures defined as databases, some relational others Not Only SQL (NoSQL). The need for integration of XBRL technology with other data storage technologies has been growing continuously, requiring research in order to seek a solution for data mapping between these environments. The possible difficulties in integrating XBRL with other technologies, relational database or NoSQL, CSV files, JSON, need to be mapped and overcome. The generation of XBRL documents from the database can be costly, since there is no native alternative that export the data in XBRL from the database management system. For this, third-party and specific systems are required to generate the XBRL documents, which adds complexity since these documents do not connect to the database management system. These difficulties lead to performance and storage problems, and in cases of large data volume, such as data delivery to government agencies, the complexity increases. Thus, it is essential to study techniques and methods that allow inferring a solution to perform this integration and / or mapping, preferably in a generic way, that encompasses the XBRL data structure and the main data models currently used, i.e. Relational DBMS, NoSQL, JSON or CSV files. This dissertation presents an Extract, Transform and Load (ETL) solution, aimed at reading data from various sources and the generation of XBRL instances, entitled XBRL Processor. The tool contemplates different types of data sources, retrieves information from model XBRL instances, and generates XBRL instance with the data loaded from the data source and the information provided in the model instance. In addition, the parameterizations were implemented to meet the delivery of the Matrix of Accounting Balances to the Accounting and Financial Information System of the Secretariat of the National Treasury of Brazil and a case study was carried out, in order to validate the XBRL Processor tool.

Keywords: XBRL; XML; data integration; data mapping.

LISTA DE FIGURAS

Figura 1 - Etapas da metodologia.....	20
Figura 2 - Distribuição dos 393 artigos encontrados nos repositórios de pesquisa.....	31
Figura 3 - Modelo conceitual para geração de instâncias XBRL.....	47
Figura 4 - Estrutura de pastas do código.....	50
Figura 5 - Dados de conexão com o SGBD MySQL.....	51
Figura 6 - Implementação do SGBD MongoDB.....	52
Figura 7 - Implementação do SGBD SQLite3.....	53
Figura 8 - Implementação do SGBD MySQL.....	54
Figura 9 - Estrutura da MSC definida pelo Siconfi.....	55
Figura 10 - Diagrama de sequência com o sistema gerenciador de banco de dados.....	56
Figura 11 - Instância XBRL modelo da MSC.....	57
Figura 12 - Diagrama de componentes da ferramenta XBRL Processor.....	58
Figura 13 - Fragmento do código do arquivo main contendo a taxonomia da MSC.....	59
Figura 14 - Resposta da equipe de atendimento do Siconfi.....	62
Figura 15 - Primeiras linhas da Instância XBRL gerada pela ferramenta XBRL Processor....	63
Figura 16 - Resposta da validação da instância XBRL feita pelo sistema do Siconfi.....	64
Figura 17 - Pedido de desconsideração do exercício 2017.....	64
Figura 18 - Validação da instância em ambiente de produção.....	65

LISTA DE TABELAS

Tabela 1 - Objetivos e soluções propostas dos trabalhos relacionados à XBRL.....	32
Tabela 2 - Objetivos e soluções propostas dos trabalhos selecionados que tratam da XML ...	33

LISTA DE SIGLAS

ANSI	<i>American National Standards Institute</i>
BD	<i>Banco de Dados</i>
BI	<i>Business Intelligence</i>
CSBE	<i>Comitê de Supervisores Bancários Europeu</i>
CSV	<i>Comma Separated Values</i>
DW	<i>Data Warehouse</i>
ETL	<i>Extract, Transform and Load</i>
LRF	<i>Lei de Responsabilidade Fiscal</i>
JSON	<i>JavaScript Object Notation</i>
MSC	<i>Matriz de Saldos Contábeis</i>
NoSQL	<i>Not Only SQL</i>
RDB	<i>Relational Data-base</i>
RSL	<i>Revisão Sistemática da Literatura</i>
RTA	<i>Remote Table Access</i>
SEFIN – RO	Secretaria de Finanças do Estado de Rondônia
SGBD	<i>Sistema Gerenciador de Banco de Dados</i>
SIAF	<i>Sistemas Integradores da Administração Financeira</i>
SICONFI	<i>Sistema de Informações Contábeis e Fiscais</i>
SQL	<i>Structured Query Language</i>
STN	<i>Secretaria do Tesouro Nacional</i>
TCP	<i>Transmission Control Protocol</i>
US-SEC	<i>United States - Securities and Exchange Commission</i>
WOD	<i>Web Of Data</i>
XBRL	<i>Extensible Business Report Language</i>
XML	<i>eXtensible Markup Language</i>

SUMÁRIO

1. INTRODUÇÃO	14
1.1. CONTEXTUALIZAÇÃO	14
1.2. MOTIVAÇÃO	17
1.3. OBJETIVOS	19
1.4. METODOLOGIA	19
1.5. ORGANIZAÇÃO DA DISSERTAÇÃO	20
2. FUNDAMENTAÇÃO TEÓRICA	22
2.1. EXTENSIBLE BUSINESS REPORTING LANGUAGE (XBRL)	22
2.2. BANCOS DE DADOS	23
2.2.1. BANCOS DE DADOS RELACIONAIS.....	23
2.2.2. BANCO DE DADOS SQLite3.....	24
2.2.3. BANCO DE DADOS MySQL.....	24
2.2.4. BANCO DE DADOS NoSQL	24
2.2.5. BANCO DE DADOS MongoDB	25
2.3. ARQUIVOS CSV	25
2.4. LINGUAGEM TypeScript	25
2.5. SECRETARIA DO TESOURO NACIONAL (STN)	26
2.6. MATRIZ DE SALDOS CONTÁBEIS	26
2.7. CONSIDERAÇÕES FINAIS	27
3. TRABALHOS CORRELATOS	28
3.1. METODOLOGIA	28
3.2. RESULTADOS DA REVISÃO DA LITERATURA	30
3.3. MAPEAMENTO DE DADOS PARA XBRL.....	36
3.4. MAPEAMENTO DE DADOS RELACIONAIS OU NoSQL PARA XML.....	37
3.5. MAPEAMENTO DE DADOS RELACIONAIS OU NoSQL PARA JSON	39
3.6. MAPEAMENTO DE DADOS RELACIONAIS OU NoSQL PARA CSV	40
3.7. RESULTADOS REFERENTES ÀS QUESTÕES DE PESQUISA DA RSL	40
3.8. SOLUÇÕES PROPRIETARIAS	42
3.9. CONSIDERAÇÕES FINAIS	43
4. FERRAMENTA XBRL PROCESSOR	45
4.1. MODELO CONCEITUAL DA XBRL PROCESSOR	46
4.2. CONEXÃO COM BANCOS DE DADOS	47
4.2.1. GERAÇÃO DA INSTÂNCIA XBRL.....	48
4.3. IMPLEMENTAÇÃO	49
4.3.1. ASPECTOS DE IMPLEMENTAÇÃO.....	49

4.4. CONSIDERAÇÕES FINAIS	60
5. ESTUDO DE CASO PARA GERAÇÃO DA INSTÂNCIA XBRL E VALIDAÇÃO DA SUA ESTRUTURA	61
5.1. EXPERIMENTO	61
5.2. RESULTADOS	61
5.3. CONSIDERAÇÕES FINAIS	65
6. CONCLUSÃO	66
6.1. CONSIDERAÇÕES FINAIS	66
6.2. CONTRIBUIÇÕES	67
6.3. LIMITAÇÕES E TRABALHOS FUTUROS.....	68
6.4. PUBLICAÇÕES	69
7. REFERÊNCIAS	70

1. INTRODUÇÃO

Este capítulo tem a finalidade de contextualizar o tema investigado neste trabalho. Na Seção 1.1 é discutida a importância da exportação de dados na linguagem *eXtensible Business Reporting Language* (XBRL)¹ oriundos de sistemas gerenciadores de banco de dados relacionais e *Not Only SQL* NoSQL². A Seção 1.2 motiva a solução de exportação de dados na linguagem XBRL. A Seção 1.3 descreve os problemas de pesquisa e hipótese. A Seção 1.4 apresenta os objetivos almejados nesta dissertação. A Seção 1.5 descreve a metodologia usada para o desenvolvimento deste trabalho. Por fim, a Seção 1.6 descreve como esta dissertação está organizada.

1.1. CONTEXTUALIZAÇÃO

Transportar dados financeiros e contábeis através da internet mantendo sua integridade estrutural e semântica tem sido alcançado utilizando a tecnologia XBRL. A XBRL é uma tecnologia derivada da *Extensible Markup Language* (XML), que é considerada pelo *World Wide Web Consortium* (W3C³) como um importante meio de intercâmbio de dados na internet, uma vez que possibilita a troca de dados independente de plataforma (Jayashree; Priya, 2020). O projeto da tecnologia XML favorece a simplicidade, generalidade e a usabilidade (Zhu, Yu, Fan, Sun, 2017), estes atributos propiciam o uso da tecnologia XML na internet e como meio de integração e intercâmbio de dados, para a qual técnicas e alternativas foram propostas para a integração e mapeamento de dados entre documentos XML, os quais são usados como meio de transporte de dados na internet, e diferentes bases de dados.

No entanto, propostas como a da Fujitsu⁴ ou Oracle⁵ adicionam um custo computacional, pois é necessário criar bases de dados paralelas com os dados advindos dos documentos XML, além de armazenar o documento XML para resguardar a integridade das informações. Em

¹ <https://www.xbrl.org/>

² É um acrônimo para “Not Only SQL”

³ <https://www.w3.org/>

⁴ <https://www.fujitsu.com/global/products/software/middleware/application-infrastructure/interstage/solutions/xbrl/>

⁵ <https://www.oracle.com/performance-management/#rc30p7>

outros casos é preciso desenvolver aplicações específicas para mediar o intercâmbio de dados entre os diferentes ambientes, i.e. XML, bases relacionais, CSV⁶, NoSQL e JSON⁷.

A tecnologia XBRL, foi desenvolvida para o intercâmbio de informações financeiras na internet, sendo consolidada como o padrão a ser utilizado por órgãos governamentais em diversos países (Dunce; Silva; Viana, 2013), esta tecnologia possui a capacidade de transportar os dados financeiros juntamente com a sua semântica, algo imprescindível para uma análise correta da informação. Por portar os dados juntamente com a sua semântica, exclui-se a necessidade de descobrir o significado da informação fornecida, quando não é acompanhada do seu contexto, ou seja, sem a semântica (Bragança; Ladislau; Silva; Silva, 2019).

A XBRL é uma tecnologia que se tornou padrão internacional para o intercâmbio de dados financeiros, e.g. a US-SEC⁸, Comitê Europeu de Supervisores Bancários (CEBS)⁹, Banco da Espanha, Banco do Japão, são apenas alguns exemplos de órgão governamentais que adotaram a XBRL para o intercâmbio de dados com suas organizações supervisionadas¹⁰. No Brasil, diante das imposições normativas, como é o caso da lei 12.527/11 que dispõe sobre o acesso às informações dos Estados, Municípios e do Distrito Federal, popularmente chamada de Lei da Transparência, coadunado com a norma 896/17 da Secretaria do Tesouro Nacional¹¹, direciona as esferas governamentais para o uso do XBRL em suas publicações conexas às demonstrações contábeis junto aos órgãos de controle.

Tanto os governos em seus diferentes níveis (federal, estadual e municipal), quanto as organizações privadas, não possuem ferramentas gratuitas e *open-source* que possibilitem a exportação das informações financeiras e contábeis para o formato XBRL, principalmente quando extraídas diretamente das bases de dados relacionais ou NoSQL, criando uma dificuldade para as organizações privadas e governos para representar seus dados financeiros e fiscais em XBRL.

Existem diferentes tecnologias para o armazenamento de dados, baseadas em diferentes modelos, entre eles, o mais comum é o relacional e, mais recentemente, o NoSQL, comumente utilizado na internet. O modelo de dados relacional representa o banco de dados como uma

⁶ Comma Separated Values (valores separados por vírgula)

⁷ JSON é o acrônimo de Java Script Object Notation, é um padrão de troca de dados entre sistemas.

⁸ <https://www.sec.gov/>

⁹ <https://www.europeansources.info/record/committee-of-european-banking-supervisors-cebs/>

¹⁰ <https://www.xbrl.org/the-standard/why/ten-countries-with-open-data/>

¹¹ <https://www.gov.br/tesouronacional/pt-br>

coleção de relações (Navathe, 2013), que após a sua concepção na década de 60, passou a ser utilizado por grande parte das aplicações de software, sendo um modelo que deverá continuar a ser usado por muitos anos em razão da sua robustez (Navathe, 2013). Devido a importância e ampla utilização nas aplicações privadas e governamentais, é imprescindível considerar esse modelo como fonte de dados. Também a abordagem relacionada a arquivos CSV como fonte de dados é pertinente, uma vez que existem cenários¹² nos quais os dados são exportados apenas em arquivos CSV (Bragança et al., 2019).

Os bancos de dados NoSQL trabalham com dados desnormalizados (B. E. Soares & Boscaroli, 2013). Os bancos de dados NoSQL estão sendo adotados por grandes empresas como a Google, o Facebook e o Twitter (B. E. Soares & Boscaroli, 2013), uma vez que este modelo possui arquitetura que facilita o tratamento de expressiva demanda por consultas, vinculada ao elevado número de informações, com alta escalabilidade dos dados (B. E. Soares & Boscaroli, 2013). Por conseguinte, os bancos de dados NoSQL também são considerados como alternativa de fonte de dados na solução proposta neste trabalho.

A Secretaria do Tesouro Nacional (STN) através do Sistema de Informações Contábeis e Fiscais do Setor Público Brasileiro (Siconfi)¹³ definiu como o principal documento para a melhoria da qualidade da informação no Setor Público Brasileiro as instâncias XBRL da Matriz de Saldos Contábeis (STN, 2017). A Matriz de Saldos Contábeis (MSC) é uma estrutura planejada para conter os detalhes da informação contábil do ente federativo¹⁴ sendo recomendado pela STN a extração direta da fonte de dados para minimizar erros de preenchimento de planilhas ou formulários (STN, 2017).

O propósito desta dissertação é desenvolver uma solução *open-source* que atenda aos mais diversos cenários de fontes de dados, exportando as informações referentes a MSC para instâncias XBRL, atendendo a exigência da STN através do Siconfi. A partir desse contexto, a seguir, a motivação desse trabalho é descrita.

¹² Os Sistemas Integradores da Administração Financeira (SIAF) dos Estados na sua maioria são incapazes de gerar a instância XBRL por utilizarem o SGBD Adabas/SoftwareAG, mas possuem a capacidade de gerar arquivos CSV.

¹³ <https://Siconf.tesouro.gov.br/Siconf/index.jsf>

¹⁴ Ente federativo pode ser entendido também como um Estado, sendo uma unidade autônoma (autogoverno, autolegislação e autoarrecadação) dotada de governo próprio, constituição e que, com outros estados, forma uma federação.

1.2. MOTIVAÇÃO

Para que uma informação seja contextualizada em determinado domínio, ela deve ser interligada com outras informações, sejam elas contextuais, informações do passado, experiências ou informações futuras (Cerqueira; Silva, 2021; Beelitz, 2017). No domínio financeiro e contábil a tecnologia que atende a esses requisitos é a XBRL.

Em 2008 Silva afirmou que a tecnologia XBRL estava se tornando um padrão tecnológico para a troca, armazenamento e divulgação de informações financeiras na internet (Silva; Silva; Santos; Cruz, 2008; Gray; Miller, 2009), e de fato a previsão se confirmou, a adoção da tecnologia por relevantes instituições tem impulsionado a utilização da XBRL, como é o caso da US-SEC¹⁵ que apontou a tecnologia XBRL como a chave para a sua modernização (GRAY; MILLER, 2009).

A tecnologia XBRL por ser robusta e representar completamente o domínio financeiro e contábil, traz consigo a complexidade inerente a soluções destinadas a problemas complexos, como é o caso do transporte de informações financeiras e contábeis com todo o seu arcabouço contextual.

A geração de instâncias XBRL utilizando como fonte os repositórios de dados financeiros e contábeis não é trivial (Asimadi; Reiff-Marganec; Donnelly; Baker; Fang, 2017), apenas ferramentas proprietárias (e.g. Amelkis, aSISSt, Vizer, etc.)¹⁶ disponíveis no mercado são capazes de extrair de forma automática os dados armazenados em diferentes repositórios que usam diversas tecnologias de armazenamento (e.g. relacional, NoSQL, CSV, etc). Ainda assim, normalmente essas ferramentas manipulam dados de um modelo específico, i.e. elas não possuem generalidade suficiente para manipularem esses diversos tipos de formatos de dados e XBRL. A ausência de ferramentas *open-source* pode dificultar a adoção desta tecnologia de forma massiva, principalmente no território brasileiro, cujo único projeto em operação é o da Secretaria do Tesouro Nacional (STN), denominado Siconfi. O Siconfi exige dos entes federados a entrega das informações contábeis e financeiras baseada na tecnologia XBRL, para isto, sistemas proprietários são utilizados. Atender as exigências da STN no que se refere à entrega da MSC em instância XBRL permite ao ente cumprir a Lei de Responsabilidade Fiscal a qual todo o setor público brasileiro está sujeito.

¹⁵ <https://www.sec.gov/>

¹⁶ <https://www.xbrl.org/the-standard/how/tools-and-services/>

Para atender a legislação, as unidades da federação precisam inserir os dados contábeis e financeiros em sistemas proprietários, em muitos casos de forma manual, para gerar o documento XBRL, uma vez que a conexão com a fonte de dados, e.g. bancos de dados relacionais, não é simples. Isto gera custos de recursos humanos e computacionais, oriundos da baixa interoperabilidade entre os ambientes XBRL e as fontes de dados, o que implica em custos financeiros adicionais.

Pode-se perceber no trabalho de Bragança et al. (2019) que existe uma carência nos poderes executivos do Brasil, seja na esfera federal, estadual ou municipal, por soluções que atendam a demanda na geração das instâncias XBRL a serem entregues ao Siconfi.

Para atender essa demanda, busca-se neste trabalho desenvolver uma solução ETL *open-source* dividida em dois estágios de execução de modo a facilitar o desenvolvimento de novas funcionalidades, o primeiro estágio destina-se a conexão com a fonte de dados e o segundo a vincular os dados com a taxonomia do Siconfi, possibilitando a geração da instância XBRL¹⁷. Nesta dissertação é apresentada uma ferramenta que busca atender aos mais diversos cenários de intercâmbio de dados financeiros, com configurações reutilizáveis e com redução da complexidade do ambiente. A solução foi planejada de forma a facilitar a manutenção e extensão para gerar instâncias das mais variadas taxonomias.

Desta forma, o problema de pesquisa central que norteia o desenvolvimento desta dissertação é:

- Questão 1: Existem soluções que permitam o mapeamento de dados de diferentes formatos para XBRL?

A partir desta questão de pesquisa principal (Questão 1) pôde-se inferir outras questões de pesquisa:

- Questão 2: As soluções existentes são *open-source*, ou seja, são de código aberto e gratuitas?
- Questão 3: as soluções propostas possuem generalização suficiente de maneira que permitam o mapeamento de qualquer modelo de dados (e.g. relacional, NoSQL, CSV, JSON) para XBRL?
- Questão 4: as soluções propostas atendem aos requisitos do Siconfi e Matriz de Saldo Contábeis (MSC) da STN?

¹⁷ A taxonomia XBRL é um conjunto de documentos baseados nas tecnologias XML Schema e Xlink que permite a definição de conceitos financeiros e contábeis e seus relacionamentos.

- Questão 5: É possível construir uma ferramenta com a qual qualquer taxonomia possa ser utilizada para gerar instâncias XBRL de qualquer fonte de dados?

Para o problema de pesquisa aqui investigado, formulou-se a seguinte hipótese:

- Hipótese 1: É possível construir uma ferramenta de extração, transformação e carga de dados a partir de qualquer modelo de dados, seja de SGBD's relacionais, NoSQL ou arquivos CSV, e exportá-los como instâncias XBRL, capaz de adaptar-se às necessidades conexas à fonte de dados e taxonomias dos utilizadores.

1.3. OBJETIVOS

Este trabalho tem como objetivo geral investigar sobre o mapeamento de diferentes modelos de dados para XBRL de forma que se possa propor uma solução generalizada para a extração de dados e geração de instâncias XBRL baseadas em qualquer taxonomia. Pretende-se fornecer uma ferramenta que atenda aos mais diversos cenários e possibilite a extensão da solução. Para alcançar esse objetivo, algumas metas foram almeçadas:

- Investigar e identificar na literatura e no mercado sobre soluções que permitam realizar o mapeamento de dados de diferentes formatos para XBRL;
- A partir dos resultados encontrados na revisão da literatura propor uma solução genérica, i.e. utilizando dados de diferentes modelo para instâncias XBRL modeladas por qualquer taxinomia;
- Projetar uma solução que seja modular, com código simplificado e parametrizações reutilizáveis;
- Desenvolver uma solução que consiga conectar-se as mais diversas fontes de dados, sejam elas em ambientes relacionais ou NoSQL;
- Desenvolver uma solução que vincule dados a taxonomia XBRL de forma a propiciar a manutenção e reutilização;
- Validar as instâncias XBRL geradas pela ferramenta a partir da taxonomia da matriz de saldos contábeis (MSC) do Siconfi;

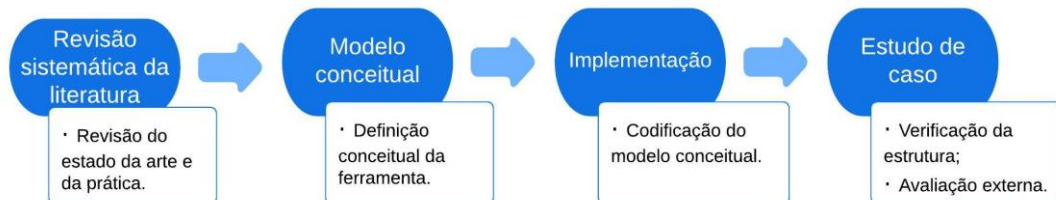
1.4. METODOLOGIA

Para elaboração desta dissertação, utilizou-se uma metodologia composta por quatro etapas. Conforme ilustrado na Figura 1, as etapas são: revisão sistemática da literatura, modelo conceitual, implementação e estudo de caso.

Na etapa da revisão sistemática da literatura, procurou-se entender como se dá o mapeamento dos dados, seja de bancos de dados relacionais ou outras fontes diversas para

serem instanciadas em XBRL. Por fim, nesta etapa da metodologia identificou-se o problema abordado nesta dissertação, ou seja, ausência de ferramentas que gerem instâncias XBRL com dados provenientes de fontes de dados, sejam elas SGBD's relacionais ou NoSQL.

Figura 1 - Etapas da metodologia



A segunda etapa foi utilizada para projetar o conceito de uma solução ETL *open-source* dividida em dois estágios de execução de modo a facilitar o desenvolvimento de novas funcionalidades. O modelo conceitual busca atender aos mais diversos cenários com configurações reutilizáveis e com redução da complexidade inerente à taxonomia XBRL.

Em seguida, a ferramenta foi codificada. Desse modo, foi definido o escopo do desenvolvimento onde foram selecionados a taxonomia a ser utilizada, a linguagem de programação e os SGBD's, adicionou-se os comentários no código seguindo as definições da linguagem.

Por conseguinte, avaliou-se a ferramenta utilizando um estudo de caso. Em vista disso, a ferramenta foi submetida a dois cenários realistas. Submeteu-se a instância gerada pela ferramenta a dois ambientes para certificar a estrutura da instância.

1.5. ORGANIZAÇÃO DA DISSERTAÇÃO

O restante deste trabalho está organizado da seguinte maneira:

- O Capítulo 2 mostra a fundamentação teórica contendo os conceitos necessários ao entendimento do problema e solução abordada nesta dissertação;
- O Capítulo 3 apresenta os trabalhos relacionados a esta dissertação, encontrados no estado da arte e da prática. Esses trabalhos mostram definições conceituais de termos, técnicas de mapeamento de dados, soluções baseadas em frameworks e percalços superados ou identificados, descritos na literatura para gerar instâncias XBRL;
- O Capítulo 4 descreve a abordagem conceitual proposta para exportar dados de fontes diversas em instâncias XBRL;

- O Capítulo 5 mostra o desenvolvimento da ferramenta para extração, transformação e carga de dados (ETL) para a tecnologia XBRL, nomeada como XBRL Processor;
- O Capítulo 6 expõe as conclusões desta dissertação, suas contribuições, limitações e propostas de trabalhos futuros.

2. FUNDAMENTAÇÃO TEÓRICA

Neste capítulo, são apresentados os conceitos essenciais ao entendimento desta dissertação. Assim, a Seção 2.1 mostra a definição da tecnologia XBRL. O conceito de bancos de dados e suas abordagens na Seção 2.2. A Seção 2.3 apresenta o arquivo CSV. Em seguida, a linguagem TypeScript na Seção 2.4. A Secretaria do Tesouro Nacional e o Sistema de Informações Contábeis e Fiscais do Setor Público Brasileiro são apresentados na Seção 2.5. Portanto, a Seção 2.6 apresenta o conceito da Matriz de Saldos Contábeis. Por fim, as considerações finais deste capítulo são apresentadas na Seção 2.7.

2.1. EXTENSIBLE BUSINESS REPORTING LANGUAGE (XBRL)

A XBRL é uma tecnologia aberta estendida da linguagem XML, especificamente criada para o intercâmbio e análise de informações financeiras na internet (Silva et al., 2008). Sua estrutura básica é composta por: (i) documento de instância, que contém os dados a serem informados, juntamente com as informações de um contexto específico e (ii) a taxonomia com a definição dos conceitos dos termos contábeis e financeiros e seus relacionamentos semânticos (Cerqueira; Silva, 2016).

No documento de instância os fatos financeiros estão contidos, é neste documento que os dados contábeis e financeiros são inseridos e para isto segue uma estrutura que é definida na taxonomia (Cerqueira; Silva, 2016).

A taxonomia é descrita por um documento XML Schema (Silva et al., 2008), que contém os conceitos relacionados aos dados contábeis e financeiros. Esses conceitos possuem nome e um tipo (Cerqueira; Silva, 2016). Outro componente importante da taxonomia é conhecido como linkbase. Os linkbases da taxonomia, baseados em XLink (Silva et al., 2008), expressam as relações semânticas existentes entre os conceitos e entre eles e o documento de instância, associando os conceitos com sua documentação (Cerqueira & Silva, 2016).

No ano de 2021 o consórcio responsável pela XBRL definiu novas formas de utilizar o padrão, definindo como recomendação final além da XBRL estendida da XML, a XBRL estendida do JSON e CSV¹⁸.

¹⁸ <https://www.xbrl.org/news/2021-saw-new-specifications-launched-lift-off-for-xbrl-json-and-xbrl-csv>

A XBRL tem benefícios bem estabelecidos e documentados, para Riccio, Sakata, Moreira, e Quoniam (2006) a XBRL fornece independência de plataforma tecnológica, interoperabilidade e preparação eficiente de relatórios financeiros.

2.2. BANCOS DE DADOS

Os bancos de dados fazem parte do cotidiano da população moderna (Navathe, 2013), portanto é possível inferir que as organizações, das mais variadas atividades (e.g. indústria, entretenimento ou no contexto da Internet), fazem uso de diferentes tecnologias de banco de dados para se manter competitiva (Navathe, 2013).

O conceito dado por Spink para bases de dados nos diz que “uma base de dados é uma coleção de dados logicamente relacionados, com algum significado” (Spink; Arouca; Teixeira, 2002). No entanto, o gerenciamento da base de dados é feito por uma coleção de softwares que permitem aos clientes criar e manipular bases de dados (Spink et al., 2002), esses gerenciadores são conhecidos pelo acrônimo SGBD (Sistemas Gerenciadores de Bancos de Dados). O uso de bancos de dados tornou-se indispensável no cotidiano da Internet e em ambientes onde é indispensável as características de Atomicidade, Consistência, Isolamento e Durabilidade (Soares; Boscarioli, 2013) que os Sistemas Gerenciadores de Banco de Dados proporcionam às aplicações.

Os bancos de dados relacionais utilizam a linguagem de consulta estruturada “*Structured Query Language*” (SQL) para manipular os dados e controlar os SGBD’s incluindo, definições de dados, controle de acesso e compartilhamento de dados. SQL é, portanto, uma linguagem abrangente para controlar e interagir com SGBD’s relacionais (Groff; Weinberg, 1999).

2.2.1. Bancos de dados relacionais

Existem diferentes tecnologias para o armazenamento de dados, baseadas em diferentes modelos, entre eles o mais comum é o Relacional e, mais recentemente, o NoSQL, comumente usado na internet. O modelo de dados relacional representa o banco de dados como uma coleção de relações (Navathe, 2013), que após a sua aparição na década de 60, passou a ser utilizado por grande parte das aplicações de software, sendo um modelo que deverá continuar a ser usado por muitos anos em razão da sua robustez (Navathe, 2013). Devido a importância e ampla

utilização nas aplicações privadas e governamentais, é imprescindível considerar esse modelo como fonte de dados.

2.2.2. Banco de dados SQLite3

O sistema gerenciador SQLite3 é uma biblioteca escrita em C que implementa um mecanismo de banco de dados SQL estável, multiplataforma com ampla utilização em dispositivos móveis e comumente incorporado em softwares e.g. Adobe, Dropbox, McAfee, Firefox, Skype (Bin; Xu; Cong; Zhen-Hua; Nan, 2022).

A ampla aceitação dessa biblioteca está associada a não utilização de um processo de servidor separado, o SQLite3 lê e grava os dados diretamente em arquivos de disco com os recursos de tabelas, índices e gatilhos. O SQLite3 com todos os recursos ativados possui tamanho inferior a 750 KiB mantendo a atomicidade, consistência, isolamento e durabilidade esperados de SGBD (Prammer; Sahadevan Rajesh; Chen; Patel, 2022).

2.2.3. Banco de dados MySQL

O MySQL é um popular SGBD desenvolvido, distribuído e suportado pela Oracle Corporation (Christudas, 2019), possui licença de utilização disponível de duas maneiras, como produto gratuito sob os termos *General Public License* (GNU) ou pode ser licenciado por sua detentora dos direitos, a Oracle Corporation (Christudas, 2019).

O SGBD MySQL destina-se a sistemas considerados como críticos que exigem a utilização de *multithread's*, multiusuários e consultas concorrentes, ao mesmo passo que mantem a flexibilidade e velocidade de acesso (Rodriguez; Guardo, 2005).

2.2.4. Banco de dados NoSQL

Os bancos de dados NoSQL trabalham com dados denormalizados em formato JSON (JavaScript Object Notation) (Soares; Boscarioli, 2013). Os bancos de dados NoSQL estão sendo adotados por grandes empresas como a Google, o Facebook e o Twitter (Soares; Boscarioli, 2013), uma vez que este modelo possui arquitetura diferenciada para facilitar o tratamento de expressiva demanda por consultas, vinculado ao elevado número de informações que manipulam, com alta escalabilidade de dados (Soares; Boscarioli, 2013). Por conseguinte, os bancos de dados NoSQL também são considerados como alternativa de fonte de dados na solução proposta neste trabalho.

2.2.5. Banco de dados MongoDB

Quando relacionado ao volume de consultas efetuadas por análises de Big Data, a alternativa escolhida no contexto de armazenamento NoSQL é o MongoDB (Győrödi; Dumșe-Burescu; Zmaranda; Győrödi, 2022). O nome MongoDB é originário de “*humongous*” que significa algo grande, portanto, foi desenvolvido com foco em processamento de dados em larga escala (Celesti; Fazio; Villari, 2019).

A principal qualidade do MongoDB que coaduna com o volume de consultas aplicados ao sistema é a escalabilidade do armazenamento. O MongoDB fornece ferramentas úteis para distribuir com eficiência dados em várias máquinas, como *sharding* e replicação (Celesti et al., 2019).

2.3. Arquivos Comma-Separeted Value (CSV)

Os arquivos CSV (Comma-separeted value) ou em tradução livre, valores separados por vírgula, contém os registros de dados em textos simples separados por vírgula, onde cada linha representa um novo conjunto de registros (Debinski; Breitinger; Mohan, 2019).

O formato de arquivo CSV é utilizado para a transferência de informações entre softwares diferentes, inicialmente utilizado em 1972 pelo compilador IBM Fortran e continua sendo um recurso para a transferência de dados encontrado em softwares como e.g. Microsoft Excel, Google Sheets, Apple Numbers, OpenOffice (Mäs et al., 2018).

2.4. LINGUAGEM TypeScript

TypeScript é uma linguagem derivada do JavaScript e tem por objetivo o desenvolvimento de aplicativos em larga escala. A linguagem TypeScript oferece um sistema de módulos, classes e interfaces (Bierman; Abadi; Torgersen, 2014).

TypeScript é amplamente usada para aplicativos da web, aplicativos móveis, clientes de *desktop* e até mesmo desenvolvimento de *back-end*. Essa linguagem tem se mostrado uma opção superior a JavaScript devido ter as soluções de erros registrados no gitrub, solucionados 60% mais rápido que a observada nas aplicações JavaScript (Bogner; Merkel, 2022).

2.5. SECRETARIA DO TESOURO NACIONAL (STN)

A Secretaria do Tesouro Nacional (STN), órgão específico singular do Ministério da Fazenda e órgão central dos Sistemas de Administração Financeira Federal e de Contabilidade, diretamente subordinado ao Ministro de Estado da Fazenda do Brasil, onde sua missão é “gerir as contas públicas de forma eficiente e transparente, zelando pelo equilíbrio fiscal e pela qualidade dos gastos públicos, com vistas a contribuir para o desenvolvimento sustentável” (STN, 2017), que para tanto recebe as informações contábeis dos entes federados. Para isso, utiliza-se do Sistema de Informações Contábeis e Fiscais do Setor Público Brasileiro (Siconfi).

O foco principal do Siconfi é criar condições para a integração da contabilidade orçamentária, patrimonial e analítica, para a conformação de uma contabilidade pública moderna que viesse a ser aliada dos gestores públicos, ao permitir-lhes efetivo controle financeiro das diferentes instâncias administrativas e o fornecimento de informação precisa, confiável e oportuna aos múltiplos agentes públicos interessados (Bragança et al., 2019).

2.6. MATRIZ DE SALDOS CONTÁBEIS

A STN publicou a Instrução Normativa 896/17 que estabelece regras acerca da periodicidade, formato e sistema relativos à disponibilização das informações e dos dados contábeis, orçamentários e fiscais da União, dos estados, do Distrito Federal e dos municípios, e em seu Art. 14, alínea IV instrui o envio das informações no padrão XBRL (Bragança et al., 2019).

Diante da imposição legal a União, os estados, o Distrito Federal e os municípios passaram a ser obrigados a fornecer os dados contábeis e financeiros na tecnologia XBRL.

Dentre as peças contábeis a serem entregues a STN através do Siconfi, está a Matriz de Saldos Contábeis que é considerada como o principal documento de auxílio à melhoria da qualidade da informação no Setor Público Brasileiro (STN, 2017).

A Matriz de Saldos Contábeis pode ser entendida como uma declaração de formato estruturado e padronizado para envio de informações contábeis e fiscais extraídas do sistema contábil, semelhante a um balancete de verificação (Soares; Mallone; Andrade, 2021).

Para a STN a MSC deve ser extraída pelos entes da federação diretamente dos seus sistemas contábil-financeiros e encaminhados ao Siconfi, sem a necessidade de digitação, de modo a reduzir a possibilidade de erros no preenchimento das declarações e permitir o

compartilhamento dos dados com outros órgãos de governo, e.g. Secretaria da Previdência Social, Tribunais de Contas, Ministério da Saúde, Ministério da Educação, entre outros¹⁹.

2.7. CONSIDERAÇÕES FINAIS

Este capítulo listou os conceitos essenciais ao entendimento deste trabalho. Na Seção 2.1 mostrou-se a definição da tecnologia XBRL e seus atributos. Por seguida, a Seção 2.2 abordou o conceito de bancos de dados, bancos de dados relacionais e NoSQL. Ainda, nessa seção, foi apresentada a definição dos SGBD's SQLite3, MySQL e MongoDB.

Na Seção 2.3, apresentou-se o significado do arquivo CSV. As definições e vantagens relacionadas a utilização da linguagem TypeScript para o desenvolvimento da ferramenta XBRL Processor foram apresentados na Seção 2.4. Por fim, a Seção 2.5 e 2.6 trouxeram as definições relacionadas a Secretaria do Tesouro Nacional, o sistema utilizado para a recepção dos dados contábeis por parte do governo federal e a Matriz de Saldos Contábeis, que trabalham em conjunto para a análise e transparência dos dados financeiros e contábeis dos entes federados do Brasil.

Esses conceitos, listados neste capítulo, são utilizados em trabalhos encontrados no estado da arte. Porém, existe uma lacuna relacionada as Seções 2.5 e 2.6, conforme apresentado no Capítulo 3. Essa lacuna é preenchida pela proposta desta dissertação, a qual se trata de uma ferramenta para a leitura dos dados em fontes variadas e exportados em instâncias XBRL. A especificação dessa arquitetura proposta é descrita no Capítulo 4.

¹⁹ <https://Siconf.tesouro.gov.br/Siconf/pages/public/conteudo/conteudo.jsf?id=12302>

3. TRABALHOS CORRELATOS

Este capítulo apresenta trabalhos e estudos realizados pela comunidade científica, os quais estão relacionados a esta dissertação por tratarem de mapeamento de dados em bancos de dados para instâncias XBRL ou XML. Inicialmente, neste capítulo, a metodologia utilizada na pesquisa e as propostas dos trabalhos relacionados são descritas. Posteriormente, uma avaliação sobre esses trabalhos é apresentada, na qual se identifica algumas lacunas, que foram consideradas e abordadas nessa pesquisa.

Este capítulo está dividido da seguinte forma. A Seção 3.1 descreve a metodologia utilizada bem como a string de pesquisa. Na sequência, a Seção 3.2 mostra os trabalhos encontrados e selecionados para análise. Na Seção 3.3, o mapeamento de dados para XBRL é discutido. Posteriormente, a Seção 3.4, 3.5 e 3.6 discutem o mapeamento de dados relacionais ou NoSQL para XML, JSON e CSV respectivamente. A Seção 3.7 apresenta os resultados frente às questões da pesquisa. Finalmente, as considerações finais desse capítulo são descritas na Seção 3.8.

3.1. METODOLOGIA

Diante da abrangência e importância que XBRL tem alcançado no mercado financeiro, é oportuno conhecer o estado da arte das técnicas e aplicações voltadas para a geração do documento composto por fatos contextualizados, denominado instância XBRL, a partir de fontes de dados diversas. Objetiva-se conhecer as propostas de mapeamentos e/ou integração de dados para XBRL, de forma a se saber sobre as visões existentes referentes a instanciação dos dados em linguagem de marcação, com é o caso da XBRL. Para isso foi efetuada uma revisão sistemática da literatura (RSL) baseada nos trabalhos de (Kitchenham; Charters, 2007).

A formulação da questão de pesquisa teve como premissa identificar todos os trabalhos que tratam do mapeamento de dados para XBRL e a partir de XBRL para diferentes formatos de armazenamento de dados, independente do segmento de aplicação. Partindo desta premissa, a questão de pesquisa formulada para esta revisão sistemática é a seguinte: (Q1) Existem soluções de integração ou mapeamento de dados para XBRL e vice-versa? A partir desta questão, questões secundárias foram definidas de forma a se ter um melhor entendimento do problema: (Q1.1) Qual a necessidade de mapear os dados de fontes diversas para XBRL e vice-versa? (Q1.2) Quais tecnologias são usadas para mapear os dados XBRL? e (Q1.3) As propostas existentes são eficientes para abarcar qualquer tipo de fonte de dados e XBRL?

A partir destas questões de pesquisa foi planejada uma *string* de busca que pudesse facilitar se encontrar o maior número de artigos relacionados com a temática deste trabalho. Devido a massiva maioria dos trabalhos serem publicados na língua inglesa, os termos foram colocados neste idioma para alcançar maior quantidade. Fazendo uso dos recursos dos motores de busca, foi definida a *string* seguindo a lógica booleana na elaboração e aplicação da pesquisa.

(XBRL AND (Integration OR Mapping OR Framework OR Middleware) AND
(Relational OR Data OR Database OR NoSQL OR CSV OR JSON))

As buscas foram realizadas nas principais bibliotecas digitais disponíveis na área de computação: AIS²⁰, IEEE Xplore²¹, Biblioteca Digital ACM²², Science Direct²³ e Springer Link²⁴. Complementamos a busca utilizando o Google Scholar, dando assim uma maior abrangência da pesquisa em busca de trabalhos ainda não identificados e que pudessem estar em repositórios pouco conhecidos.

De modo a selecionar os trabalhos que tem relação com as questões da pesquisa definiu-se então os critérios de inclusão e exclusão.

Critérios de Inclusão:

- (CI1) O artigo aborda os temas das questões de pesquisa;
- (CI2) O artigo trata de dificuldades, questões críticas ou desafios relacionados às questões de pesquisa;
- (CI3) O artigo propõe alguma metodologia, modelo conceitual ou quadro de estudo relacionado às questões de pesquisa.

Critérios de Exclusão:

- (CE1) O foco principal não está relacionado com esta pesquisa;
- (CE2) O artigo é anterior ao ano de 2015;
- (CE3) O artigo não está disponível para download.

Para a condução do trabalho, a revisão sistemática selecionou os estudos, seguindo cinco etapas: (1) execução da busca; (2) aplicação do primeiro filtro, (3) aplicação do segundo filtro

²⁰ <https://aisel.aisnet.org/>

²¹ <https://ieeexplore.ieee.org/>

²² <https://dl.acm.org/>

²³ <https://www.sciencedirect.com/>

²⁴ <https://link.springer.com/>

e (4) aplicação do terceiro filtro (5) leitura e análise dos trabalhos selecionados (Felizardo, 2017).

Executando a primeira etapa, não limitamos a busca com nenhum tipo de critério além da *string* pré-definida. Neste momento foram identificados os títulos de artigos que coadunam com a questão da pesquisa, os quais foram inseridos no gerenciador de bibliografias Mendeley²⁵ e na ferramenta de auxílio à revisão sistemática da literatura Parsifal²⁶.

Em seguida aplicou-se o primeiro filtro, i.e. artigos selecionados por meio dos critérios de inclusão e exclusão, através da leitura dos *abstracts* dos trabalhos. No segundo filtro, foi realizada a leitura integral dos trabalhos resultantes do primeiro filtro, aplicando novamente os critérios de inclusão e exclusão. Nesta etapa os trabalhos que não atendiam aos critérios de inclusão, bem como aqueles que estavam fora da data de corte delimitada, trabalhos anteriores ao ano de 2015, foram rejeitados. Por fim, foram rejeitados os artigos que não estavam disponíveis para download.

Também é essencial avaliar a qualidade dos trabalhos (Kitchenham; Charters, 2007), logo, definiu-se critérios para verificar a qualidade apresentada nos artigos e que satisfizessem no mínimo 80% das seguintes questões:

- (QQ1) A solução está claramente detalhada?
- (QQ2) As tecnologias propostas não são obsoletas?
- (QQ3) Os resultados e conclusões são claramente explicados?
- (QQ4) Existe uma descrição clara do ambiente da solução proposta?
- (QQ5) Os objetivos da pesquisa e a motivação estão claramente definidos?
- (QQ6) O estudo apresenta uma solução de mapeamento de dados XBRL?

3.2. RESULTADOS DA REVISÃO DA LITERATURA

Apenas 4 estudos resultaram da pesquisa com a *string* definida na metodologia. Portanto, em função da necessidade de buscar uma solução para o mapeamento e integração entre XBRL e outros modelos de armazenamento de dados e por XBRL ser uma tecnologia derivada da XML, a *string* foi adaptada para XML, de forma que se pudesse identificar outros

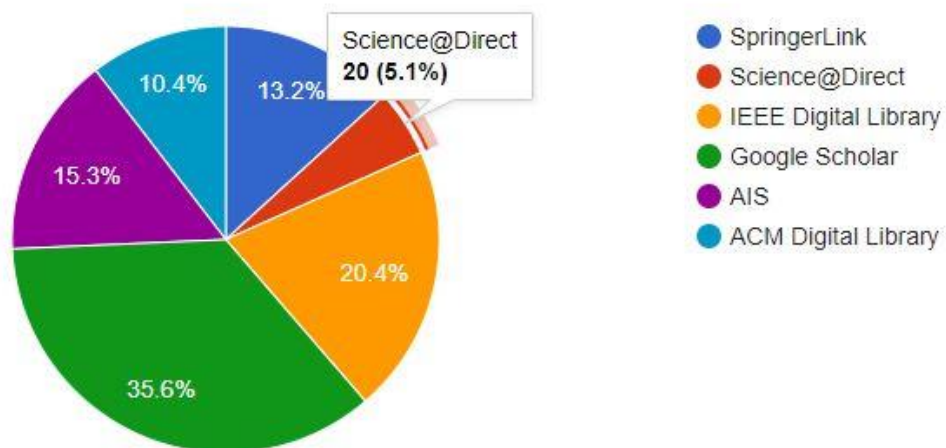
²⁵ <https://www.mendeley.com>

²⁶ <https://parsif.al>

trabalhos relacionados, com foco em XML, que auxiliem na solução das questões de pesquisa. Destarte, pesquisas relacionadas ao mapeamento e integração de dados para XML podem atender as expectativas para o mapeamento de dados para XBRL.

Foram encontrados um total de 393 artigos distribuídos nos repositórios conforme ilustrado na Figura 2, destes artigos, 24 estavam presentes em mais de uma plataforma.

Figura 2 - Distribuição dos 393 artigos encontrados nos repositórios de pesquisa



Por serem anteriores a 2015, 177 artigos foram excluídos, no entanto, tiveram o *abstract* analisado e 188 não possuíam o foco principal relacionado com esta pesquisa. Portanto, restaram 28 artigos para serem analisados e com a seguinte distribuição nos repositórios: SpringerLink com 5 artigos (17,85%), Science Direct com 3 artigos (10,71%), IEEE com 7 artigos (25%), Google Scholar com 7 artigos (25%), AIS com 1 artigo (3,57%) e ACM com 5 artigos (17,85%). Os objetivos e as soluções propostas dos 4 artigos que tratam especificamente da XBRL são apresentados na Tabela 1 e as soluções e propostas dos 24 trabalhos que tratam da XML, JSON e CSV são apresentados na Tabela 2.

Tabela 1 - Objetivos e soluções propostas dos trabalhos relacionados à XBRL

Artigo	Objetivos Principais	Solução Proposta
(Bragança et al., 2019)	Desenvolver uma ferramenta para transformar dados provenientes de arquivos CSV em instância XBRL.	Apresenta uma solução intitulada XBRL-ETL ENGINE que gera as instâncias XBRL em conformidade com a taxonomia do Siconfi.
(Belev, 2019)	Identificar alternativas de armazenamento e validação de instâncias XBRL.	Os autores afirmam que a forma mais eficiente de armazenar as instâncias XBRL é utilizando recursos de SGBD's (Sistemas Gerenciadores de Banco de Dados) que nativamente utilizam internamente o modelo XML, em paralelo com o processamento em memória e o uso de plataformas proprietárias para gerar as instâncias XBRL. Embora os autores tenham afirmado que a melhor solução para armazenar dados provenientes de instâncias XBRL seja em SGBD's XML nativos, a literatura sustenta o contrário [12] [13]. Ademais nenhuma solução é proposta para mapeamento de outras fontes de dados para XBRL.
(Asimadi et al., 2017)	Desenvolver uma abordagem semântica para integrar, processar e consultar as informações financeiras embutidas em instâncias XBRL.	Apresenta uma abordagem para integrar registros financeiros instanciados em XBRL em um formato semanticamente identificável, que permite executar consultas.
(Liu, Etudo, Yoon, 2020)	Desenvolver um método para a recuperação das informações em instâncias XBRL.	Apresenta um método denominado X-IM que permite a interoperabilidade dos dados, mapeando elementos com nomes diferentes, porém semanticamente idênticos em instâncias XBRL.

Trabalhos que tratam da problemática de mapeamento e integração XBRL são escassos, como foi evidenciado na Tabela 1, portanto, se estendeu a investigação para XML.

Tabela 2 - Objetivos e soluções propostas dos trabalhos selecionados que tratam da XML

Artigo	Objetivos Principais	Solução Proposta
(Jayashree; Priya, 2020)	Discute métodos de integração de dados e seus desafios usando ETL ²⁷ .	Foi apresentado um estudo de caso em que dados de um banco em operação foi exportado e validado em XML. O processo foi realizado através de um fluxo ETL de modo a ser aplicado em qualquer ferramenta de BI (Business Intelligence).
(Mao; Ye, 2018)	Analisar as vantagens do uso de XML no intercâmbio de dados, seguindo com a análise hierárquica de XML para representar um esquema relacional de dados. Também objetiva desenvolver um algoritmo de mapeamento que expõe a estrutura de dados do banco de dados relacional preservando a integridade semântica e restrições.	Apresentou uma estrutura conceitual para o mapeamento dos dados contidos em bancos de dados relacionais e dados em instâncias XML, preservando a estrutura do dado, integridade e restrições do esquema relacional.
(Zhu et al., 2017)	Realizar mapeamento de dados em instâncias XML para bancos de dados relacionais.	Foi apresentado uma abordagem intitulada Mini-XML tendo como principal vantagem mapear e armazenar os relacionamentos dos nós XML. Considera mais eficiência em tempo de armazenamento e consumo de espaço em disco quando comparado com a proposta S-XML ²⁸ .
(Song; Haw; Chua, 2019)	Propõe uma análise das vantagens e desvantagens de diferentes abordagens, tais como as com rotulagem de nós ou rotulagem híbrida baseado em modelos como S-XML, XMap, XParent, Mini-XML, para a exportação dos dados de instâncias XML para bancos de dados relacionais.	Foram apontadas limitações e vantagens de cada abordagem de mapeamento, onde foram encontrados dois componentes comuns a todos os mapeamentos, a rotulagem de nós e o mapeamento dos dados. Entre as abordagens analisadas é necessário no mínimo duas tabelas para mapear os dados de instâncias XML para bancos de dados relacionais. Essas duas tabelas armazenam a expressão do caminho e os detalhes do caminho com o dado do nó folha.
(Nassiri; Machkour; Hachimi, 2018)	Propõe uma sintaxe única para consultar dados em estruturas XML, bancos de dados relacionais e estruturas híbridas.	Foi apresentado um tradutor capaz de converter consultas SQL para atender a recuperação de dados em estruturas relacionais, XML e sugere uma estrutura híbrida que permita modelar os dados em

²⁷ ETL é o acrônimo de Extract, Transform e Load (Extração, Transformação e Carga)

²⁸ S-XML é um esquema de mapeamento XML.

Artigo	Objetivos Principais	Solução Proposta
		XML e representá-los em tabelas relacionais, surgindo assim, um SGBD híbrido.
(Chen, 2018)	Busca desenvolver uma estrutura de processamento multimodelo para dados relacionais e XML, e projetar um algoritmo de junção.	Apresenta um algoritmo de consulta intitulado XJOIN que, segundo a proposta, supera em tempo de execução os algoritmos tradicionais de consultas em XML e banco de dados relacionais.
(Gamal; Ahmed; Hefny; El-Moneim, 2016)	Objetiva identificar diferentes abordagens e técnicas que mapeiam esquemas fuzzy em XML para bancos de dados relacionais ou bancos de dados orientados a objetos. Além disso, espera identificar diferentes modelos fuzzy, modelos de dados XML e processo de integração de técnicas fuzzy em diferentes bancos de dados.	Apresenta uma revisão da literatura relacionada a fuzzy em XML, bancos de dados relacionais e bancos de dados orientados a objetos.
(Salem et al., 2017)	Desenvolver um algoritmo ETL que integre dados advindos da WEB em XML com banco de dados relacionais para facilitar a análise por sistemas de DW/BI.	Apresenta uma plataforma capaz de lidar com a integração de dados complexos, e.g. dados semiestruturados, não estruturados, logs de bate-papo, e-mails, imagens, vídeos, originados na WEB em tempo real (real-time).
(Lyamin; Cherepovskay, 2018)	Definir regras de mapeamento para a integração de bancos de dados relacionais e instâncias XML, para fornecer uma representação de dados reduzindo as redundâncias típicas dos modelos hierárquicos.	Apresenta regras para integração de bancos de dados relacionais e instâncias XML superando problemas de redundância das estruturas. Como resultado também foi desenvolvido uma biblioteca para tratar a integração dos dados de XML para bancos de dados relacionais.
(Alami; Bahaj, 2017)	Criar um framework para migração de bancos de dados relacionais para outros tipos de bancos de dados e estruturas hierárquicas, como o XML.	Apresenta um framework para migração de bancos de dados relacionais para outros tipos de bancos de dados com mapeamento semiautomático e utilizando XML.
(Song; Haw, 2020)	Propor uma solução para mapeamento de instâncias XML para bancos de dados relacionais.	É apresentado um mapeamento chamado XML-REG que atua na leitura da instância XML, tratamento dos nós e dados e carga no banco de dados relacional.

Artigo	Objetivos Principais	Solução Proposta
(Qtaish; Ahmad, 2016)	Desenvolver um mapeamento de instâncias XML para bancos de dados relacionais.	Apresenta um mapeamento baseado em dois algoritmos, um de leitura e o outro traduz XPath para consultas SQL, o mapeamento é intitulado XAncestor.
(Nassiri; Machkour; Hachimi, 2017)	Desenvolver uma conexão entre XML e bancos de dados relacionais, recuperando os dados abstraindo as sintaxes das linguagens e os modelos estruturais (XML ou RDB).	Apresenta um sistema que tem por finalidade extrair dados independentemente da linguagem de consulta e do modelo de armazenamento de dados.
(Bikakis; Tsinaraki; Stavarakontakis; Gioldasis; Christodoulakis, 2015)	Desenvolver um framework que traduz consultas SPARQL para XQuery.	Apresenta um framework intitulado SPARQL2XQuery que permite a interoperabilidade entre a Web Semântica e XML.
(Niewerth; Schwentick, 2018)	Desenvolver um framework que identifique as relações entre os dados obtidos em instância XML e os campos das tabelas em bancos de dados relacionais.	Apresenta um framework para o mapeamento entre XML e banco de dados relacionais utilizando um padrão em árvore, onde considera um documento XML como uma tupla, com uma única raiz e todas as arestas estão distantes da raiz. O trabalho se restringe as implicações nas dependências dos relacionamentos no banco de dados, sem considerar valores nulos.
(Maatuk; Ali; Aljawarneh, 2015)	Gerar esquemas XML, baseados em XML Schema, com dados provenientes de bancos de dados relacionais.	Apresenta uma solução para traduzir bancos de dados relacionais em XML Schema validando a estrutura do Schema, a semântica dos dados e restrições de integridade.
(Bai; Yan; Ma; Xu, 2015)	Criar um modelo de dados espaço-temporal fuzzy de XML para bancos de dados relacionais.	Apresenta uma metodologia de modelagem de dados meteorológicos denominado como espaço-temporal fuzzy em XML e a transformação de dados espaço-temporal fuzzy de XML para banco de dados relacionais.
(Petković, 2017a)	Discute os recursos ANSI SQL JSON investigando como diferentes sistemas de banco de dados relacional (SGBD's) os integram.	Apresenta códigos de inserção (<i>insert</i>) e códigos de consulta (<i>select</i>) em formato JSON e conclui que o SGBD Oracle implementa nativamente os conceitos ANSI SQL JSON, o Microsoft SQL Server parcialmente e o PostgreSQL não implementa qualquer conceito ANSI SQL JSON.

Artigo	Objetivos Principais	Solução Proposta
(Habib; Shinnar; Hirzel; Pradel, 2021)	Propõe uma maneira de detectar erros em dados por meio da verificação prévia do esquema JSON.	Oferece uma aplicação que valida os documentos JSON com um esquema JSON. Oferece no repositório github os códigos na linguagem de programação Python.
(Yaghmazade; Wang; Dillig, 2018)	Desenvolver uma abordagem para a integração de estruturas em árvore, como JSON e XML para bancos de dados relacionais.	Apresenta um sistema intitulado MITRA como acrônimo para “Migrating Information from Trees to RelAtions”. É exibido visualmente os passos seguidos pelo algoritmo para efetuar a migração dos dados, no entanto o acesso ao código fonte não está disponível através do link informado no artigo.
(Petković, 2017b)	Descreve vantagens e desvantagens da integração de JSON e Sistemas Gerenciadores de Banco de Dados e sugestões para solucionar os problemas identificados.	Identifica as vantagens no armazenamento do JSON em SGBD’s, tais como: (i) Armazenamento de dados semiestruturados; (ii) Os bancos de dados oferecem custos no gerenciamento reduzido; (iii) Maior produtividade para desenvolvedores. Também identificou que o problema mais relevante é a falta de SGBD’s com capacidade nativa de integração com o JSON.
(Frezza; Mello, 2020)	Desenvolver um padrão (<i>schema</i>) para representar dados geográficos utilizando JSON, que atenda a integração do JSON <i>Schema</i> com SGBD’s NoSQL.	Sugere uma extensão do JSON (<i>schema</i>) para dados espaciais compatíveis com SGBD’s NoSQL, denominado como JS4Geo, o padrão sugerido facilita a integração entre documentos JSON com dados geográficos e SGBD’s NoSQL.
(Doi; Toyama, 2019)	Discute o uso do Framework <i>Remote Table Access</i> (RTA) para disponibilizar na internet tabelas de bancos de dados relacionais para consulta.	Descreve o funcionamento do Framework RTA e para os casos em que os dados são originados em <i>Comma Separated Values</i> (CSV), apresenta a extensão do Framework chamada <i>Table on Top</i> (ToT).

Os trabalhos investigados evidenciam o esforço em superar as dificuldades relacionadas a inserção dos dados provenientes de bancos de dados baseados nos diferentes modelos (Relacional, NoSQL, CSV) em documentos XML, JSON ou XBRL. Na seção seguinte discutiremos os artigos identificados que fazem esses mapeamentos ou integração.

3.3. MAPEAMENTO DE DADOS PARA XBRL

Trabalhos que propõem mapeamento de dados para instâncias XBRL foram identificados a partir desta revisão da literatura. Embora sejam escassos, os trabalhos que tratam de integração e mapeamento de dados, apesar de não lidarem especificamente com a XBRL,

permitiu investigar as linguagens que suportam o padrão XBRL disciplinado pelo consórcio, sejam elas XML, JSON ou arquivos CSV.

Analisando os trabalhos que tratam especificamente da XBRL percebemos que a opinião predominante é a de que as restrições governamentais tornam os dados financeiros complexos. Segundo Belev (Belev, 2019), para a troca de informações nesse domínio, com a validação dos dados e regras, optou-se pelo uso da XBRL. O trabalho de Belev (Belev, 2019) objetiva identificar a maneira com maior eficiência de armazenar os dados financeiro contidos nas instâncias XBRL, sem definir um SGBD específico, o trabalho recomenda o uso de SGBD's que nativamente utilizam internamente o modelo XML sem apresentar de forma convincente a análise que sustentou esta conclusão, contrariando a literatura existente que trata especificamente do desempenho de SGBD's XML. Nos trabalhos (Bragança et al., 2019), (Asimadi et al., 2017) e (Liu et al., 2020) podemos identificar o esforço para o mapeamento dos dados de fontes diversas e instanciá-los em XBRL e a leitura da instância XBRL. Foi alcançado com êxito a geração da instância XBRL no trabalho de Bragança (Bragança et al., 2019) a partir de arquivos CSV, no entanto, não foi desenvolvido no trabalho a validação dos dados, a partir da taxonomia, na geração da instância.

As principais considerações relacionadas ao mapeamento dos dados entre bancos de dados relacionais ou NoSQL para instâncias, sejam elas XML, JSON ou CSV e no sentido oposto, analisaremos nas próximas seções.

3.4. MAPEAMENTO DE DADOS RELACIONAIS OU NoSQL PARA XML

Propostas relacionadas a técnicas para o mapeamento dos dados em bancos de dados relacionais para instâncias XML foram identificadas nos trabalhos (Zhu et al., 2017), (Song; Haw, 2020) e (Qtaish; Ahmad, 2016). Os trabalhos (Zhu et al., 2017) e (Qtaish; Ahmad, 2016) são citados e avaliados no trabalho de Song (Song; Haw, 2020), que define a técnica intitulada Mini-XML, descrita no trabalho de Zhu (Zhu et al., 2017), como uma abordagem que reduz a redundância de dados armazenando o nó folha separadamente da tabela de dados, essa técnica rotula cada nó com $(l, [n, d])$, onde l representa o nível do nó, n o nó pai e d a posição do nó atual (Song et al., 2019). O artigo de Song (Song et al., 2019) ainda faz suas ponderações relacionadas ao trabalho de Qtaish e Ahmad (Qtaish; Ahmad, 2016) que traz a técnica XAncestor que aborda a ancoragem em três componentes, esquema de bancos de dados relacionais, mapeamento XtoDB e algoritmo XtoSQL (Song et al., 2019). A comparação feita entre estas duas técnicas, Mini-XML e XAncestor resulta conclusivo que, XAncestor é capaz de corrigir o excesso de

descrição gerado pelo Mini-XML com relação a posição do nó para a descrição de caminhos do armazenamento feito nas tabelas do banco de dados (Song et al., 2019).

A partir do trabalho de Song (Song et al., 2019), seus autores apresentam posteriormente uma técnica de mapeamento de dados com o nome de XML-REG (Song; Haw, 2020). Os autores fazem uma comparação com o Mini-XML e XAncestor de modo que a técnica XML-REG tem desempenho superiores àquelas outras no que se refere ao processo de armazenamento, processo de recuperação de consultas, tamanho do banco de dados e teste de escalabilidade (Song; Haw, 2020).

O artigo de Bikakis (Bikakis et al., 2015) além de sua contribuição na criação de um framework que traduz consultas SPARQL em consultas XQuery, observa que devido a adoção universal do XML, para que evoluções da WEB prosperem, como é o caso da WoD (Web of Data), é imprescindível a capacidade de leitura/consulta e exportação dos dados em instâncias XML (Bikakis et al., 2015). Nesta perspectiva, observa-se que o trabalho condiciona a evolução da WEB a capacidade de integração das informações representadas de forma semiestruturadas (XML/XBRL) com os mais diversos modelos de dados, esta visão é corroborada no trabalho de Niewerth e Schwentick (Niewerth; Schwentick, 2018).

A utilização de *framework's* foi popularizada por facilitar o desenvolvimento de softwares fornecendo uma solução genérica para uma necessidade específica, isto possibilita ao desenvolvedor uma abstração em seu código. Esta abordagem foi explorada nos trabalhos (Jayashree; Priya, 2020), (Nassiri et al., 2018), (Chen, 2018), (Salem et al., 2017), (Lyamin; Cherepovskaya, 2018), (Alami; Bahaj, 2017), (Nassiri et al., 2017), (Liu et al., 2020), (Bikakis et al., 2015), (Niewerth; Schwentick, 2018), (Maatuk et al., 2015), (Asimadi et al., 2017), (Bai et al., 2015) e (Bragança et al., 2019). Todas estas abordagens se esforçam para criar uma forma genérica para integrar XML em estruturas tradicionais de armazenamento de dados, seja utilizando técnicas de ETL ou propostas inéditas de integração.

No artigo de Bai (Bai et al., 2015) foi apontado que a necessidade de interoperabilidade dos dados em bancos de dados homogêneos e heterogêneos, força o desenvolvimento de formas de interconectar estas bases de dados sem que sejam perdidos meta-dados e restrições, portanto, a forma mais viável encontrada pelo trabalho foi utilizar XML para este fim, mesmo não sendo simples a conversão dos dados provenientes de instâncias XML que possuem a característica de serem ordenados e hierarquizados em formatos relacionais não ordenados.

3.5. MAPEAMENTO DE DADOS RELACIONAIS OU NoSQL PARA JSON

O Java Script Object Notation (JSON) é o principal formato considerado pelos sistemas de gerenciamento de banco de dados NoSQL, pois fornece uma representação de dados flexível adequada para modelar entidades de dados (Frozza; Mello, 2020).

A documentação que trata da XBRL, disponibilizada em seu sítio na internet (www.xbrl.org) contempla a XBRL baseada em JSON, como iniciativa estratégica do consórcio XBRL *International* para simplificar e modernizar a tecnologia XBRL.

A alternativa ao XBRL baseado em XML foi analisada no trabalho “The dilemma of XBRL-XML versus XBRL-JSON regarding linkage of financial information” (Beelitz, 2017). Inicialmente as vantagens relacionadas ao uso do JSON, em substituição ao XML, para XBRL são identificadas, sendo elas: o JSON é um formato de dados que precisa de menos caracteres para descrever itens financeiros; facilidade para ser compreendido por humanos; estrutura semelhante à encontrada em linguagens de programação, tornando fácil e rápido o uso a partir da perspectiva técnica; o JSON é compatível com bancos de dados orientados a documentos (NoSQL) (Beelitz, 2017).

No entanto Beelitz (Beelitz, 2017) afirmou que o uso da XBRL-JSON possui a desvantagem de não ser possível utilizar um arquivo de *schema* contendo a taxonomia, impossibilitando o uso da XBRL em sua plenitude. No entanto a documentação oferecida pelo consorcio que administra a XBRL declara que a XBRL-JSON possui um arquivo chamado *documentinfo* com as informações contextuais²⁹ e o trabalho de Habib (Habib et al., 2021) é conclusivo para refutar esta afirmação, uma vez que está tratando justamente da busca de erros ou inconsistências em arquivos de *schema* JSON, assim refutando a afirmação de Beelitz (Beelitz, 2017).

O trabalho de Frozza e Mello (Frozza; Mello, 2020) afirma que utilizar JSON associado a SGBD's NoSQL proporciona uma facilidade de integração devido o SGBD NoSQL ser orientado a documentos e sua estrutura ter similaridade com documentos JSON.

Já os trabalhos (Petković, 2017a, 2017b) e (Yaghmazadeh et al., 2018) tem o foco em integrar documentos JSON com bancos de dados relacionais. Nestes trabalhos foram tratados conceitos de integração e no caso do trabalho de Petković (Petković, 2017a) alguns códigos foram ensaiados, ou seja, executados em situações controladas para averiguar quais SGBD's

²⁹ <https://www.xbrl.org/guidance/xbrl-json-tutorial/#11-xbrl-json-report-structure>

aceitam códigos escritos utilizando o formato JSON, portanto não sendo uma integração dos dados instanciados em JSON para bancos de dados relacionais e sim a verificação da capacidade do SGBD de interpretar a linguagem JSON para suas operações. Para os trabalhos (Petković, 2017a) e (Yaghmazadeh et al., 2018) o conceito da integração foi tratado e problemas e vantagens foram identificadas sem apresentar uma solução para o problema.

Os trabalhos investigados não apresentaram soluções de integração de JSON com bancos de dados relacionais ou NoSQL.

3.6. MAPEAMENTO DE DADOS RELACIONAIS OU NoSQL PARA CSV

A perspectiva de integração de arquivos CSV com bancos de dados relacionais ou NoSQL ou no sentido contrário, tem literatura escassa nos repositórios de pesquisa, apenas dois trabalhos foram identificados. No entanto um dos trabalhos (Bragança et al., 2019) trata da instanciação de dados proveniente de arquivos CSV para instâncias XBRL. Neste caso a necessidade do uso de CSV é dada devido à baixa interoperabilidade do SGBD Adabas³⁰ que possibilita a exportação dos dados apenas em arquivos CSV, sem a possibilidade de conexão com qualquer SGBD, devido ao elevado custo financeiro para aquisição dos componentes (*softwares*) que permitam a conexão.

O segundo trabalho de Doi e Toyama (Doi; Toyama, 2019) tem o seu foco voltado para a disponibilidade de dados na internet para atender o projeto de dados abertos, neste caso o CSV também é usado como fonte de dados para o framework RTA.

O uso de arquivos CSV pode ter um impulso após a disponibilização da XBRL também em CSV, no entanto, pesquisas relacionadas a integração de CSV com bancos de dados relacionais ou NoSQL ou até mesmo do próprio XBRL-CSV não foram encontradas nos repositórios conhecidos.

3.7. RESULTADOS RERERENTES ÀS QUESTÕES DE PESQUISA DA REVISÃO SISTEMÁTICA DA LITERATURA

As questões da pesquisa utilizadas nesta revisão sistemática da literatura foram parcialmente satisfeitas. A seguir discute-se como os trabalhos investigados satisfizeram as questões propostas neste trabalho.

³⁰ https://www.softwareag.com/en_corporate/platform/adabas-natural.html

(Q1) Existem soluções de integração ou mapeamento de dados para XBRL e vice-versa? A investigação desenvolvida neste trabalho identificou apenas dois trabalhos (Bragança et al., 2019) e (Liu et al., 2020) que atendem a questão. É salutar observar que as propostas encontradas na pesquisa, terem o seu foco em solucionar problemas específicos como atender a legislação ou identificar similaridade semântica em instâncias XBRL, isto posto, podemos ainda perceber que os trabalhos se limitam a tratar de um caminho único da informação, sendo majoritariamente no sentido de gerar a instância XBRL. Devido à escassez de trabalhos que tratam da XBRL e ao consórcio responsável pela XBRL admitir as linguagens XML, JSON e arquivos CSV, optou-se por investigar mapeamentos relacionados a XML, JSON e CSV na busca de trabalhos que satisfizessem essa questão. Nesta perspectiva identificamos os trabalhos (Jayashree; Priya, 2020), (Zhu et al., 2017), (Song et al., 2019), (Nassiri et al., 2018), (Chen, 2018), (Salem et al., 2017), (Lyamin; Cherepovskaya, 2018), (Alami; Bahaj, 2017), (Song; Haw, 2020), (Qtaish; Ahmad, 2016), (Nassiri et al., 2017), (Liu et al., 2020), (Bikakis et al., 2015), (Niewerth; Schwentick, 2018), (Maatuk et al., 2015), (Asimadi et al., 2017), (Bai et al., 2015), (Petković, 2017a, 2017b), (Yaghmazadeh et al., 2018) e (Doi; Toyama, 2019) que tratam do mapeamento dos dados de instâncias XML ou JSON e arquivos CSV para bancos de dados.

Portanto existem trabalhos que tratam do mapeamento dos dados da XBRL e mapeamentos relacionados as linguagens admitidas pelo consorcio responsável pela XBRL.

(Q1.1) Qual a necessidade de mapear os dados de fontes diversas para XBRL e vice-versa?

No Brasil o fator preponderante para o uso da XBRL é a imposição legal, este regramento é evidenciado por Belev em seu trabalho (Belev, 2019). No trabalho de Bragança (Bragança et al., 2019) o desenvolvimento da solução foi criado especificamente para atender a Instrução Normativa 896/17 da Secretaria do Tesouro Nacional (STN).

As empresas brasileiras que pretendem negociar os seus papéis nas bolsas de valores norte americanas, precisam atender a legislação da *US Securities and Exchange Commission*³¹ (US-SEC) que determina a entrega das informações financeiras e contábeis no formato XBRL.

Portanto as determinações legais dos governos impõem a necessidades aos entes federados e as pessoas jurídicas quanto ao uso da XBRL, tornando imprescindível a recuperação e exportação dos dados em *eXtensible Business Reporting Language*.

³¹ <https://www.sec.gov/>

(Q1.2) Quais tecnologias são usadas para mapear os dados XBRL?

Nos trabalhos investigados a discussão sobre as tecnologias utilizadas foi feita superficialmente, a maior parte dos trabalhos não citou ou fez qualquer menção a respeito das tecnologias usadas. Apenas no trabalho de Bragança (Bragança et al., 2019) que o uso da ferramenta de ETL do Pentaho foi comentado. Quando investigado as linguagens XML e JSON, foi possível perceber o uso de JDBC para a conexão do SGBD no trabalho de Mao e Ye (Mao; Ye, 2018) e o uso do SGBD *SQL Server* da *Microsoft* no trabalho de Song (Song et al., 2019). No trabalho de Bikakis (Bikakis et al., 2015) identificamos o uso de SPARQL e XQuery devido ser o uso destas tecnologias o objetivo do trabalho.

(Q1.3) As propostas existentes são eficientes para abarcar qualquer tipo de fonte de dados e XBRL?

Somente uma única proposta atendeu esse quesito, isto para atender especificamente uma imposição legal (Bragança et al., 2019), este trabalho utiliza arquivos CSV como fonte de dados. Quando relacionado a linguagem JSON juntamente com arquivos CSV, também investigados neste trabalho, não foram localizadas soluções que tivessem como objetivo instanciar os dados em XBRL-JSON ou XBRL-CSV.

3.8. SOLUÇÕES PROPRIETARIAS

Conforme observado na seção 3.7, as imposições legais são os maiores impulsionadores da tecnologia XBRL. Essas imposições legais suscitaram a necessidade de softwares capazes de gerar instâncias XBRL e assim surgiram iniciativas comerciais como Amelkis³², aSIS³³ e Vizor³⁴ que se propõem a suprir essa necessidade de gerar instâncias XBRL para atender a legislação vigente.

A aplicação Amelkis oferece uma solução ERP e dentre as funcionalidades oferecidas está a geração de instâncias XBRL, utilizando taxonomias criadas na própria ferramenta ou importando taxonomias geradas em outras ferramentas. Já a aSIS é um software desenvolvido para atender o setor bancário, está dividido em módulos de modo a ser utilizado como ETL para a instanciação dos dados em XBRL. A ferramenta está preparada, segundo seu fabricante, para atender as especificações legais da União Europeia (UE). Outro software que se propõe a atuar

³² <https://www.amelkis-solutions.com/>

³³ <https://asist-xbrl.eu/>

³⁴ <https://www.vizorsoftware.com/>

na gestão empresarial é o Vizor Software, a solução ERP, a partir dos dados contidos em suas bases de dados gera instâncias XBRL. Oferece também um módulo de validação e exibição dos dados contidos em instâncias XBRL.

O software Arelle³⁵ é uma aplicação *open-source* que objetiva a visualização de instâncias XBRL, no entanto para a leitura de instâncias XBRL, faz-se necessário o desenvolvimento de plug-in. Os desenvolvedores do software afirmam que o Arelle possui a capacidade de gerar instâncias XBRL, no entanto não oferece nativamente conexão com fontes de dados.

Nenhum dos softwares publicou os valores cobrados por suas subscrições. No entanto, todos os softwares descritos anteriormente são homologados pelo consorcio responsável pela tecnologia XBRL. Existem outras soluções homologadas pelo xbrl.org, contudo, não são soluções que atendem a expectativa dessa pesquisa que busca um ETL *open-source* capaz gerar instâncias XBRL.

3.9. CONSIDERAÇÕES FINAIS

Neste capítulo, trabalhos relacionados foram apresentados. Inicialmente, mostrou-se o estado da arte do mapeamento de dados em bancos de dados para instâncias XBRL ou XML. Em seguida, foi discutido as diferentes possibilidades de mapeamento sendo de bancos de dados relacionais para XBRL, XML, JSON ou CSV. Posteriormente, foram descritos os estudos frente as questões da pesquisa, que comparou todos os trabalhos analisados.

Conforme apresentado no Capítulo 1, a XBRL é uma tecnologia que se tornou padrão internacional para o intercâmbio de dados financeiros. A geração de instâncias XBRL utilizando como fonte os repositórios de dados financeiros e contábeis não é trivial (Asimadi et al., 2017). Conforme visto na Tabela 1 e 2, a maioria dos trabalhos mostram que a preocupação primaria está relacionada com a leitura das instâncias XML, mapeamento dos dados e inserção em um banco de dados relacional, já com as instâncias XBRL apenas um trabalho apresentou solução para gerá-la.

³⁵ <https://arelle.org/arelle/>

Além disso, identificou-se que nos trabalhos investigados, a atenção está voltada majoritariamente para os bancos de dados relacionais, não sendo abundante a análise com a perspectiva dos SGBD's NoSQL. Tendo em vista que os dados provenientes de instâncias JSON são comumente armazenados em bancos de dados NoSQL (Bahta; Atay, 2019). Também foram investigados os softwares homologados pelo consorcio xbrl.org em busca de algum que fosse similar ao investigado nessa pesquisa, e não foi identificado uma solução gratuita capaz de atender a conexão com bases de dados diversas e gerar instâncias XBRL. Desta forma, a proposta de uma ferramenta capaz de conectar as mais diversas fontes de dados, vincule os dados à taxonomia escolhida e gere a instância XBRL, guiou o desenvolvimento desta dissertação.

4. FERRAMENTA XBRL PROCESSOR

Iniciativas para suprir a ausência de soluções que tenham a capacidade de gerar instâncias XBRL podem ser observadas no trabalho de Bragança, propondo uma ferramenta ETL (Bragança et al., 2019), no entanto, a solução proposta coleta apenas dados de arquivos CSV, restringindo a capacidade de adoção da solução. Pode-se perceber ainda no trabalho de Bragança que existe uma carência nos poderes executivos do Brasil, seja na esfera federal, estadual ou municipal, por soluções que atendam a demanda na geração das instâncias XBRL a serem entregues ao Siconfi (Bragança et al., 2019).

A revisão sistemática da literatura apresentada no Capítulo 3, identificou que existe uma preocupação primária em desenvolver softwares que sejam capazes de ler e exibir os relatórios contidos nas instâncias XBRL, no entanto, não existe essa mesma preocupação em gerar instâncias XBRL. Partindo dessa constatação, desenvolveu-se neste trabalho uma ferramenta voltada para a geração das instâncias XBRL de forma a preencher essa lacuna.

Esta seção descreve o desenvolvimento da ferramenta para leitura, transformação e carga de dados (ETL) para a tecnologia XBRL, intitulada como XBRL Processor. O principal objetivo do desenvolvimento desta solução é permitir a extração de dados de diferentes sistemas de armazenamento (SGBD's relacionais, NoSQL, CSV), reutilizando as parametrizações, e carregá-los em instâncias XBRL, definidas por uma taxonomia. Deste modo, a codificação da ferramenta, descrita na Seção 4.2, mostra que o conceito proposto na Seção 4.1 é exequível. Para realizar o desenvolvimento da ferramenta utilizou-se alguns critérios:

- A utilização da MSC (Matriz de Saldos Contábeis) no formato e taxonomia XBRL definidos pelo Siconfi (Sistema de Informações Contábeis e Fiscais do Setor Público Brasileiro) para que a aplicação possa ser validada e esteja pronta para solucionar o problema da entrega da MSC no formato XBRL ao Siconfi/STN, uma exigência feita aos entes federados do Brasil;
- O desenvolvimento da ferramenta proposta foi baseado na linguagem TypeScript (Bogner; Merkel, 2022) em razão da facilidade de compreensão do código, de possuir performance de execução, disponibilidade de biblioteca gratuita e editável para leitura e geração de instâncias XML;

- Por fim, utilizou-se como fonte de dados os SGBD's MySQL (Christudas, 2019), Sqlite3 (Bin et al., 2022) e MongoDB (Győrödi et al., 2022), com os quais foi possível aferir a conexão da ferramenta com SGBD's relacionais e NoSQL, seguindo o princípio de utilização de fontes de dados de diferentes formatos apresentado na Seção 4.1.

O desenvolvimento da ferramenta XBRL Processor consistiu em três etapas:

- Na primeira etapa, construiu-se a ferramenta ETL para XBRL baseada na taxonomia XBRL, versão 2022, definida para a Matriz de Saldos Contábeis do Siconfi. A fonte de dados utilizada foi o SGBD Sqlite3 com dados fictícios e anonimizados, no entanto coerentes, fornecidos pela Secretaria de Finanças do Estado de Rondônia (SEFIN - RO);

- Na segunda etapa novos SGBD's foram incorporados no projeto do código da ferramenta. Desta forma, foi possível verificar a adaptabilidade do código a diferentes formatos e fontes de dados. Foram adicionados conectores para os SGBD's MySQL e MongoDB;

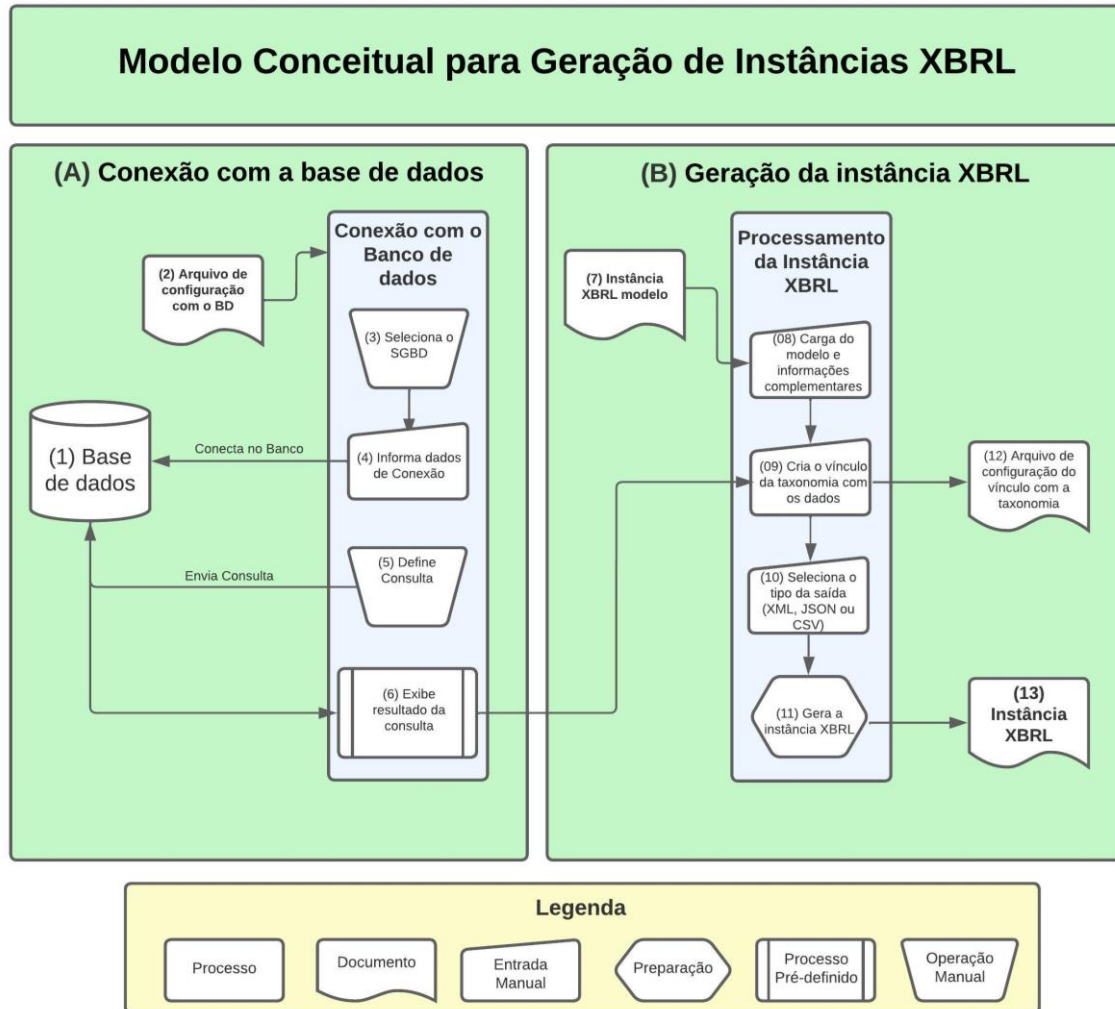
- Na terceira etapa, comparou-se a instância XBRL gerada pelo XBRL Processor com a fornecida pela Secretaria de Finanças do Estado de Rondônia. Por fim, a instância XBRL gerada com os dados fictícios foram validadas pela STN no Siconfi e pela Secretaria de Finanças do Estado de Rondônia nos ambientes de testes e de produção respectivamente.

4.1. MODELO CONCEITUAL DA XBRL PROCESSOR

O modelo proposto, representado na Figura 3, tem a finalidade de conectar bases de dados relacionais ou NoSQL e possibilitar a junção dos dados extraídos dessas bases de dados com a taxonomia a ser utilizada gerando a instância XBRL.

A proposta do processo de geração de instâncias XBRL foi dividida em duas etapas: (A) conexão com a base de dados e (B) geração da instância XBRL a partir da taxonomia informada.

Figura 3 - Modelo conceitual para geração de instâncias XBRL



4.2. CONEXÃO COM BANCOS DE DADOS

Dividir a solução em duas etapas permitirá uma melhor manutenção no código e desenvolvimento de possíveis melhorias, com isso cada etapa possui o seu contexto funcional, aumentando a coesão e diminuindo o acoplamento. Em softwares a modularização permite que uma mudança em um componente tenha impacto mínimo em outros (Lazzari; Farias, 2022).

O processo de conexão com o banco de dados inicia com a escolha do SGBD, como podemos observar na Figura 3(A-3), a lista dos SGBD's suportados será limitada apenas pelos *drivers* disponíveis, no entanto, pode ser adicionado quaisquer *driver*, e.g. driver para

PostgreSQL, Oracle, MySQL, SQL Server, MongoDB, ou outro que seja necessário para a extração de dados para a geração da instância XBRL.

Na sequência Figura 3(A-4), os dados de conexão são solicitados de modo a permitir a conexão com o SGBD. Dados como porta TCP, login e senha são informações indispensáveis para a conexão. Quando os dados de conexão são informados, um teste de conexão é executado.

Após completado o processo de conexão, se inicia a execução da inserção do código da consulta Figura 3(A-5), e.g. SQL ou o código pertinente ao SGBD selecionado nas atividades anteriores. A possibilidade de inserir o código direto na aplicação, permite uma flexibilidade na consulta dos dados de modo a facilitar ajustes e adição de colunas com dados estáticos, e.g. a informação relacionada aos valores financeiros serem em moeda brasileira (BRL). Após a execução do código, pode-se verificar na Figura 3(A-6), que a atividade seguinte é a exibição do resultado da consulta para conferência.

A conexão com o banco de dados foi planejada de modo que todos os dados e parametrizações possam ser reaproveitados em uma posterior execução, para isso o arquivo com as configurações é reaproveitado nas execuções subsequentes Figura 3(A-2).

4.2.1. GERAÇÃO DA INSTÂNCIA XBRL

O processo de vincular a taxonomia aos dados não é um processo ordinário, Asimad (Asimadi et al., 2017) apontou em seu trabalho que o processo de integração da XBRL permanece complexo, portanto, o início do processamento da instância XBRL, Figura 3(B-7), traz a possibilidade de usar uma instância XBRL modelo para recuperar o maior número possível de vínculos entre a taxonomia e os dados. A instância modelo pode ser fornecida pelo criador da taxonomia para o qual os dados serão enviados.

No passo seguinte, dados complementares podem ser adicionados Figura 3(B-8), entende-se como informações complementares aquelas que são exclusivas da organização que está gerando a instância XBRL, e.g. código de cadastro da organização junto ao órgão que receberá os dados.

Após os primeiros ajustes na inserção da taxonomia modelo e informações complementares, cria-se o vínculo dos dados recebidos do banco de dados com a taxonomia Figura 3(B-9), informações que são utilizadas repetidas vezes podem ser preservadas, se necessário para posterior recuperação, em arquivo de configuração pertinente, Figura 3(B-12) (Dimou et al., 2014).

O consórcio XBRL após o ano de 2021 definiu novas formas de utilizar a tecnologia, definindo como recomendação além da XBRL estendida da XML, a XBRL estendida do JSON e de arquivos CSV (Cerqueira; Silva, 2016). O XBRL em JSON e CSV são limitados a instância não sendo admitido nesses formatos o arquivo contendo a taxonomia. Também, não foi esclarecido pelo consórcio responsável pela XBRL como serão as instâncias XBRL-JSON e XBRL-CSV quando geradas a partir da taxonomia GL³⁶. No entanto, para estar alinhado com as recomendações do consórcio XBRL é preciso permitir a possibilidade de gerar a instância nos três formatos admitidos pelo consorcio XBRL. Essa previsão existe nesta etapa do processo, como pode ser observado na Figura 3(B-10).

Por fim, é executado o processamento dos dados para gerar a instância XBRL, Figura 3(B-11), tendo como produto a instância com todos os dados e seus vínculos com a taxonomia, bem como os dados complementares de unidade e referências através dos links, Figura 3(B-13).

4.3. IMPLEMENTAÇÃO

O desenvolvimento da ferramenta XBRL Processor utilizou o conceito proposto na Seção 4.1, o qual propõe uma solução para o problema latente identificado no Capítulo 3 para a geração de instâncias XBRL a partir de SGBD's, desse modo, a codificação foi executada de modo a permitir a validação externa das instâncias geradas pela aplicação.

4.3.1. ASPECTOS DE IMPLEMENTAÇÃO

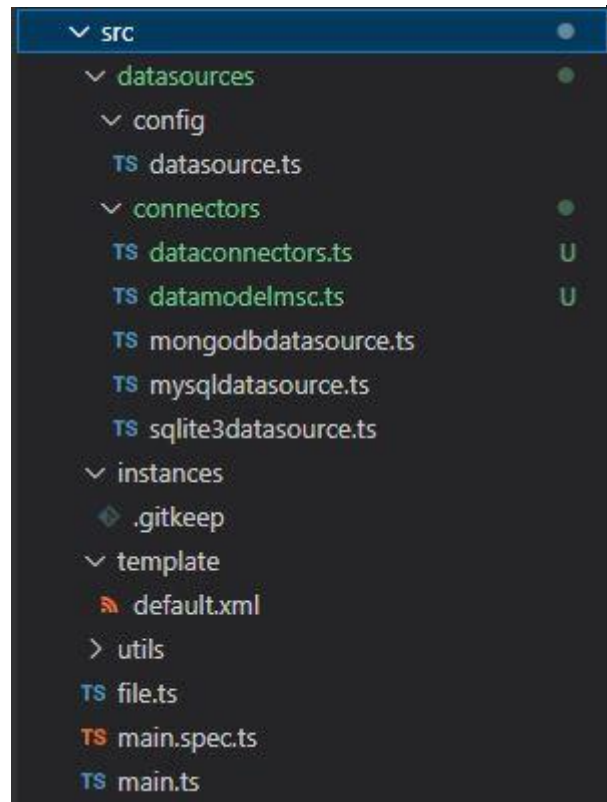
A organização do código fonte se deu com foco na arquitetura limpa (Sommerville, 2011) de forma a facilitar o entendimento do código, recursos e funções. Assim, a arquitetura limpa possibilita a evolução da aplicação sem o consumo excessivo de recursos humanos ou de tempo. O fato de separar o projeto em camadas independentes, propicia que alterações em uma camada não interfiram nas demais (Sommerville, 2011). Dessa maneira, espera-se que o código possa ser estendido ou incorporado a aplicações financeiras e contábeis. O código fonte pode ser acessado em: <https://github.com/splhead/sxbrl.git>.

Seguindo o conceito definido na Seção 4.1, o código está dividido em dois componentes, no qual o primeiro componente definido na Figura 3(A), que trata da conexão com a fonte de

³⁶ <http://www.xbrl.org/int/gl/2016-12-01/gl-framework-2017-PWD-2016-12-01.html>

dados, pode-se relacionar com a pasta `src/datasource` do código, sendo subdividida em `src/datasource/config` e `src/datasource/connectors`. O segundo componente exibido na Figura 3(B), que trata da geração da instância XBRL, foi integrado ao arquivo `src/main.ts`. A organização do código pode ser observada na Figura 4, que expõe a estrutura de pastas utilizada para organizar o código.

Figura 4 - Estrutura de pastas do código

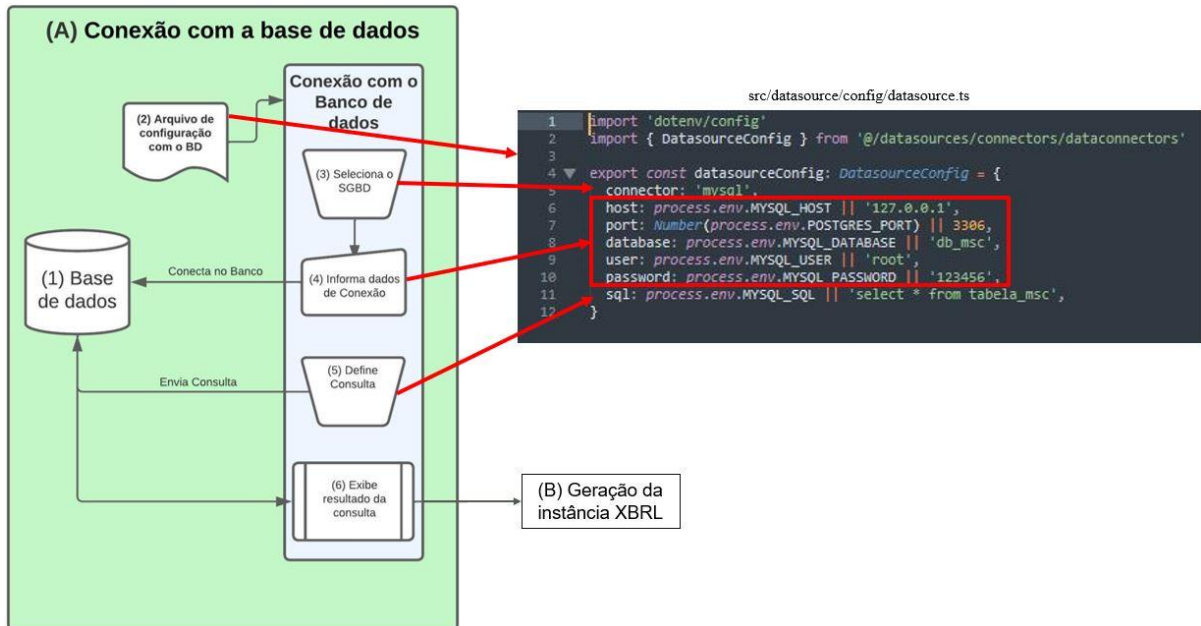


A fonte de dados foi separada em: os códigos relacionados à conexão com o SGBD, configurações da fonte de dados (SGBD) e a estrutura da MSC esperada na consulta.

Os arquivos contendo o código relacionado a conexão, tais como, portas TCP, login, podem ser acessados no arquivo `src/datasource/config/datasource.ts`. Esse arquivo é representado conceitualmente na Figura 3(A-4) e pode ter seu código para o SGBD MySQL observado na Figura 5.

O arquivo definido na Figura 3 (A-2) como sendo os dados de conexão com a base de dados a serem reaproveitados pela ferramenta, não foi gerado separadamente dos arquivos do código, sendo mantido no arquivo `src/datasource/config/datasource.ts`, Figura 5, de modo a atender a expectativa do conceito de reaproveitar as parametrizações.

Figura 5 - Dados de conexão com o SGBD MySQL



Na Figura 3(A-5) é definido que um teste de conexão com o banco de dados a partir dos dados informados deve ser executado, no entanto, o teste de conexão é efetuado no momento que a aplicação é iniciada. Também na Figura 3(A-6) o conceito determina que os dados retornados da consulta precisam ser exibidos, porém a exibição dos dados, conforme o modelo, não foi atendido.

Cada fonte de dados possui um código para a leitura dos dados armazenados em sua estrutura. Para acomodar esse código foi criada a pasta `src/datasources/connectors/`, onde cada fonte de dados possui, separadamente, o seu respectivo código. As configurações para as fontes de dados SQLite3 (`sqlite3datasource.ts`), MySQL (`mysqldatasource.ts`) e MongoDB (`mongodbdatasource.ts`) estão disponíveis, para novas fontes de dados e.g. SGBD PostgreSQL, novos arquivos com os códigos referente a configuração, serão necessários. Os códigos referentes aos SGBD's disponíveis podem ser observados nas Figuras 6, 7 e 8.

Figura 6 - Implementação do SGBD MongoDB

```

1  import { DataModelMSC, Datasource } from './datamodelmsc'
2  import { MongoClient } from 'mongodb'
3  import { datasourceConfig } from '../config/datasource'
4  import { assertConnectorType } from '@utils/functions'
5  import { MongodbConnector } from './dataconnectors'
6
7  /**
8   * Area destinada as implementações gerais da fonte de dados MongoDB.
9   */
10
11  class MongoDBDatasource implements Datasource {
12    private client: MongoClient | null = null
13
14    private async connect() {
15      assertConnectorType<MongodbConnector>(datasourceConfig, 'mongodb')
16      this.client = new MongoClient(datasourceConfig.connection_string)
17      console.Log('Conectado ao MongoDB')
18    }
19
20    async getData(): Promise<DataModelMSC[]> {
21      try {
22        assertConnectorType<MongodbConnector>(datasourceConfig, 'mongodb')
23
24        await this.connect()
25
26        const database = this.client?.db(datasourceConfig.database)
27        const collection = database?.collection(datasourceConfig.collection)
28
29        const rows = (await collection
30          ?.find({}))
31          .toArray() as unknown as DataModelMSC[]
32
33        console.Log('Lido os dados do Mongodb')
34
35        return rows
36      } catch (error) {
37        console.error('Erro ao ler dados no Mongodb ${error}')
38      }
39      const data = {} as DataModelMSC
40      return [data]
41    }
42
43    async close() {
44      await this.client?.close()
45      console.Log('Conexão com Mongodb encerrada')
46    }
47  }
48  export { MongoDBDatasource }

```

Figura 7 - Implementação do SGBD SQLite3

```

1  import sqlite3, { Database } from 'sqlite3'
2  import path from 'path'
3  import { DataModelMSC, Datasource } from './datamodelmsc'
4  import { datasourceConfig } from '../config/datasource'
5  import { assertConnectorType } from '@utils/functions'
6  import { SqliteConnector } from './dataconnectors'
7  /**
8   * Area destinada as implementações gerais da fonte de dados SQLITE.
9   */
10 class Sqlite3Datasource implements Datasource {
11   private database: Database | null = null
12   private async connect() {
13     return new Promise((resolve, reject) => {
14       assertConnectorType<SqliteConnector>(datasourceConfig, 'sqlite')
15       const absolutePath = path.resolve(
16         __dirname,
17         '..',
18         datasourceConfig.databaseFileName,
19       )
20       this.database = new sqlite3.Database(
21         absolutePath,
22         sqlite3.OPEN_READWRITE,
23         (err) => {
24           if (err) reject(err)
25           resolve(
26             console.Log('Conectou com a fonte de dados sqlite ${absolutePath}'),
27           ) },
28       )
29     }) }
30   async getData(sql?: string): Promise<DataModelMSC[]> {
31     await this.connect()
32     return new Promise((resolve, reject) => {
33       if (!sql) reject(new Error('SQL não definido'))
34       else {
35         this.database?.all(
36           sql,
37           [],
38           (error: Error | null, rows: DataModelMSC[]) => {
39             if (error) reject(error)
40             resolve(rows)
41           },
42         )
43       }
44     }) }
45   close() {
46     console.Log('Encerrou a conexão com a fonte de dados sqlite')
47     this.database?.close()
48   } }
49 export { Sqlite3Datasource }

```

Figura 8 - Implementação do SGBD MySQL

```

1  import mysql from 'mysql2'
2  import { DataModelMSC, Datasource } from './datamodelmsc'
3  import { datasourceConfig } from '../config/datasource'
4  import { MysqlConnector } from './dataconnectors'
5  import { assertConnectorType } from '@/utils/functions'
6  /**
7   * Area destinada as implementações gerais da fonte de dados MYSQL.
8   */
9  class MysqlDatasource implements Datasource {
10     private connection: mysql.Connection | null = null
11     private connect() {
12         assertConnectorType<MysqlConnector>(datasourceConfig, 'mysql')
13         this.connection = mysql.createConnection({
14             host: datasourceConfig.host,
15             port: datasourceConfig.port,
16             user: datasourceConfig.user,
17             password: datasourceConfig.password,
18             database: datasourceConfig.database,
19         })
20         console.Log('Conectado com mysql')
21     }
22     async getData(sql?: string): Promise<DataModelMSC[]> {
23         return new Promise((resolve, reject) => {
24             if (!sql) reject(new Error('SQL não definido'))
25             else {
26                 this.connect()
27                 console.Log('obtendo dados do mysql')
28                 this.connection?.query(sql, (error: Error, rows: DataModelMSC[]) => {
29                     if (error) {
30                         console.Log('Erro ao obter dados no mysql')
31                         reject(error)
32                     }
33                     resolve(rows)
34                 })
35             }
36         })
37     }
38     close() {
39         console.Log('Encerrando a conexão com mysql')
40         this.connection?.end()
41     }
42 }
43 export { MysqlDatasource }
44

```

Manteve-se a estrutura proposta pelo Siconfi para a MSC, onde os campos seguem as definições do *template*³⁷ definido no manual do Siconfi. O layout da MSC foi planejado para melhor representar as informações com base no padrão XBRL (STN, 2017). Embora a ferramenta não faça a validação dos dados na taxonomia, a consulta precisa retornar os dados seguindo a estrutura determinada pelo Siconfi. Pode-se observar a estrutura definida pelo Siconfi codificada na Figura 9 nas linhas 1 a 18.

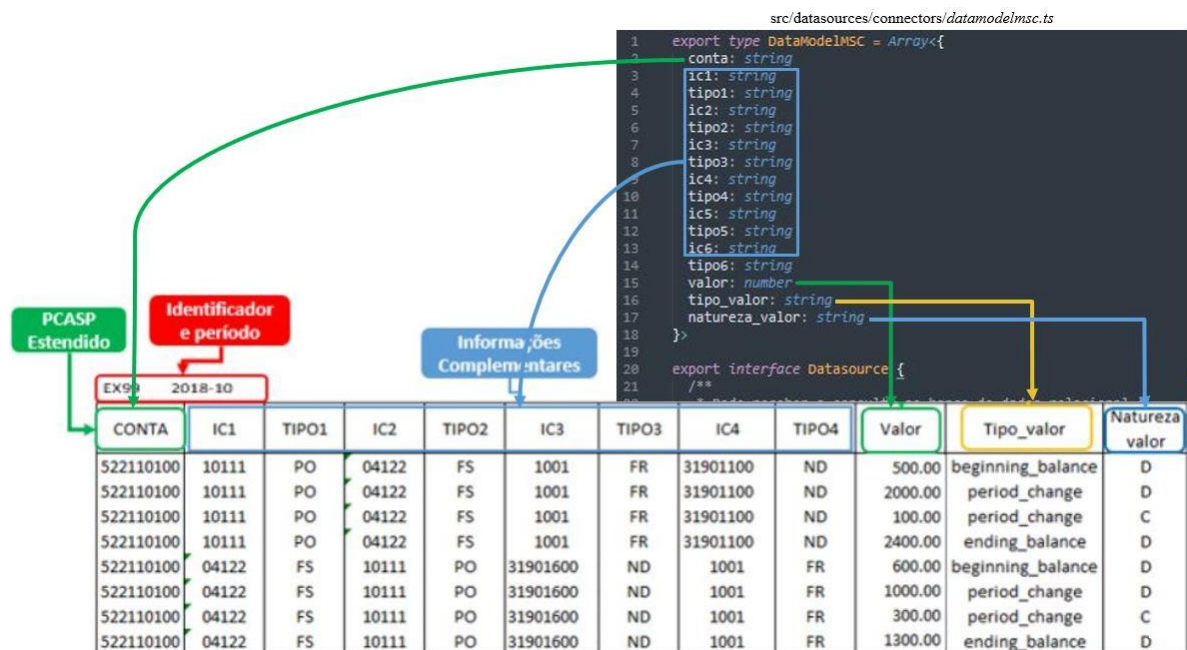
A diversidade de SGBD's utilizados na aplicação, facilitará a adequação a qualquer ambiente. Na Figura 10 é ilustrado o diagrama de sequência que descreve a conexão com o

³⁷ <https://Siconf.tesouro.gov.br/Siconf/pages/public/conteudo/conteudo.jsf?id=12503>

SGBD no primeiro módulo da ferramenta XBRL Processor. Espera-se que, aplicado em um ambiente de produção, o SGBD utilizado pela organização seja selecionado e parametrizado, eliminando a necessidade de troca de SGBD e, por conseguinte, a modificação do código.

O segundo módulo, conforme descrito na Figura 3(B), executa o carregamento dos dados, carrega o modelo da instância, a taxonomia e procede com a geração da instância XBRL da MSC.

Figura 9 - Estrutura da MSC definida pelo Siconfi

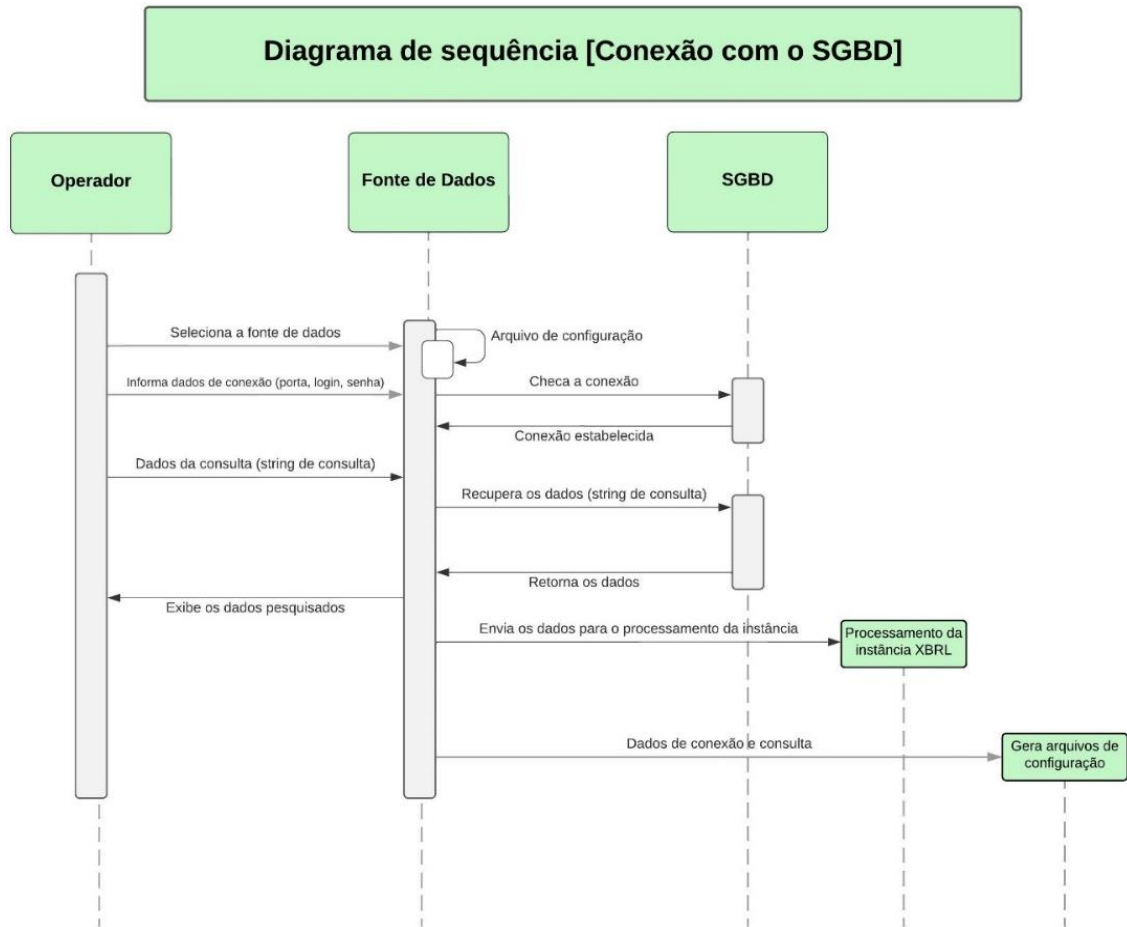


O modelo da instância contido na pasta src/template, conterá o cabeçalho, o código da unidade fornecido pelo Siconfi, a referência a moeda brasileira e os períodos para o qual a MSC se refere, e.g. 2023-01-31.

Quando necessário, as atualizações desses dados devem ser feitas no arquivo modelo. Os demais dados (e.g. contexto, conta contábil etc.) contidos na instância modelo, são descartados pela ferramenta. Na Figura 11 pode-se verificar os dados carregados pela aplicação provenientes do arquivo modelo.

O segundo módulo da aplicação está concentrado no arquivo src/main.ts. Podemos observar as dependências dos componentes e suas referências ao código na Figura 12.

Figura 10 - Diagrama de sequência com o sistema gerenciador de banco de dados



A ferramenta utiliza a biblioteca `fast-xml-parser`³⁸ para a leitura e escrita de documentos XML. Devido as restrições da biblioteca `fast-xml-parser`, parte da taxonomia foi adicionada ao código para possibilitar a vinculação dos dados à taxonomia. A biblioteca `fast-xml-parser` é aberta e permite a evolução do código, no entanto, a edição da biblioteca para atender a leitura da taxonomia, sem que elementos da taxonomia estejam presentes no código, não foi contemplada, portanto o arquivo definido na Figura 3(B-12) será gerado em trabalhos futuros. Um fragmento do código no qual é tratada a taxonomia utilizando a biblioteca `fast-xml-parser` presente no código, pode ser observado na Figura 13.

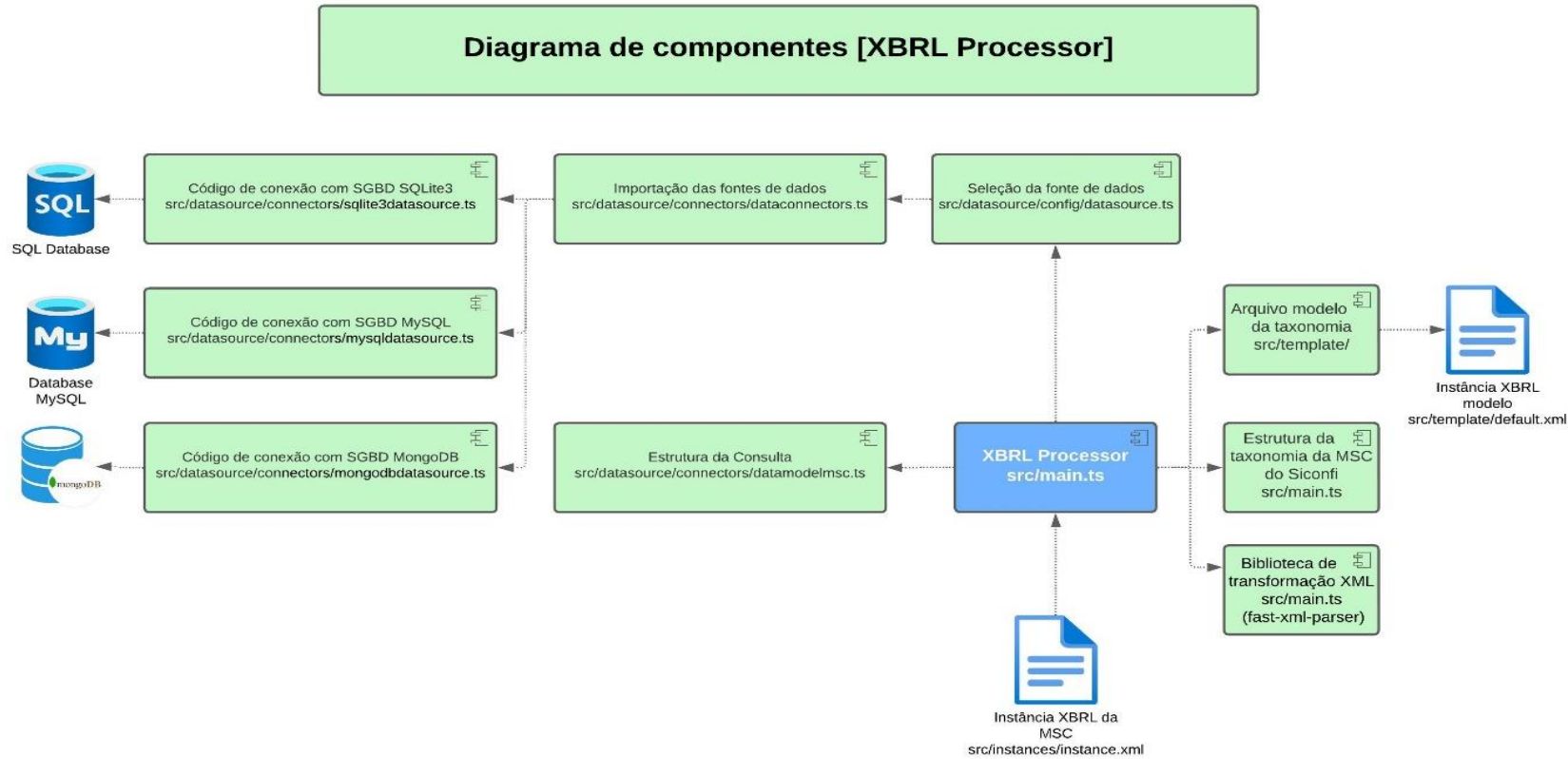
As últimas linhas do arquivo `main`, possui o código que gera a instância XBRL da MSC com os dados provenientes da fonte de dados escolhida. A instância é gerada e carregada na pasta `src/instances` com o nome `instance.xml`.

³⁸ <https://github.com/NaturalIntelligence/fast-xml-parser>

Figura 11 - Instância XBRL modelo da MSC



Figura 12 - Diagrama de componentes da ferramenta XBRL Processor



Para auxiliar na configuração e utilização da aplicação, um arquivo *README*³⁹ foi adicionado aos arquivos da ferramenta com o resumo das atividades a serem seguidas e configurações necessárias para o funcionamento do XBRL Processor.

```

62
63     const accountSubs = state.map(({ subTypeText, subId }) => {
64         return {
65             'gl-cor:accountSubID': {
66                 '#text': subId,
67                 '@_contextRef': 'C1',
68             },
69             'gl-cor:accountSubType': {
70                 '#text': subTypeText,
71                 '@_contextRef': 'C1',
72             },
73         }
74     })
75     const result = {
76         'gl-cor:lineNumberCounter': {
77             '#text': index + 1,
78             '@_contextRef': 'C1',
79             '@_decimals': '0',
80             '@_unitRef': 'u',
81         },
82         'gl-cor:account': {
83             'gl-cor:accountMainID': {
84                 '#text': item.conta,
85                 '@_contextRef': 'C1',
86             },
87             'gl-cor:accountSub': accountSubs,
88         },
89         'gl-cor:amount': {
90             '#text': item.valor,
91             '@_contextRef': 'C1',
92             '@_decimals': '2',
93             '@_unitRef': 'BRL',
94         },
95         'gl-cor:debitCreditCode': {
96             '#text': item.natureza_valor,
97             '@_contextRef': 'C1',
98         },
99         'gl-cor:xbrlInfo': {
100             'gl-cor:xbrlInclude': {
101                 '#text': item.tipo_valor,
102                 '@_contextRef': 'C1',
103             },
104         },
105     }
106     return result
107 })

```

Figura 13 - Fragmento do código do arquivo main contendo a taxonomia da MSC

³⁹ <https://github.com/splhead/sxbrl/blob/main/README.md>

4.4. CONSIDERAÇÕES FINAIS

Neste capítulo, descreveu-se o desenvolvimento do conceito e a codificação da ferramenta XBRL Processor. Dessa forma, nesse capítulo mostrou-se os componentes implementados para o funcionamento da ferramenta e as interações entre eles.

Após testes iniciais efetuados no código e comparações com as instâncias fornecidas pela Secretaria de Finanças do Estado de Rondônia, seguiu-se para a validação da estrutura da instância gerada pelo XBRL Processor que são apresentadas no estudo de caso do Capítulo 5.

5. ESTUDO DE CASO PARA GERAÇÃO DA INSTÂNCIA XBRL E VALIDAÇÃO DA SUA ESTRUTURA

Este capítulo mostra os resultados obtidos na validação da instância gerada com a ferramenta XBRL Processor proposta nesta dissertação. Ele está dividido da seguinte forma. A Seção 5.1 descreve a organização do experimento. Na sequência, a Seção 5.2 mostra os resultados obtidos após a execução do experimento. Finalmente, as considerações finais desse capítulo são descritas na Seção 5.3.

5.1. EXPERIMENTO

A ferramenta foi planejada de modo que fosse possível aferir o resultado do processamento dos dados e resolvesse uma das lacunas identificadas na revisão da literatura apresentada no Capítulo 3. Foi realizada a validação do resultado da geração da instância por uma organização externa a esta pesquisa.

Portanto, desenvolver a ferramenta XBRL Processor parametrizada para a taxonomia e padrão de dados da MSC para o Siconfi, conforme observado nas Figuras 9 e 13, teve alguns benefícios: (i) resolver o problema de geração de instância XBRL; (ii) Resolver o problema da entrega da MSC dos Estados e Municípios para a STN; (iii) ter a estrutura da instância para a MSC verificada por órgão regulador do Brasil.

Para compreender as validações apresentadas nesta seção, é importante saber que a instância foi submetida ao Siconfi de duas formas distintas, a primeira utilizando o ambiente de teste do Siconfi através da central de atendimento, e a segunda submetendo a instância XBRL ao ambiente de produção do Siconfi através da Secretaria de Finanças do Estado de Rondônia seguindo a entrega padrão da instância XBRL da MSC dos Estados.

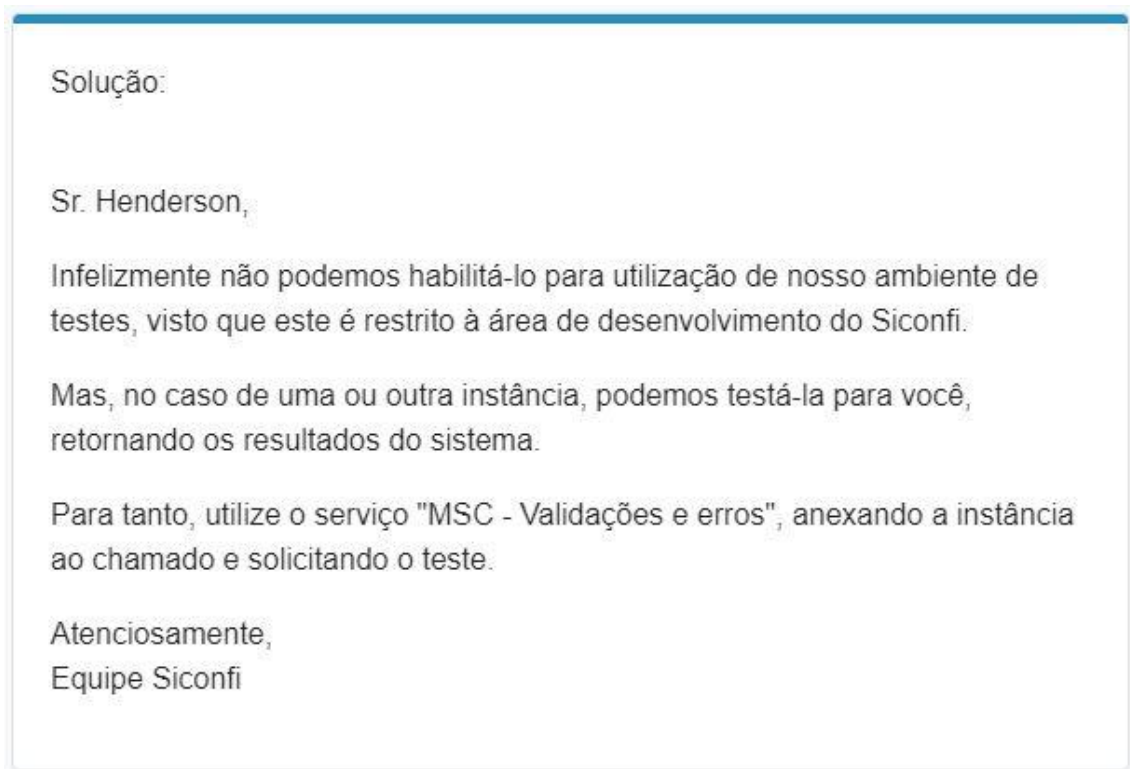
5.2. RESULTADOS

Inicialmente, para efeito de comparação, a Secretaria de Finanças do Estado de Rondônia forneceu dados de meses anteriores, já validados no Siconfi, para facilitar a comparação da instância XBRL gerada pela aplicação ETL para XBRL e a instância entregue ao Siconfi pela Secretaria de Finanças do Estado de Rondônia, e validada sem erros ou inconsistências.

Planejou-se que a validação da estrutura da instância XBRL seria efetuada pelo Siconfi, uma vez que a comparação da instância gerada pela ferramenta XBRL Processor com as instâncias fornecidas pela Secretaria de Finanças do Estado de Rondônia se mostravam idênticas.

Portanto, a Secretaria de Finanças do Estado de Rondônia forneceu dados fictícios e anônimos, ou seja, dados utilizados na base de dados de desenvolvimento, os quais foram carregados nos SGBD's SQLite3, MySQL e MongoDB, para que a instância pudesse ser gerada e entregue ao Siconfi para validação. Após contato com a equipe de suporte do Siconfi, foi orientado solicitar através do sistema de chamado do Siconfi⁴⁰ a validação da instância utilizando a opção "MSC - Validações e erros". A Figura 14 contém a resposta a solicitação de acesso ao ambiente de testes do Siconfi.

Figura 14 - Resposta da equipe de atendimento do Siconfi



⁴⁰ <https://e-servicos.tesouro.gov.br/#/>

Seguiu-se a orientação da equipe de suporte do Siconfi e através do sistema de chamados foi solicitado “MSC - Validações e erros” e fornecido a instância XBRL gerada pelo XBRL Processor com dados fictícios fornecidos pela Secretaria de Finanças do Estado de Rondônia. A Figura 15 mostra a primeira parte da instância XBRL gerada através do XBRL Processor com os dados fornecidos pela Secretaria de Finanças do Estado de Rondônia.

O retorno foi positivo para a estrutura da instância XBRL. Na Figura 16 é mostrada a resposta oficial do Siconfi sobre a instância fornecida.

Figura 15 - Primeiras linhas da Instância XBRL gerada pela ferramenta XBRL Processor

```

1 <?xml version="1.0" encoding="ISO-8859-1"?>
2 <xbrli:xbrl xmlns:xbrli="http://www.xbrl.org/2003/instance" xmlns:gl-bus="http://www.xbrl.org/int/gl/bus/2015-03-25"
3     xmlns:gl-cor="http://www.xbrl.org/int/gl/cor/2015-03-25" xmlns:iso4217="http://www.xbrl.org/2003/iso4217"
4     xmlns:link="http://www.xbrl.org/2003/linkbase" xmlns:xlink="http://www.w3.org/1999/xlink"
5     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
6   <link:schemaRef xlink:href="SICONFI/cor/ext/gl/plt/case-c-b-m-u-t-s/gl-plt-all-2015-03-25.xsd" xlink:type="simple"/>
7   <xbrli:context id="C1">
8     <xbrli:entity>
9       <xbrli:identifiier scheme="http://siconfi.tesouro.gov.br">11EX</xbrli:identifiier>
10    </xbrli:entity>
11    <xbrli:period>
12      <xbrli:instant>2023-01-31</xbrli:instant>
13    </xbrli:period>
14  </xbrli:context>
15  <xbrli:unit id="BRL">
16    <xbrli:measure>iso4217:BRL</xbrli:measure>
17  </xbrli:unit>
18  <xbrli:unit id="u">
19    <xbrli:measure>xbrli:pure</xbrli:measure>
20  </xbrli:unit>
21  <gl-cor:accountingEntries>
22    <gl-cor:documentInfo>
23      <gl-cor:entriesType contextRef="C1">trialbalance</gl-cor:entriesType>
24    </gl-cor:documentInfo>
25    <gl-cor:entityInformation>
26      <gl-bus:reportingCalendar>
27        <gl-bus:reportingCalendarPeriod>
28          <gl-bus:periodIdentifier contextRef="C1">2023-01</gl-bus:periodIdentifier>
29          <gl-bus:periodDescription contextRef="C1">2023-01-01</gl-bus:periodDescription>
30          <gl-bus:periodStart contextRef="C1">2023-01-01</gl-bus:periodStart>
31          <gl-bus:periodEnd contextRef="C1">2023-01-31</gl-bus:periodEnd>
32        </gl-bus:reportingCalendarPeriod>
33      </gl-bus:reportingCalendar>
34    </gl-cor:entityInformation>
35    <gl-cor:entryHeader>
36      <gl-cor:entryDetail>
37        <gl-cor:lineNumberCounter contextRef="C1" decimals="0" unitRef="u">1</gl-cor:lineNumberCounter>
38        <gl-cor:account>
39          <gl-cor:accountMainID contextRef="C1">236110000</gl-cor:accountMainID>
40          <gl-cor:accountSub>
41            <gl-cor:accountSubID contextRef="C1">10111</gl-cor:accountSubID>
42            <gl-cor:accountSubType contextRef="C1">PO</gl-cor:accountSubType>
43          </gl-cor:accountSub>
44        </gl-cor:account>
45        <gl-cor:amount contextRef="C1" decimals="2" unitRef="BRL">2664427.31</gl-cor:amount>
46        <gl-cor:debitCreditCode contextRef="C1">C</gl-cor:debitCreditCode>
47      </gl-cor:xbrlInfo>
48    </gl-cor:entryDetail>
49    <gl-cor:xbrlInclude contextRef="C1">beginning_balance</gl-cor:xbrlInclude>

```

O Siconfi solicita que seja desconsiderado a data do exercício, que consta no documento, por ser o ambiente onde a taxonomia 2023 está instalada no Siconfi. Na Figura 17 temos a solicitação para desconsiderar o ano de 2017. Com relação as linhas 571093 e 702450 criticadas, são referentes aos dados e não a estrutura da instância XBRL. Como os dados são fictícios, erros como este eram esperados.

Figura 16 - Resposta da validação da instância XBRL feita pelo sistema do Siconfi

 Sistema de Informações Contábeis e Fiscais do Setor Público Brasileiro Tesouro Nacional	Secretaria do Tesouro Nacional - STN
	Ministério da Fazenda - MF
	Carregamento da MSC

Parâmetros da Solicitação

Ente:	Rondônia	Tipo de Declaração:	MSC Agregada
Poder:	Executivo	Periodicidade:	Mensal
Exercício:	2017	Período:	Janeiro
Instituição:	Governo do Estado de Rondônia		
Instância:	XBRL_Instance_Generation_MSC.zip		

Erros encontrados durante a validação da instância MSC submetida:

Linha 571093: Elemento 'gl-cor:accountMainID': O valor " com tamanho = '0' não tem um aspecto válido em relação ao minLength '1' do tipo 'nonEmptyString'.

Linha 702450: Elemento 'gl-cor:accountMainID': O valor " com tamanho = '0' não tem um aspecto válido em relação ao minLength '1' do tipo 'nonEmptyString'.

Seguindo a estratégia planejada, a mesma instância que foi entregue ao Siconfi também foi entregue a Secretaria de Finanças do Estado de Rondônia para que fosse submetida ao Siconfi seguindo os procedimentos rotineiros de entrega da MSC. Dessa forma, é possível submeter a instância gerada pela ferramenta XBRL Processor ao ambiente de produção, ou seja, o ambiente real de validação.

Figura 17 - Pedido de desconsideração do exercício 2017

Prezado,

O seguinte comentário foi feito por **E-Serviços do Tesouro Nacional** para o chamado **CH202304887**:

Sr. Henderson,

Anexamos o resultado da tentativa de carregamento de sua instância MSC.

Não estranhe o exercício de 2017 no arquivo, pois é onde está instalada a taxonomia 2023 em nosso ambiente de teste.

Atenciosamente,

Equipe Siconfi

Após submeter a instância ao Siconfi através da Secretaria de Finanças do Estado de Rondônia o retorno foi exatamente o mesmo recebido pelo Siconfi através de seu sistema de atendimento. Na Figura 18 podemos observar a validação da instância feita pelo ambiente de produção do Siconfi.

Figura 18 - Validação da instância em ambiente de produção

Módulo Declarações e MSC / Elaborar MSC / Formulário

MSC Agregada

Exercício: 2023
 Período: Janeiro
 Ente: Rondônia
 Poder: Executivo
 Órgão: Governo do Estado de Rondônia

Taxonomia: Versão 2023 - 04-01-23 - v11
 Vigente a partir de: 04/01/2023

O Arquivo carregado possui os seguintes erros de validação:

Legenda dos erros: ● Estrutura (Impeditivos) ● Conteúdo (Impeditivos) ● Conteúdo (Alerta)

Tipo de Erro	Mensagem
●	Linha 571093: Elemento 'gl-cor:accountMainID'. O valor "" com tamanho = '0' não tem um aspecto válido em relação ao minLength '1' do tipo 'nonEmptyString'.
●	Linha 702450: Elemento 'gl-cor:accountMainID'. O valor "" com tamanho = '0' não tem um aspecto válido em relação ao minLength '1' do tipo 'nonEmptyString'.

✖ Foram encontrados 2 erros impeditivos. Para mais informações, clique em "Imprimir Validação"

Nota-se que os erros nas linhas 571093 e 702450 são os mesmos da validação do ambiente de testes do Siconfi. Portanto a ferramenta XBRL Processor gerou a instância XBRL da Matriz de Saldos Contábeis do Estado de Rondônia para o Sistema de Informações Contábeis e Fiscais do Setor Público Brasileiro da Secretaria do Tesouro Nacional com a sua estrutura sem erros ou inconsistências.

Portanto, a aplicação está em funcionamento para conectar com fontes de dados relacionais ou NoSQL e gerar a instância XBRL da Matriz de Saldos Contábeis de acordo com a taxonomia definida pelo Siconfi.

5.3. CONSIDERAÇÕES FINAIS

Descreveu-se neste capítulo o estudo de caso para validação da instância XBRL gerada pelo XBRL Processor proposto nesta dissertação. Pode-se verificar que o órgão pertencente ao governo federal do Brasil validou de duas formas a instância XBRL sendo que em uma das submissões foi através do ambiente de produção do Siconfi simulando uma entrega real dos dados do Estado de Rondônia.

Por fim, com essa avaliação também foi possível visualizar o funcionamento da ferramenta XBRL Processor que entregou o resultado exatamente como proposto.

6. CONCLUSÃO

Este capítulo apresenta as considerações finais sobre esta dissertação, incluindo as principais contribuições alcançadas, as limitações e trabalhos futuros, por fim, os trabalhos publicados.

6.1. CONSIDERAÇÕES FINAIS

De acordo com o que foi apresentado no Capítulo 1, gerar instâncias XBRL não é um processo simples, e a ausência de ferramentas *open-source* que tratam da problemática pode ser um fator contribuinte para a não adesão da tecnologia XBRL de forma expressiva por diversos órgãos reguladores e instituições privadas, principalmente nas organizações no Brasil. Este cenário é agravado pela escassez de ferramentas que utilizam os SGBD's como fonte de dados.

Conforme mostrado no Capítulo 3, existe carência de pesquisas que atendam as questões levantadas neste trabalho, mesmo com a necessidade de gerar instâncias XBRL de dados nos formatos avaliados estarem crescendo continuamente em função da evolução e expansão da internet e de imposições legais.

A partir dos resultados apresentados nesta dissertação, pode-se responder às questões de pesquisa apresentadas no Capítulo 1, as quais são:

- Questão 1: Existem soluções que permitam o mapeamento de dados de diferentes formatos para XBRL?

Existem soluções, no entanto, são soluções proprietárias que se limitam a atender seguimentos específicos como o bancário. Outra limitação importante está relacionada a geração de instâncias XBRL apenas de dados contidos em suas bases de dados, não permitindo a utilização de outras fontes ou SGBD's.

- Questão 2: As soluções existentes são *open-source*, ou seja, são de código aberto e gratuitas?

Apenas o software Arelle é *open-source*, no entanto, tem como finalidade primária a leitura de instâncias XBRL, já a geração de instâncias XBRL precisa ser desenvolvida, assim como o carregamento dos dados oriundos de SGBD's.

- Questão 3: as soluções propostas possuem generalização suficiente de maneira que permitem o mapeamento de qualquer modelo de dados (e.g. relacional, NoSQL, CSV, JSON) para XBRL?

Nenhum software investigado apresentou a capacidade de ler dados de diversas fontes.

- Questão 4: as soluções propostas atendem aos requisitos do Siconfi e Matriz de Saldos Contábeis (MSC) da STN?

Um dos trabalhos investigados no Capítulo 3 apresentou solução para a entrega da MSC da STN, o trabalho de Bragança (Bragança et al., 2019) é limitado a leitura dos dados apenas no formato CSV e utilização de softwares ETL de terceiros para gerar a instância XBRL.

- Questão 5: É possível construir uma ferramenta com a qual qualquer taxonomia possa ser utilizada para gerar instâncias XBRL de qualquer fonte de dados?

Esta dissertação demonstrou que é possível construir uma ferramenta para gerar instâncias XBRL de qualquer taxonomia com dados de variadas fontes.

Também, a hipótese formulada nesta dissertação é válida, a qual verificou-se que:

- Hipótese 1: É possível construir uma ferramenta de extração, transformação e carga de dados no formato XBRL, desacoplada de qualquer taxonomia e de diferentes modelos de dados e capaz de adaptar-se à taxonomia e fonte de dados necessárias.

Foi apresentado no Capítulo 4 a ferramenta XBRL Processor para a geração de instâncias XBRL com dados de fontes relacionais ou denormalizados. A ferramenta possui código fonte aberto e está parametrizada com a taxonomia do Siconfi para gerar a MSC, no entanto, a parametrização pode ser atualizada ou configurada para outra taxonomia XBRL.

Por fim, o Capítulo 5 exibiu por meio de um estudo de caso que a ferramenta está apta a atender a necessidade dos estados e municípios brasileiros na entrega da MSC para o Siconfi, atendendo assim a determinação legal do governo federal.

6.2. CONTRIBUIÇÕES

A principal contribuição desta dissertação é o modelo proposto para a extração, transformação e carga de dados provenientes de diferentes fontes e formatos de dados para gerar instâncias XBRL, adicionalmente, a partir do modelo proposto no Capítulo 4, teve-se como contribuição o desenvolvimento de uma ferramenta adaptável que processa dados de diferentes fontes e gera instâncias XBRL a partir da parametrização de diferentes taxonomias. Essa ferramenta foi desenvolvida com o foco na adaptabilidade do código aos diferentes ambientes

tecnológicos, sendo capaz de extrair dados de variados modelos distintos de tecnologias de armazenamento de dados, simplificando a geração de instâncias XBRL.

Desta forma a ferramenta XBRL Processor, com seu código fonte aberto, consegue atender a necessidade de gerar instâncias XBRL da MSC para o Siconfi e permite ser estendida para solucionar outras necessidades de geração de instâncias XBRL, até mesmo ser incorporado a aplicações em desenvolvimento ou nortear o desenvolvimento de soluções próprias para ambientes legados.

6.3. LIMITAÇÕES E TRABALHOS FUTUROS

Apesar dos benefícios alcançados por este trabalho, algumas limitações foram identificadas e que precisam ser superadas em trabalhos futuros:

- Embora a ferramenta facilite a adição de novas fontes de dados, não está disponível o carregamento de arquivos CSV como fonte de dados;
- Validar os dados utilizados para gerar a instância XBRL confrontando com as especificações contidas na taxonomia, extrapola o escopo abordado nesta dissertação. Contudo, é possível que a ferramenta XBRL Processor evolua para superar essa limitação;
- Outra limitação identificada nesta dissertação é a presença de elementos da taxonomia diretamente no código, sendo necessário, para usufruir de outras taxonomias ou atualizações, a edição do código fonte;
- Outra limitação refere-se à geração das instâncias XBRL apenas no formato XML, o consorcio responsável pela tecnologia XBRL admite também instâncias nos formatos JSON e CSV.

Como investigação futura, sugere-se adicionar a ferramenta outras funcionalidades:

- Desenvolver interface gráfica do utilizador (GUI) juntamente com testes de usabilidade (Akram; Aqeel; Hamid, 2023) e *User Experience* de modo a permitir a inclusão de utilizadores sem conhecimento de desenvolvimento de software;
- Desenvolver módulo de verificação e validação dos dados carregados das fontes de dados, frente a taxonomia selecionada;
- Gerar instâncias nos formatos XBRL-JSON e XBRL-CSV, admitidos pelo consórcio responsável pela tecnologia XBRL além do formato XBRL-XML;

- Adequar a biblioteca fast-xml-parser para o carregamento da taxonomia através de seus arquivos, eliminando a presença de elementos da taxonomia no código fonte, conforme Figura 13, possibilitando a troca e atualização facilitada, com a finalidade de abranger um maior número de organizações capazes de usufruir da tecnologia XBRL no intercâmbio de informações contábeis e financeiras;
- Integração da ferramenta XBRL Processor com o Motor XBRL proposto no trabalho de Bragança (Bragança et al., 2019) para geração de instâncias a partir de arquivos CSV; muito embora a ferramenta XBRL Processor atenda inicialmente a MSC para o Siconfi.

6.4. PUBLICAÇÕES

Durante o desenvolvimento desta dissertação, artigos científicos foram submetidos para diferentes eventos, com o objetivo de validar na comunidade acadêmica as contribuições. Tais publicações são listadas a seguir:

1. Bragança, Henderson Acosta, Silas Pinho Ladislau, Marcio Alexandre Pereira da Silva and Paulo Caetano. 2019. “XBRL-ETL Engine: A Data Transformation Tool for Xbrl-siconfi Taxonomy.” International Conference on Information Systems and Technology Management - CONTECSI, (1), 1–19. <https://doi.org/10.5748/16CONTECSI/XBR>
2. Braganca, Henderson Acosta, Paulo Caetano, and Nacles Bernadino. 2022. “Data Mapping for XBRL: A Systematic Literature Review.” American Academic Scientific Research Journal for Engineering, Technology, and Sciences 90:124–43
3. Bragança, Henderson Acosta, Paulo Caetano, and Nacles Bernadino. 2023. “Web Service and Internet of Things: A Systematic Literature Review.” International Journal of Applied Nonlinear Science.
4. Bragança, Henderson Acosta, Silas Pinho Ladislau, Paulo Caetano and Daniel Díaz. 2023. “Proposal of Conceptual Model for Generation of Xbrl Instances.” Journal of Information Systems and Technology Management – Jistem USP.

REFERÊNCIAS

- AKRAM, S.; AQEEL, M.; HAMID, K. Enhancing Software Quality Through Usability Experience and Hci Enhancing Software Quality Through Usability Experience and Hci Design Principles, p.45–75, feb.2023. <https://doi.org/10.17605/OSF.IO/MFE45>
- ALAMI, A. E.L.; BAHAJ, M. Framework for a complete migration of relational databases to other types of databases(object oriented OO, object-relational OR, XML). *In: IEEE/ACS INTERNATIONAL CONFERENCE ON COMPUTER SYSTEMS AND APPLICATIONS, AICCSA. 2017. Proceedings [...]* 2017. <https://doi.org/10.1109/AICCSA.2016.7945763>
- ASIMADI, E.; REIFF-MARGANIEC, S.; DONNELLY, B.; BAKER, J.; FANG, D. Semantic approach to financial data integration for enabling new insights. *In: CEUR WORKSHOP 1890, 2017. Proceedings [...]*, 2017. p.1–15.
- BAHTA, R.; ATAY, M. Translating JSON data into relational data using schema-oblivious approaches. *In: ACMSE 2019 - 2019 ACM SOUTHEAST CONFERENCE, 2019, New York. Proceedings [...]*. New York, USA: ACM Press, 2019. p. 233–236. <https://doi.org/10.1145/3299815.3314467>
- BAI, L.; YAN, L.; MA, Z. M.; XU, C. Incorporating fuzziness in spatiotemporal XML and transforming fuzzy spatiotemporal data from XML to relational databases. *Applied Intelligence*, v. 43, p. 707-721, 2015. <https://doi.org/10.1007/s10489-015-0677-7>
- BEELITZ, C.. The dilemma of XBRL-XML versus XBRL-JSON regarding linkage of financial information. *CEUR WORKSHOP 2017. Proceedings [...]* 2017. p.1–11.
- BELEV, I. Alternatives for Storing and Validating XBRL Data. *American Scientific Research Journal for Engineering, Technology, and Sciences (ASRJETS)*, v. 60, n. 1, p. 191-201, 2019. Retrieved from https://asrjetsjournal.org/index.php/American_Scientific_Journal/article/view/5289
- BIERMAN, Gavin; ABADI, Martín; TORGERSEN, Mads. *Understanding TypeScript*. 2014. Retrieved from <http://blogs.msdn.com/b/typescript/>
- BIKAKIS, N.; TSINARAKI, C.; STAVRAKANTONAKIS, I.; GIOLDASIS, N.; CHRISTODOULAKIS, S. The SPARQL2XQuery interoperability framework: Utilizing Schema Mapping, Schema Transformation and Query Translation to Integrate XML and the Semantic Web. *World Wide Web*, v. 18, p. 403-490, 2015. <https://doi.org/10.1007/s11280-013-0257-x>
- BIN, Y. U.; XU, L. U.; CONG, T.; ZHEN-HUA, D.; NAN, Z.. Parallel Runtime Verification for Calling Sequences of SQLite3 Database APIs. *Journal of Software*, v. 33, n. 8, p. 2755-2768, 2022.Retrieved from <http://josen/article/abstract/6596>
- BOGNER, J.; MERKEL, M. To Type or Not to Type? A Systematic Comparison of the Software Quality of JavaScript and Typescript Applications on GitHub. *In: INTERNATIONAL CONFERENCE ON MINING SOFTWARE REPOSITORIES, 19., 2022. Proceedings [...]*. . New York, NY, USA: Association for Computing Machinery, 2022. p. 658–669. <https://doi.org/10.1145/3524842.3528454>
- BRAGANÇA, H. A.; LADISLAU, S. P.; da SILVA, M. A. P.; da SILVA, P. C. XBRL-ETL Engine: A Data Transformation Tool For XBRL-Siconfi Taxonomy. *INTERNATIONAL CONFERENCE ON INFORMATION SYSTEMS AND TECHNOLOGY MANAGEMENT - CONTECSI, 1. 2019. Proceedings [...]*. 2019. p. 1–19. <https://doi.org/10.5748/16CONTECSI/XBR>

- CELESTI, A.; FAZIO, M.; VILLARI, M. A study on join operations in MongoDB preserving collections data models for future internet applications. *Future Internet*, v. 11, n. 4, p. 83, 2019. <https://doi.org/10.3390/fi11040083>
- CERQUEIRA, M. G. de; SILVA, P. C. da. A survey of XBRL adoption impact on financial software development processes and software quality. *International Journal of Business Information Systems*, v. 37, n. 2, p. 263-286, 2021.. <https://doi.org/10.1504/IJBIS.2021.115366>
- CERQUEIRA, M. G. de; SILVA, P. C. da. Coming Impacts of Xbrl Adoption in Financial Software Development Processes and Software Quality Factors: a Systematic Mapping. In: CONTECSI INTERNATIONAL CONFERENCE ON INFORMATION SYSTEMS AND TECHNOLOGY MANAGEMENT, 13, 2016. *Proceedings [...]*. 2016. p.3185–3209. <https://doi.org/10.5748/9788599693124-13contecsi/ps-4103>
- CHEN, Y. Worst case optimal joins on relational and XML data. In: *Proceedings of the ACM SIGMOD INTERNATIONAL CONFERENCE ON MANAGEMENT OF DATA Proceedings [...]*. New York, NY, USA: Association for Computing Machinery, 2018. p. 1833–1835. <https://doi.org/10.1145/3183713.3183721>
- CHRISTUDAS, B. MySQL. In: *Practical microservices architectural patterns: event-based Java Microservices with spring boot and spring cloud*. Berkeley, CA: Apress, 2019. p. 877–884 https://doi.org/10.1007/978-1-4842-4501-9_27
- DEBINSKI, M.; BREITINGER, F.; MOHAN, P. . Timeline2GUI: A Log2Timeline CSV parser and training scenarios. *Digital Investigation*, v. 28, p. 34-43, 2019. <https://doi.org/10.1016/J.DIIN.2018.12.004>
- DIMOU, A.; SANDE, M. VANDER, COLPAERT, P.; VERBORGH, R.; MANNENS, E.; VAN DE WALLE, R. RML: A generic language for integrated RDF mappings of heterogeneous data. In: CEUR WORKSHOP, 2014. *Proceedings [...]*. 2014. p.1184.
- DOI, Y.; TOYAMA, M. ToT for CSV: Accessing Open Data CSV Files through SQL. In: INTERNATIONAL CONFERENCE ON INFORMATION INTEGRATION AND WEB-BASED APPLICATIONS & SERVICES, 21., 2019, New York, NY, USA. *Proceedings [...]* New York, NY, USA: Association for Computing Machinery, 2019. p. 423–429. <https://doi.org/10.1145/3366030.3366130>
- DUNCE, M. M. M.; SILVA, P. C. da, VIANA, S.. Similarity Evaluation Between Concepts Represented By Xbrl, 3933–3963. In: CONTECSI-INTERNATIONAL CONFERENCE ON INFORMATION SYSTEMS AND TECHNOLOGY MANAGEMENT, 10., 2013. *Proceedings [...]* 2013. <https://doi.org/10.5748/9788599693094-10contecsi/ps-457>
- FELIZARDO, K. R. N.. *Revisao sistematica da literatura em engenharia de software: teoria e prática*. Rio de Janeiro: Elsevier, 2017.
- FROZZA, A. A.; MELLO, R. dos S. JS4Geo: a canonical JSON Schema for geographic data suitable to NoSQL databases. *GeoInformatica*, v.24, n.4, p.987–1019, 2020. <https://doi.org/10.1007/s10707-020-00415-w>
- GAMAL, M. M.; AHMED, A. E. A.; HEFNY, H. A.; EL-MONEIM, M. A. A literature survey on mapping between fuzzy XML databases and relational or object oriented databases. In: IEEE WORLD CONFERENCE ON COMPLEX SYSTEMS, WCCS, 2016. *Proceedings [...]* 2016. p. 1–6. <https://doi.org/10.1109/ICoCS.2015.7483293>
- GRAY, G. L.; MILLER, D. W. XBRL: Solving real-world problems. *International Journal of Disclosure and Governance*, v.6, n.3, p.207–223, 2009. <https://doi.org/10.1057/jdg.2009.8>

GROFF, J. R.; WEINBERG, P. N. *SQL, the complete reference*. Osborne: McGraw-Hill, 1999. Disponível em: [http://englishonlineclub.com/pdf/SQL - The Complete Reference](http://englishonlineclub.com/pdf/SQL-The-Complete-Reference) [EnglishOnlineClub.com].pdf Acesso em: 2 jan. 2019.

GYÖRÖDI, C. A.; DUMȘE-BURESCU, D. V.; ZMARANDA, D. R.; GYÖRÖDI, R. A Comparative Study of MongoDB and Document-Based MySQL for Big Data Application Data Management. *Big Data and Cognitive Computing*, v.6, n.2, 2022. <https://doi.org/10.3390/bdcc6020049>

HABIB, A.; SHINNAR, A.; HIRZEL, M.; PRADEL, M. Finding Data Compatibility Bugs with JSON Subschema Checking. In: ACM SIGSOFT IN: INTERNATIONAL SYMPOSIUM ON SOFTWARE TESTING AND ANALYSIS, 30., 2021, New York, NY, USA. *Proceedings [...]* New York, NY, USA: Association for Computing Machinery, 2021. p. 620–632. <https://doi.org/10.1145/3460319.3464796>

JAYASHREE, G.; PRIYA, C. Data Integration with XML ETL Processing. In: INTERNATIONAL CONFERENCE ON COMPUTER SCIENCE, ENGINEERING AND APPLICATIONS, ICCSEA, 2020. *Proceedings [...]* 2022. <https://doi.org/10.1109/ICCSEA49143.2020.9132936>

KITCHENHAM, B.; CHARTERS, S.. Guidelines for performing systematic literature reviews in software engineering. *Technical Report, Ver. 2.3 EBSE Technical Report*. EBSE. Durham: University of Durham, 2007.

LAZZARI, L.; FARIAS, K. An exploratory study on the effects of pair programming. In: Proceedings of Make sure to enter the correct conference title from your rights confirmation email. In: CONFERENCE ACRONYM ' 20., 2022. *Proceedings [...]* 2022. <https://doi.org/10.1049/ic:20040395>

LIU, D.; ETUDO, U.; YOON, V. X-IM framework to overcome semantic heterogeneity across XBRL filings. *Journal of the Association for Information Systems*, v.21, n.4, p.971–1000, 2020. <https://doi.org/10.17705/1jais.00626>

LYAMIN, A. V.; CHEREPOVSKAYA, E. N. XML-Relational mapping using production rule system. In: INTELLIGENT SYSTEMS CONFERENCE, INTELLISYS, 2018. *Proceedings [...]* 2018. p. 422–429. <https://doi.org/10.1109/IntelliSys.2017.8324328>

MAATUK, A. M.; ALI, M. A.; ALJAWARNEH, S.. An algorithm for constructing XML Schema documents from relational databases. In: ACM INTERNATIONAL CONFERENCE, 2015, New York, NY, USA. *Proceedings [...]* 2015. New York, NY, USA: Association for Computing Machinery. <https://doi.org/10.1145/2832987.2833007>

MAO, J.; YE, X. Relational schema and XML schema bidirectional mapping algorithm based on the intermediate object tree. In: IEEE INTERNATIONAL CONFERENCE ON COMPUTER AND COMMUNICATIONS, 3., ICC3 2018 *Proceedings [...]* 2018. 2380–2383. <https://doi.org/10.1109/CompComm.2017.8322961>

MÄS, S.; HENZEN, D.; BERNARD, L.; MÜLLER, M.; JIRKA, S.; SENNER, I. Generic schema descriptions for comma-separated values files of environmental data. *Int. Conf. Geogr. Inf. Sci. (AGILE)*, 2018.

NASSIRI, H.; MACHKOUR, M.; HACHIMI, M.. Integrating XML and Relational Data. *Procedia Computer Science*, v.110, p.422–427, 2017. <https://doi.org/10.1016/j.procs.2017.06.107>

NASSIRI, H.; MACHKOUR, M.; HACHIMI, M. One query to retrieve XML and Relational Data. *Procedia Computer Science*, v.134, p.340–345, 2018.

<https://doi.org/10.1016/j.procs.2018.07.201>

NAVATHE, E. Sistemas de Banco de Dados. *Journal of Chemical Information and Modeling*, 2013.

NIEWERTH, M.; SCHWENTICK, T. Reasoning About XML Constraints Based on XML-to-Relational Mappings. *Theory of Computing Systems*, v. 62, p. 1826-1879, 2018. <https://doi.org/10.1007/s00224-018-9846-5>

PETKOVIĆ, D. JSON Integration in Relational Database Systems. *International Journal of Computer Applications*, v. 168, n. 5, p. 14-19, 2017a. <https://doi.org/10.5120/ijca2017914389>

PETKOVIĆ, D.. SQL/JSON Standard: Properties and Deficiencies. *Datenbank-Spektrum*, v.17, n.3, p.277–287, 2017b. <https://doi.org/10.1007/s13222-017-0267-4>

PRAMMER, M.; SAHADEVAN RAJESH, S.; CHEN, J.; PATEL, J. M. *Introducing a Query Acceleration Path for Analytics in SQLite3; Introducing a Query Acceleration Path for Analytics in SQLite3. In: ANNUAL CONFERENCE ON INNOVATIVE DATA SYSTEMS RESEARCH (CIDR '22), 12., 2022. Proceedings [...] 2022.*

QTAISH, A.; AHMAD, K.. XAncestor: An efficient mapping approach for storing and querying XML documents in relational database using path-based technique. *Knowledge-Based Systems*, v. 114, p.167–192, 2016. <https://doi.org/10.1016/j.knosys.2016.10.009>

RICCIO, E.; SAKATA, M.; MOREIRA, O.; QUONIAM, L. Introdução ao XBRL: nova linguagem para a divulgação de informações empresariais pela internet. *Ciência Da Informação*, v.35, n.3, 166–182, 2006. <https://doi.org/10.1590/s0100-19652006000300016>

RODRIGUEZ, J.; GUARDO, G. *MySQL introduces the essential concept*. 2005, v. 3.

SALEM, R.; DARMONT, J.; BOUSSAID, O.; SALEM, R.; DARMONT, J.; BOUSSAID, O.; BOUSSA, O. Active XML-based Web data integration To cite this version : HAL Id : hal-01433718 Active XML-based. *Web Data Integration*, v.15, n.3, 2017.

SILVA, P. C.; SILVA, L.; SANTOS, A.; CRUZ, M. O Framework Xbrl. INTERNATIONAL CONFERENCE ON INFORMATION SYSTEMS AND TECHNOLOGY MANAGEMENT, 5., 2018. *Proceedings [...] 2018*. p. 4343–4365.

SOARES, B. E.; BOSCARIOLI, C. (2013). Modelo de banco de dados colunar: características, aplicações e exemplos de sistemas. *In: ERBD–SBC, 9., Cascavel, 2013. Proceedings [...] Cascavel: Escola Regional de Banco de Dados–Sociedade Brasileira de Computação, 2013. Disponível em: https://turing.pro.br/anais/ERBD-2013/artigos/pesquisa/111410.pdf Acesso em: 20 jan. 2019.*

SOARES, C. S.; MALLONE, V.; ANDRADE, N. de. Gestão pública municipal e os processos internos determinantes para o envio da matriz de saldos contábeis. *Revista Eletrônica Gestão e Serviços*, v. 12, n. 2, p. 3502-3524, 2021.

SOMMERVILLE, I. *Software engineering*. São Paulo: Pearson, 2011.

SONG, E.; HAW, S. C.. XML-REG: Transforming xml into relational using hybrid-based mapping approach. *IEEE Access*, v.8, 2020. 177623–177639. <https://doi.org/10.1109/ACCESS.2020.3026006>

SONG, E.; HAW, S. C.; CHUA, F. F. Handling XML to relational database transformation using model-based mapping approaches. *In: IEEE CONFERENCE ON OPEN SYSTEMS, ICOS 2018. Proceedings [...] 2018*. p. 65–70. <https://doi.org/10.1109/ICOS.2018.8632805>

SPINK, P.; AROUCA, F. L.; TEIXEIRA, M. A. (2002). O Banco de Dados. *Cadernos Gestão*

Pública e Cidadania, v.7, n.22, 2002. <https://doi.org/10.12660/cgpc.v7n22.52342>

STN, S. do T. N. Matriz de Saldos Contábeis. Brasília: Secretaria do Tesouro Nacional, 2017. Retrieved from

https://siconfi.tesouro.gov.br/siconfi/pages/public/arquivo/conteudo/Cartilha_Matriz_de_Saldos_Contabeis.pdf

YAGHMAZADEH, N.; WANG, X.; DILLIG, I. Automated Migration of Hierarchical Data to Relational Tables Using Programming-by-Example. *Proc. VLDB Endow.* 11, n. 5, p. 580-593, 2018. <https://doi.org/10.1145/3177732.3177735>

ZHU, H.; YU, H.; FAN, G.; SUN, H. Mini-XML: an efficient mapping approach between XML and relational database. *In: IEEE/ACIS INTERNATIONAL CONFERENCE ON COMPUTER AND INFORMATION SCIENCE, ICIS, 16., 2017. Proceedings [...] 2017.* p. 839–843. <https://doi.org/10.1109/ICIS.2017.7960109>