



**UNIVERSIDADE ESTADUAL DA PARAÍBA
CAMPUS I – CAMPINA GRANDE
CENTRO DE CIÊNCIAS E TECNOLOGIA
DEPARTAMENTO DE COMPUTAÇÃO
CURSO DE GRADUAÇÃO EM COMPUTAÇÃO**

RAMON RODRIGUES DE SALES

**IMPLANTAÇÃO E MANUTENÇÃO DO SISTEMA DE ANÁLISE FISCAL PARA
SECRETARIA DE ESTADO DA FAZENDA DA PARAÍBA**

CAMPINA GRANDE - PB

2022

RAMON RODRIGUES DE SALES

**IMPLANTAÇÃO E MANUTENÇÃO DO SISTEMA DE ANÁLISE FISCAL PARA
SECRETARIA DE ESTADO DA FAZENDA DA PARAÍBA**

Trabalho de Conclusão de Curso apresentado ao Curso de Ciência da Computação do Centro de Ciência e Tecnologia da Universidade Estadual da Paraíba, como requisito parcial à obtenção do título de bacharel em Computação.

Orientador: Prof. Dr. Fábio Luiz Leite Júnior

**CAMPINA GRANDE - PB
2022**

É expressamente proibido a comercialização deste documento, tanto na forma impressa como eletrônica. Sua reprodução total ou parcial é permitida exclusivamente para fins acadêmicos e científicos, desde que na reprodução figure a identificação do autor, título, instituição e ano do trabalho.

S163i Sales, Ramon Rodrigues de.
Implementação e manutenção do sistema de análise fiscal para a Secretaria de Estado da Fazenda da Paraíba [manuscrito] / Ramon Rodrigues de Sales. - 2022.
58 p. : il. colorido.

Digitado.

Trabalho de Conclusão de Curso (Graduação em Computação) - Universidade Estadual da Paraíba, Centro de Ciências e Tecnologia, 2022.

"Orientação : Prof. Dr. Fábio Luiz Leite Júnior, Coordenação do Curso de Computação - CCT."

1. Base de dados. 2. Secretaria de Estado da Fazenda. 3. Sistema de Análise Fiscal. 4. Paraíba. I. Título

21. ed. CDD 004

RAMON RODRIGUES DE SALES

**IMPLANTAÇÃO E MANUTENÇÃO DO SISTEMA DE ANÁLISE FISCAL
PARA SECRETARIA DE ESTADO DA FAZENDA DA PARAÍBA**

Trabalho de Conclusão de Curso apresentado ao Curso de Ciência da Computação do Centro de Ciência e Tecnologia da Universidade Estadual da Paraíba, como requisito parcial à obtenção do título de bacharel em Computação.

Aprovado em: 09 de Agosto de 2022.

BANCA EXAMINADORA



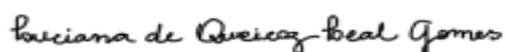
Prof. Dr. Fábio Luiz Leite Júnior (DC - UEPB)

Orientador(a)



Prof. Dr. Paulo Eduardo Lima e Silva (DC - UEPB)

Examinador(a)



Profa. Me. Luciana de Queiroz Leal Gomes (DC - UEPB)

Examinador(a)

A minha mãe, pelo amor, apoio,
força, companheirismo, DEDICO.

AGRADECIMENTOS

A minha mãe Santina Rodrigues de Oliveira, por sempre me apoiar nos meus sonhos, por todo amor, carinho, cuidado e compreensão

A meu amigo Adson de Macedo Nascimento, por sempre me apoiar nessa jornada e partilhar comigo as alegrias, tristezas e todo conhecimento que adquirimos nesse longo período de aprendizado na universidade.

A meu amigo Thaíram Michel Santos Ataíde, por sempre me apoiar nessa jornada e partilhar comigo as alegrias, tristezas e todo conhecimento que adquirimos nesse longo período de aprendizado na universidade.

A minha esposa Nayara do Nascimento Torres, pela paciência e compreensão por todas as vezes que precisei me ausentar para focar nos estudos, pelo amor, carinho e companheirismo.

Ao meu orientador Fábio Luiz Leite Júnior, por toda dedicação, atenção e orientações que foram fundamentais para conclusão deste trabalho.

A todos os meus colegas de curso que de alguma forma me ajudaram a alcançar esse objetivo.

“Por vezes sentimos que aquilo que fazemos não é senão uma gota de água no mar. Mas o mar seria menor se lhe faltasse uma gota.”

Madre Teresa de Calcutá

RESUMO

A aplicação desenvolvida para a Secretaria da Fazenda do Estado da Paraíba (SEFAZ) tem como objetivo auxiliar os fiscais no processo de fiscalização tributária das diversas empresas que possuem sede localizada no estado da Paraíba. O principal objetivo da aplicação é reunir todos os dados necessários para que os fiscais possam fazer suas análises em um único ambiente. Além disso, a ferramenta desenvolvida trabalha reunindo as informações de notas fiscais de produto eletrônico (NF-e) e notas fiscais do consumidor eletrônicas (NFC-e) que estão dissipadas em bases de dados diferentes. O projeto utiliza uma estratégia baseada no conceito de extração, transformação e carga (ETL). Inicialmente, é feita uma extração dos dados que serão utilizados na fiscalização, posteriormente são realizadas algumas transformações nos dados com intuito de simplificar a leitura. Na fase final os dados são carregados em uma base de dados onde os usuários finais irão realizar consultas e a análise dos dados. As ferramentas utilizadas para realização das diversas atividades no projeto foram: SQL Server Integration Service (SSIS) para construção do projeto de integração dos dados; banco de dados SQL Server para armazenamento dos dados; linguagem SQL para consulta e manipulação dos dados.

Palavras-Chave: Aplicação. SEFAZ. Base de dados. Dados. ETL.

ABSTRACT

This work presents an application developed for the Secretary of Treasury of the State of Paraíba (SEFAZ), which aims to assist the inspectors in the process of tax inspection of the various companies that have their headquarters located in the state of Paraíba. The main objective of the application is to gather all the necessary data so that the inspectors can make their analysis in a single environment. Furthermore, the tool developed works by bringing together the information of electronic invoices (NF-e) and electronic invoices of the consumer (NFC-e) which are spread in different databases. The project uses a strategy based on the concept of extraction, transformation and loading (ETL). Initially, the application extracts data to be applied in the inspection, subsequently, some transformations are performed in the data in order to simplify the reading. In the final phase, the data is loaded into a database in which the final users will perform queries and data analysis. The tools used to carry out the various activities in the project were: SQL Server Integration Service (SSIS) to build the data integration project; SQL Server database for data storage; SQL language for querying and manipulating data.

Keywords: Application. SEFAZ. Data base. Data. ETL.

LISTA DE ILUSTRAÇÕES

Figura 1 - Fluxograma integração de dados.	19
Figura 2 - Extração, transformação e carga dos dados.	20
Figura 3 – Estrutura de um banco utilizado na fiscalização.	21
Figura 4 – Como é definido um data mart.....	21
Figura 5 – Diferença entre data lake e data warehouse	22
Figura 6 – Diagrama de entidade-relacionamento	23
Figura 7 – Diagrama desenvolvido no modelo lógico	24
Figura 8 – Diagrama desenvolvido no modelo físico	25
Figura 9 – Exemplo de código gerado pelo SGBD a partir do diagrama do modelo físico.....	25
Figura 10 - Ranking SGBDS de acordo com a popularidade.....	26
Figura 11 - Visão macro de um pacote que realiza a extração de dados.	28
Figura 12 - Visão do Job de execução das cargas no SSMS.	29
Figura 13 – Relatório gráfico desenvolvido no SSRS	30
Figura 14 – Relatório desenvolvido no SSRS.....	31
Figura 15 – Diagrama de caso de uso.....	33
Figura 16 – Diagrama de atividades	34
Figura 17 - Visão geral do processo de carga.	35
Figura 18 - Carga da tabela “ITEM_NFE” no projeto antigo.	37
Figura 19 - Carga da tabela “ITEM_NFE” na nova versão do projeto.....	38
Figura 20 - Parte 1 do esquema de validação realizada pelo projeto antigo ..	39
Figura 21 - Parte 2 do esquema de validação realizada pelo projeto antigo ..	39
Figura 22 - Nova estratégia de validação	40
Figura 23 - Pacotes inúteis e containers desabilitados.....	42
Figura 24 - Versão atual do projeto do pacote “NFE Parte 1”.....	42
Figura 25 - Script de limpeza das bases SQL Server.	44
Figura 26 - Script de limpeza do Oracle.....	45
Figura 27 - Estrutura da rotina de carga diária.	46
Figura 28 – Pacote NFE 1 BDFisc.....	48
Figura 29 – Pacote NFE 1 BDFisc 2.0.....	48
Figura 30 - Distribuição das cargas.	49

Figura 31 – Resumo das métricas do BDFisc.....	51
Figura 31 – Resumo das métricas do BDFisc 2.0.....	52
Figura 32 – Resumo das métricas do BDFisc.....	53

LISTA DE TABELAS

Tabela 1 - Estrutura da tabela de validação.	41
Tabela 2 – Separação das cargas para os 4 servidores.....	43
Tabela 3 – Estrutura da tabela “cargaTemp”.	46
Tabela 4 – Amostra das cargas executadas na versão BDFisc.....	51
Tabela 5 – Amostra das do Projeto Piloto executadas na versão BDFisc 2.0.	52
Tabela 6 - Métricas sobre as cargas do BDFisc.	53

LISTA DE GRÁFICOS

Gráfico 1 – Cargas efetuadas por dia.....	54
Gráfico 2 - Notas carregadas por dia.....	54
Gráfico 3 - Itens carregados por dia.....	55

LISTA DE ABREVIATURAS E SIGLAS

ATF	Sistema de Administração Tributária e Financeira
ETL	Extração, Transformação e Carga
OS	Ordem de Serviço
SEFAZ	Secretária de Estado da Fazenda
SSAS	SQL Server Analysis Service
SSIS	SQL Server Integration Service
SSMS	SQL Server Management Studio
SSRS	SQL Server Reporting Service

SUMÁRIO

1	INTRODUÇÃO	15
2	JUSTIFICATIVA.....	17
3	OBJETIVOS	18
3.1	Objetivo geral.....	18
3.2	Objetivos Específicos	18
4	FUNDAMENTAÇÃO TEÓRICA	19
4.1	Integração de Dados	19
4.2	Data Warehouse	20
4.3	Data Lake.....	21
4.4	Banco de dados.....	22
4.4.1	<i>Modelagem dos dados.....</i>	<i>23</i>
4.4.1.1	Modelo Conceitual.....	23
4.4.1.2	Modelo Lógico	23
4.4.1.3	Modelo Físico	24
4.4.2	<i>Linguagem SQL.....</i>	<i>26</i>
4.5	Plataforma Microsoft para Integração e visualização de dados.....	27
4.5.1	<i>SQL Server Integration Service.....</i>	<i>27</i>
4.5.2	<i>SQL Server Management Studio</i>	<i>28</i>
4.5.3	<i>SQL Server Analysis Service.....</i>	<i>29</i>
4.5.4	<i>SQL Server Reporting Service</i>	<i>30</i>
5	MÉTODOS	32
5.1	Diagrama de caso de uso	32
5.2	Diagrama de atividades.....	33
5.3	Visão geral sobre o fluxo das cargas	34
5.4	Refatoração e melhorias do projeto SSIS	36
5.4.1	<i>Melhorias na estratégia de carga.....</i>	<i>36</i>
5.4.2	<i>Melhorias na estratégia de validação das cargas.....</i>	<i>38</i>
5.4.3	<i>Melhorias nos padrões do projeto</i>	<i>41</i>
5.4.4	<i>Integração do novo servidor</i>	<i>43</i>

5.5	Gerenciamento do consumo de disco.....	43
5.6	Automação das cargas	45
6	RESULTADO E DISCUSSÕES	47
6.1	BDFisc 2.0	47
6.2	Peculiaridade das cargas.....	49
6.3	Resultados	50
6.3.1	<i>BDFisc 2.0 x BDFisc</i>	50
6.3.2	<i>BDFisc</i>	52
7	CONCLUSÃO	56
7.1	Trabalhos Futuros	56
	REFERÊNCIAS.....	57

1 INTRODUÇÃO

Desde o início da sociedade moderna Brasileira a arrecadação de impostos tem um papel muito importante, pois é através desta atividade que os estados e municípios conseguem pagar as despesas da máquina pública e oferecer a estrutura necessária para a população no que diz respeito à educação, saúde, lazer, cultura, segurança entre outros. A cobrança de tributos está associada à ideia de que a sociedade deve contribuir para que todo cidadão, seja ele rico ou pobre, tenha direito a um ensino de boa qualidade, tratamento adequado das suas enfermidades, condições dignas de moradia entre outros. (CDES, 2011).

A Secretaria de Estado da Fazenda (SEFAZ) desempenha um papel muito importante no gerenciamento das finanças públicas a nível estadual, atividades como arrecadação, tributação, análise, estudo, pesquisa e investigação fiscal estão a cargo da SEFAZ (PARAÍBA, 2017). Dentre os principais impostos arrecadados podemos destacar: ICMS (Imposto sobre circulação de mercadorias e serviços); IPVA (Imposto sobre a propriedade de veículos automotores); ITCD (Imposto sobre transmissão “causa mortis” e doações) (NFE.IO, 2022).

Relatada a importância da arrecadação de tributos estaduais, faz-se necessário o uso de ferramentas tecnológicas que venham facilitar e auxiliar os profissionais na hora de realizar a fiscalização das empresas, pois da maneira que as informações necessárias para fiscalização são armazenadas os dados encontram-se dissipados e rebuscados, dificultando assim a análise dessas informações por parte dos fiscais.

A parte de fiscalização no contexto da SEFAZ do estado da Paraíba é feita através de ordens de serviço que são abertas no sistema chamado ATF, lá são agrupadas informações referentes ao contribuinte ou empresa que será fiscalizado.

Existe um projeto em andamento na SEFAZ da Paraíba denominado de BDFisc que tem como principal função auxiliar os fiscais no processo de fiscalização tributária das empresas reunindo as informações necessárias em um único repositório para que os fiscais possam trabalhar. O projeto era mantido por uma empresa terceirizada contratada pela SEFAZ, porém apenas uma pessoa era responsável por manter o projeto em funcionamento e fazer atualizações quando

necessário. Com a saída dessa pessoa da empresa o projeto ficou defasado e parado por alguns meses. Nesse ínterim foi feita uma parceria UEPB/SEFAZ para que o BDFisc fosse restabelecido com melhorias e novas percepções das tecnologias utilizadas.

Portanto, o presente trabalho tem como objetivo restabelecer o BDFisc que encontrava-se defasado e sem funcionar. De modo geral o projeto desempenha as seguintes atividades: extração dos dados das bases da SEFAZ, transformações nos dados com intuito de facilitar a compreensão dos mesmos, agrupamento e armazenamento em uma base de dados onde todos os fiscais possam exercer o seu trabalho seja realizando consultas diretamente nos bancos de dados, importando os dados para planilhas, ou outras formas de visualizações.

2 JUSTIFICATIVA

O presente trabalho foi construído com intuito de reformular e restabelecer o BDFisc projeto que auxilia os agentes públicos na fiscalização tributária, atividade que consiste na averiguação, checagem que os fiscais exercem com a finalidade de comprovar se as obrigações tributárias dos contribuintes estão sendo cumpridas conforme a lei. De acordo com os dados da CDL de João Pessoa, o índice de inadimplentes da Paraíba de janeiro até junho deste ano (2022) cresceu uma média de 5.26% em comparação a média do mesmo período do ano passado.

Após o responsável por manter o BDFisc sair da empresa que a SEFAZ tinha contratado para desenvolver o projeto, buscou-se dar continuidade ao projeto ainda com essa mesma empresa só que não houve muitos avanços e o projeto ficou parado por alguns meses até ser firmada uma parceria da SEFAZ com a UEPB para dar continuidade ao projeto com a adição de novas percepções e melhorias relacionadas ao tempo de carga, gerenciamento de espaço e corretude dos dados.

Com base nos dados citados, dá-se a importância de uma ferramenta que seja eficaz e auxilie os profissionais da área a realizarem a fiscalização dos contribuintes no tocante ao cumprimento de suas responsabilidades fiscais. Nesse sentido, buscamos inicialmente compreender o projeto, pois ele não possuía documentação, posteriormente trabalhamos em restabelecer o projeto com diversas melhorias e fazer com que trabalhasse de forma 100% automatizada e requeresse o mínimo de intervenção de nossa parte no gerenciamento das cargas.

3 OBJETIVOS

3.1 Objetivo geral

O objetivo geral deste trabalho é reformular, restabelecer e automatizar uma solução tecnológica que auxilia no processo de fiscalização tributária exercido pelos profissionais da SEFAZ do estado da Paraíba.

3.2 Objetivos Específicos

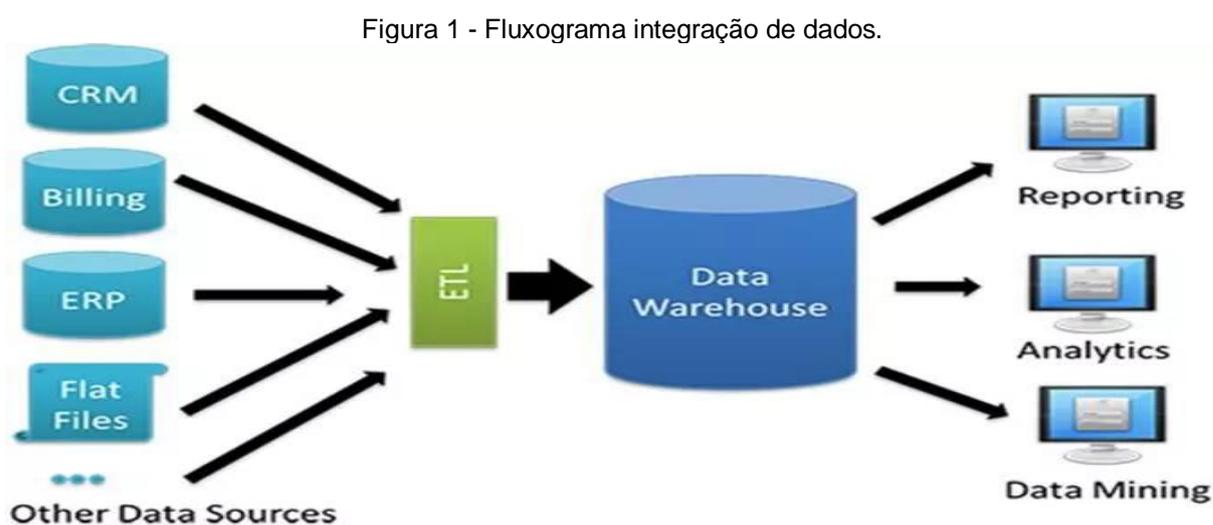
- Realização das cargas com os dados necessários para fiscalização das ordens de serviços;
- Realização da automatização da solução proposta para que funcione a maior parte do tempo sem a necessidade de alguém monitorando as cargas;
- Gerenciamento dos espaços em disco consumidos pelas cargas.

4 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo será apresentado a fundamentação teórica necessária para realização deste trabalho, ferramentas utilizadas, conceitos, metodologia de integração de dados e linguagem de pesquisa.

4.1 Integração de Dados

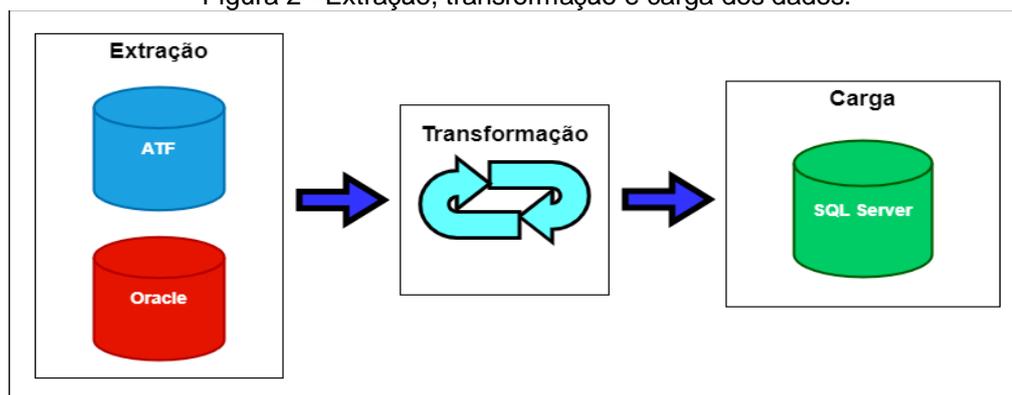
A integração de dados é um desafio comum quando se trabalha com dados advindos de sistemas autônomos e fontes de dados distintas (Halevy; Rajaraman; Ordille, 2006, p. 09), o projeto BDFisc foi construído para realizar a integração dos dados que advém de fontes heterogêneas através do conceito de ETL, que significa extrair, transformar e carregar. É um processo de integração de dados que combina dados de várias fontes de dados em um único armazenamento de dados consistente que é carregado em um data warehouse ou outro sistema de destino (IBM, 2020). A Figura 1 ilustra bem como uma aplicação baseada na integração de dados usando o conceito de ETL deve funcionar.



Fonte: <https://www.astera.com/pt/tipo/blog/defini%C3%A7%C3%A3o-de-data-warehouse/>

Nesse contexto de integração de dados, existem as ferramentas que nos auxiliam a implementar esse tipo de conceito. O projeto foi implementado majoritariamente utilizando a ferramenta da Microsoft chamada SQL Server Integration Service (SSIS) através dela foi possível pôr em prática o conceito de ETL, a Figura 2 mostra o fluxo que os dados percorrem.

Figura 2 - Extração, transformação e carga dos dados.



Fonte: Elaborada pelo Autor, 2022.

4.2 Data Warehouse

De acordo com Inmon (2005, p. 29) “Um data warehouse é um conjunto de dados baseados em assuntos, integrado, não volátil e variável em relação ao tempo, de apoio às decisões gerenciais.”. Trazendo para o contexto do projeto o data warehouse é onde são armazenadas todas as informações necessárias para realização da fiscalização de um determinado contribuinte. A Figura 3 mostra um exemplo de como estão organizados os dados em um banco que implementa este conceito.

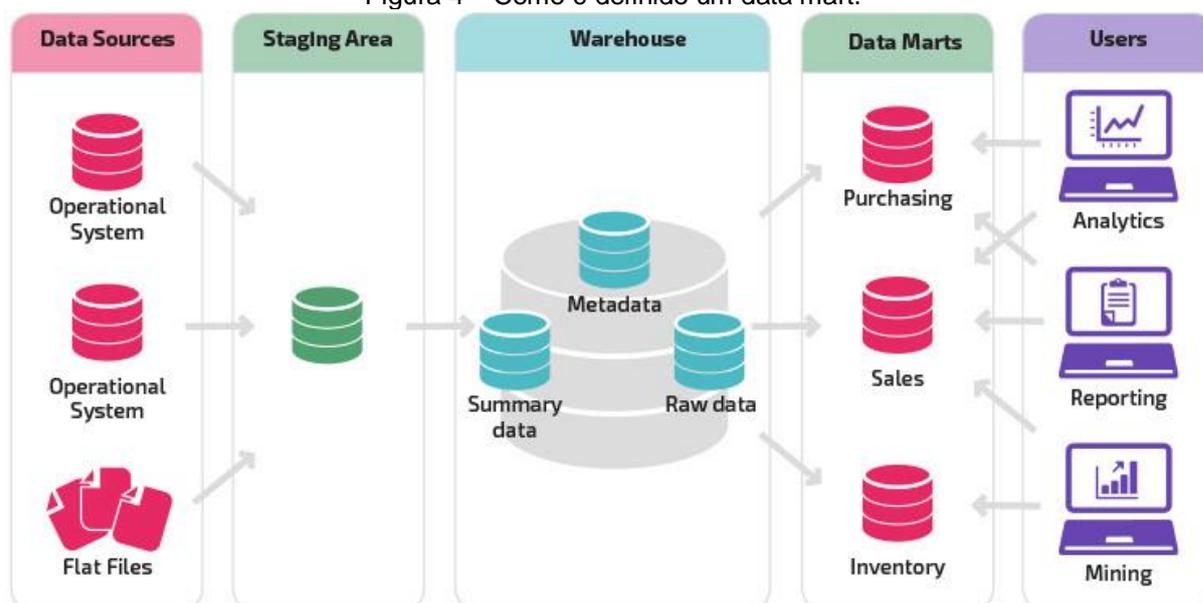
Figura 3 – Estrutura de um banco utilizado na fiscalização.

- + [ícone] dbo.ITEM_NFENVE
- + [ícone] dbo.ITEM_NFEPC
- + [ícone] dbo.ITEM_NFEUFDES
- + [ícone] dbo.ITEM_NFEVEIC
- + [ícone] dbo.NFE
- + [ícone] dbo.NFE_AUTORIZAXML
- + [ícone] dbo.NFE_EVENTO
- + [ícone] dbo.NFE_EVENTO_DETALHADO
- + [ícone] dbo.NFE_EVENTO_log
- + [ícone] dbo.NFE_EVENTO_TIPO
- + [ícone] dbo.NFE_FISCOEMIT
- + [ícone] dbo.NFE_GRP CANA
- + [ícone] dbo.NFE_GRPDEDCANA
- + [ícone] dbo.NFE_GRPPROCREF
- + [ícone] dbo.NFE_HI

Fonte: Elaborado pelo Autor, 2022.

Ainda dentro desse contexto, existe o data mart, que podemos definir como sendo um subconjunto de um data warehouse focado em uma determinada linha de negócio, departamento ou área de assunto (IBM, 2020). A Figura 4 demonstra um exemplo de data mart.

Figura 4 – Como é definido um data mart.



Fonte: <https://panoply.io/data-warehouse-guide/data-mart-vs-data-warehouse/>

4.3 Data Lake

Embora os conceitos se confundam um pouco, existem algumas características que diferenciam o data lake do data warehouse. Podemos definir um data lake como sendo

um repositório centralizado que permite armazenar todos os dados sejam eles estruturados ou não em qualquer escala (AWS, 2022). A Figura 5 mostra as principais características que diferem esses dois conceitos.

Na conjuntura do BDFisc podemos definir como um data lake o próprio ATF, pois nele existem informações advindas de diversas fontes e são dados brutos dos contribuintes, xmls de notas fiscais e documentos eletrônicos.

Figura 5 – Diferença entre data lake e data warehouse

Características	Data warehouse	Data lake
Dados	Relacionais de sistemas transacionais, bancos de dados operacionais e aplicações de linha de negócios	Não relacionais e relacionais de dispositivos de IoT, sites, aplicações móveis, mídia social e aplicações corporativas
Esquema	Definido antes da implementação do DW (esquema na gravação)	Gravado no momento da análise (esquema na leitura)
Preço/performance	Resultados de consulta mais rápidos, usando armazenamento de maior custo	Resultados de consulta ficando mais rápidos, usando armazenamento de menor custo
Qualidade dos dados	Dados altamente selecionados, que representam a versão central da verdade	Quaisquer dados, selecionados ou não (ou seja, dados brutos)
Usuários	Analistas de negócios	Cientistas de dados, desenvolvedores de dados e analistas de negócios (usando dados selecionados)
Análises	Geração de relatórios em lote, BI e visualizações	Machine learning, análises preditivas, descoberta de dados e criação de perfis

Fonte: <https://aws.amazon.com/pt/big-data/datalakes-and-analytics/what-is-a-data-lake/>

4.4 Banco de dados

Segundo (DATE, 2004, p. 06) podemos definir como banco de dados “um sistema computadorizado de manutenção de registros; em outras palavras, é um sistema cuja finalidade geral é armazenar informações e permitir que os usuários busquem e atualizem essas informações quando as solicitar”.

Dentro do universo dos bancos de dados existem os softwares que nos ajudam a realizar todas as configurações e gerenciamento dessas bases de dados. Essas ferramentas são chamadas de sistema gerenciador de banco de dados (SGBD), alguns exemplos de SGBDs são: Oracle, SQL Server, MySQL, PostgreSQL, entre outros.

Para criação de um sistema que utiliza banco de dados é importante que sejam seguidos alguns padrões de projeto com intuito de organizar as informações e técnicas que serão utilizadas para garantir uma boa performance e uma fácil manutenção no futuro, a subseção 4.4.1 trata de uma fase importante que deve ser respeitada na criação de um projeto de banco de dados.

4.4.1 Modelagem dos dados

Podemos definir como modelagem de dados o processo de analisar e definir todos os dados que serão utilizados em um projeto, é nesta fase que são criadas representações visuais dos dados conforme a lógica do negócio. Podemos dividir a modelagem de dados em três grupos que serão detalhados nas subseções a seguir.

4.4.1.1 Modelo Conceitual

Como o próprio nome sugere, esse é um modelo de alto nível dos dados que tem como principal função a captação dos requisitos e regras de negócio que envolvem os dados. Ao criar um projeto, esta deve ser a fase inicial onde serão desenvolvidos alguns artefatos, como por exemplo, um diagrama de entidade-relacionamento que pode ser visto na Figura 6.

Figura 6 – Diagrama de entidade-relacionamento



Fonte:

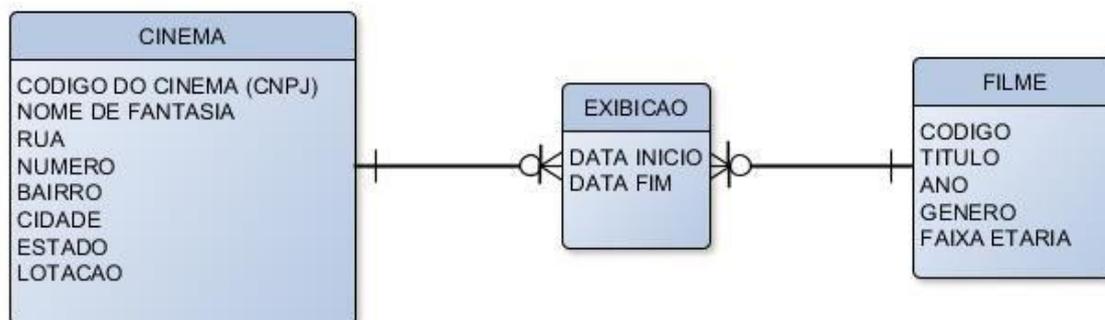
https://livrodigital.uniasselvi.com.br/GTI11_principios_de_banco_de_dados/unidade2.html?topico=1

4.4.1.2 Modelo Lógico

O modelo lógico define as entidades, tipos dos dados, atributos e relacionamentos das entidades. De modo geral esta etapa busca representar as estruturas que irão armazenar os dados dentro do banco de dados. A criação do modelo lógico deve ser iniciada apenas após a estruturação do modelo conceitual.

Ainda nesta etapa será definida a abordagem utilizada referente ao banco de dados podendo ser relacional, não-relacional, hierárquico, rede, entre outros. A Figura 7 mostra um exemplo de diagrama desenvolvido utilizando uma abordagem relacional.

Figura 7 – Diagrama desenvolvido no modelo lógico



Fonte:

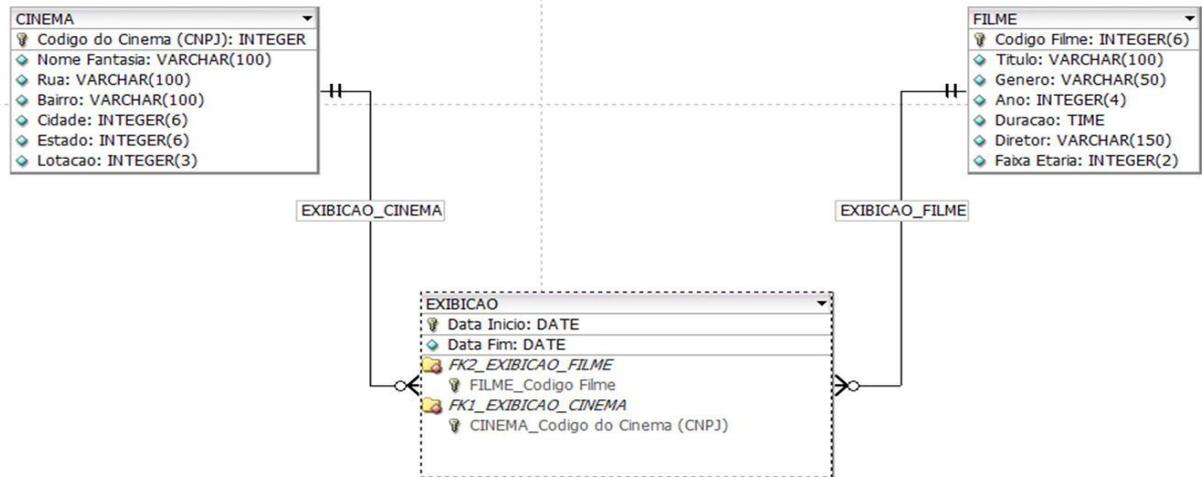
https://livrodigital.uniasselvi.com.br/GTI11_principios_de_banco_de_dados/unidade2.html?topico=1

4.4.1.3 Modelo Físico

Por fim, no modelo físico utilizamos os artefatos desenvolvidos nas etapas anteriores para modelagem física do banco de dados, aqui serão levadas em consideração questões relacionadas ao tamanho dos campos, tipos de dados, indexação, relacionamentos, restrições de acesso, entre outras coisas.

O SGBD é um grande aliado nessa etapa, pois podemos utilizá-lo para desenhar o diagrama do modelo físico e a partir dele a ferramenta já disponibiliza o código e toda a lógica envolvida para que sejam criadas as tabelas e seus relacionamentos. A Figura 8 mostra um exemplo desse tipo de diagrama.

Figura 8 – Diagrama desenvolvido no modelo físico



Fonte:

https://livrodigital.uniasselvi.com.br/GT111_principios_de_banco_de_dados/unidade2.html?topico=1

A Figura 9 mostra um exemplo de um trecho de código que o próprio SGBD gera para criação das tabelas a partir do diagrama do modelo físico.

Figura 9 – Exemplo de código gerado pelo SGBD a partir do diagrama do modelo físico

```
CREATE TABLE FILME (
Codigo Filme INTEGER(6) NOT NULL,
Titulo VARCHAR(100),
Genero VARCHAR(50),
Ano INTEGER(4),
Duracao TIME,
Diretor VARCHAR(150),
FaixaEtaria INTEGER(2) UNSIGNED,
PRIMARY KEY(Codigo Filme)
);

CREATE TABLE CINEMA (
Codigo do Cinema (CNPJ) INTEGER NOT NULL,
Nome Fantasia VARCHAR(100),
Rua VARCHAR(100),
Bairro VARCHAR(100),
Cidade INTEGER(6),
Estado INTEGER(6),
Lotacao INTEGER(3),
PRIMARY KEY(Codigo do Cinema (CNPJ))
)
```

Fonte:

https://livrodigital.uniasselvi.com.br/GT111_principios_de_banco_de_dados/unidade2.html?topico=1

4.4.2 Linguagem SQL

A linguagem SQL pode ser definida como uma linguagem declarativa, e sua criação é baseada em álgebra e cálculo relacional. Sua sintaxe é de fácil compreensão, com ela é possível não só manipular dados como também realizar a definição de estruturas de dados, as regras de manipulação e restrições de integridade (CARDOSO, V e CARDOSO, G, 2013, p. 9). De acordo com o ranking do site DB-Engines, dentre os top 10 SGBDS (Sistema de gerenciamento de banco de dados), 7 usam o modelo relacional de dados onde a linguagem predominante é a SQL ou variações da mesma, a figura abaixo mostra o top 10 desse ranking.

Figura 10 - Ranking SGBDS de acordo com a popularidade.

Rank			DBMS	Database Model	Score		
Jul 2022	Jun 2022	Jul 2021			Jul 2022	Jun 2022	Jul 2021
1.	1.	1.	Oracle	Relational, Multi-model	1280.30	-7.44	+17.63
2.	2.	2.	MySQL	Relational, Multi-model	1194.87	+5.66	-33.51
3.	3.	3.	Microsoft SQL Server	Relational, Multi-model	942.13	+8.30	-39.83
4.	4.	4.	PostgreSQL	Relational, Multi-model	615.87	-4.97	+38.72
5.	5.	5.	MongoDB	Document, Multi-model	472.98	-7.74	-23.18
6.	6.	6.	Redis	Key-value, Multi-model	173.62	-1.69	+5.32
7.	7.	7.	IBM Db2	Relational, Multi-model	161.22	+2.03	-3.94
8.	8.	8.	Elasticsearch	Search engine, Multi-model	154.33	-1.67	-1.43
9.	9.	11.	Microsoft Access	Relational	145.09	+3.27	+31.64
10.	10.	9.	SQLite	Relational	136.68	+1.24	+6.47

Fonte: <https://db-engines.com/en/ranking>

Podemos dividir a linguagem SQL em módulos que tratam de partes específicas, que são:

- **DLL** (Data Definition Language - linguagem de definição de dados), de modo geral é quem rege as regras relacionadas aos metadados tais como, criação, exclusão e alteração de esquemas dos bancos de dados;
- **DML** (Data Manipulation Language - linguagem de manipulação de dados), referente a manipulação dos dados, ou seja, inclusão, alteração e exclusão dos dados;
- **DCL** (Data Control Language - linguagem de controle dos dados), dita as regras que regem no tocante a permissão e autorização de acesso dos usuários aos dados.

- **DTL** (Data Transaction Language - linguagem de transação de dados), relativo aos comandos fornecidos para trabalhar com transações;
- **DQL** (Data Query Language - linguagem de consulta de dados), que fornece a estrutura necessária para realizar as consultas nos dados.

4.5 Plataforma Microsoft para Integração e visualização de dados

Quando se trata de integração de dados existem muitas ferramentas no mercado que podem nos ajudar, entretanto o projeto BDFisc desde o início utiliza a plataforma da Microsoft que fornece ferramentas para tratar desde a integração dos dados, armazenamento até a visualização.

Falaremos um pouco das ferramentas que a plataforma da Microsoft oferece para trabalhar com dados, nas subseções 4.5.1 e 4.5.2 falamos um pouco da ferramenta que utilizamos para realização da integração dos dados e para o armazenamento dos dados respectivamente. A parte de visualização, relatórios e afins ficam a cargo do usuário final.

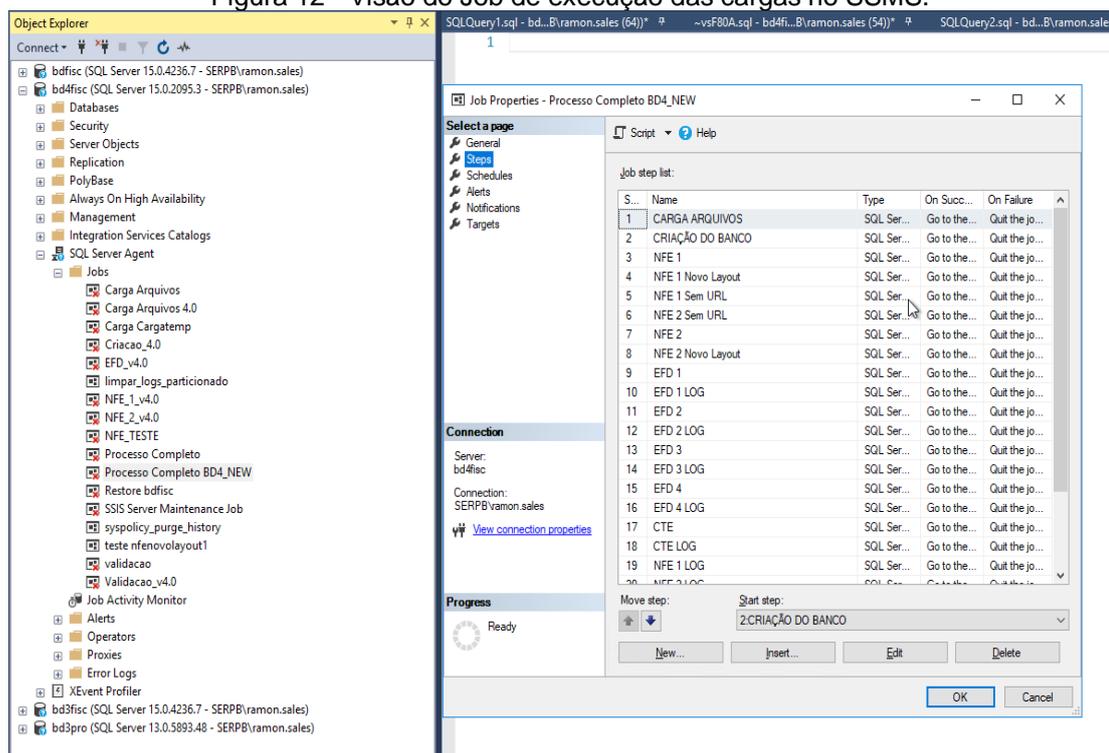
4.5.1 SQL Server Integration Service

O SSIS (SQL Server Integration Service) é uma plataforma utilizada para diversas atividades relacionadas a dados, sendo através dela possível criar soluções de integração, transformação, limpeza e mineração de dados (MICROSOFT, 2022). Essa ferramenta é amplamente utilizada para realização de ETL, além de fornecer o que é preciso para realização do gerenciamento de bases de dados SQL Server.

A utilização do SSIS é feita através do Microsoft Visual Studio que nada mais é que uma IDE (ambiente de desenvolvimento integrado). A Figura 11 demonstra uma visão macro de um pacote que executa diversas consultas cada qual

representada pelos retângulos que são chamados de containers, os dados são buscados em uma base de dados Oracle e inseridos na base de dados SQL Server.

Figura 12 - Visão do Job de execução das cargas no SSMS.



Fonte: Elaborada pelo Autor, 2022.

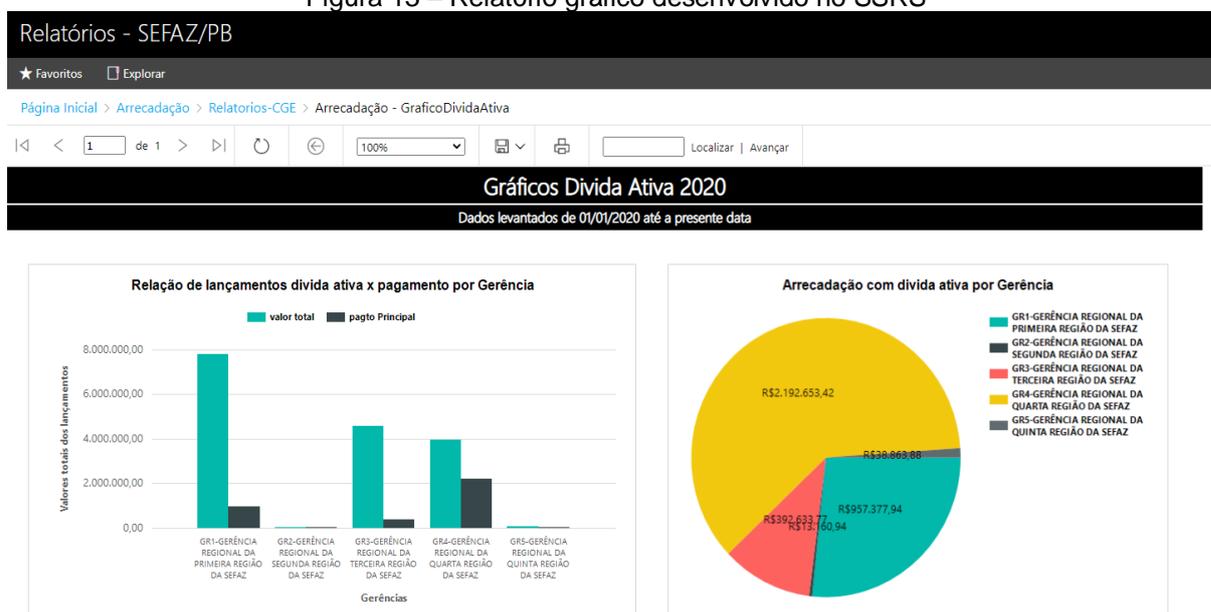
4.5.3 SQL Server Analysis Service

O SSAS (SQL Server Analysis Service) é um mecanismo de dados analíticos e é utilizado no suporte à decisão e na análise de negócios (MICROSOFT, 2022). O SSAS fornece uma plataforma com todas as ferramentas necessárias para desenvolvimento de soluções analíticas online (OLAP). De modo geral o SSAS é utilizado para criação de cubos, os cubos nada mais são do que fontes de dados multidimensionais onde possuem tabelas de dimensões e tabelas de fato (onde estão armazenadas as medidas), os cubos utilizam data marts/data warehouse para uma análise de dados rápida, profunda e eficiente.

4.5.4 SQL Server Reporting Service

Finalizando a tríplice de ferramentas para trabalhar com dados da Microsoft, temos o SSRS (SQL Server Reporting Service) pode ser definido como um conjunto de ferramentas e serviços que nos auxiliam na criação, implantação e gerenciamento de relatórios paginados e móveis (MICROSOFT, 2022). Através dele é possível criar relatórios como os que são mostrados nas Figuras 13 e 14.

Figura 13 – Relatório gráfico desenvolvido no SSRS



Fonte: Elaborado pelo Autor, 2022.

Figura 14 – Relatório desenvolvido no SSRS

Relatórios - SEFAZ/PB

★ Favoritos □ Explorar

Página Inicial > Arrecadação > Relatorios-CGE > Arrecadação - RelatórioDividaAtiva

Ano Insc. Divida Ativa: 2020 Mês de Insc. Divida Ativa: 1 Ordenado por: Data de Vencimento Ordenação: Crescente

1 de 1 100% Localizar | Avançar

Relatório Divida Ativa 2020



	Nosso Número	Data Lançamento	Data Vencimento	Receita	IE	SitDebito	Data Operação	Operação	Qt Lanc	Valor Principal	Valor Multa	Valor Total	Saldo
Ano: 2020 Mes: 1									383	RS3.020.373,52	RS528.141,74	RS3.548.515,26	RS2.533.418,33
GR1-GERÊNCIA REGIONAL DA PRIMEIRA REGIÃO DA SEFAZ									112	RS921.873,75	RS130.587,53	RS1.052.471,28	RS562.152,34
CENTRO DE ATENDIMENTO AO CIDADÃO DA GR1 DA SEFAZ - JOÃO PESSOA									91	RS630.960,25	RS109.102,98	RS740.063,23	RS454.679,62
	3008342908	10/10/2014	15/10/2014	1101	162330081	A MENOR	20/01/2020	DIVIDA ATIVA	1	RS1.464,91	RS138,14	RS1.603,05	RS459,89
	3008491779	10/11/2014	15/11/2014	1101	162330081	A MENOR	20/01/2020	DIVIDA ATIVA	1	RS2.155,93	RS203,29	RS2.359,22	RS676,80
	3008659025	13/12/2014	15/12/2014	1101	162330081	A MENOR	20/01/2020	DIVIDA ATIVA	1	RS5.265,46	RS496,47	RS5.761,93	RS1.652,85
	3008800276	14/01/2015	15/01/2015	1101	162330081	A MENOR	20/01/2020	DIVIDA ATIVA	1	RS21.233,82	RS2.107,59	RS23.341,41	RS7.016,47
	3009703856	03/07/2015	15/02/2015	1101	162443706	A MENOR	21/01/2020	DIVIDA ATIVA	1	RS1.427,72	RS94,33	RS1.522,05	RS471,65
	3009706634	06/07/2015	15/03/2015	1101	162443706	A MENOR	21/01/2020	DIVIDA ATIVA	1	RS2.773,70	RS183,24	RS2.956,94	RS916,21
	3009669258	06/07/2015	15/04/2015	1101	162443706	A MENOR	21/01/2020	DIVIDA ATIVA	1	RS3.518,68	RS232,48	RS3.751,16	RS1.162,38
	3009284392	18/04/2015	15/04/2015	1101	161336620	A MENOR	22/01/2020	DIVIDA ATIVA	1	RS15.654,68	RS1.944,50	RS17.599,18	RS6.578,87
	3009665310	06/07/2015	15/05/2015	1101	162443706	A MENOR	21/01/2020	DIVIDA ATIVA	1	RS2.571,97	RS169,03	RS2.741,00	RS849,63
	3009708145	06/07/2015	15/06/2015	1101	162443706	A MENOR	21/01/2020	DIVIDA ATIVA	1	RS2.903,93	RS191,84	RS3.095,77	RS959,22
	3014281384	10/10/2017	10/08/2015	1101	161646395	A MENOR	20/01/2020	DIVIDA ATIVA	1	RS426,52	RS30,28	RS456,80	RS132,47
	3010216350	15/10/2015	15/10/2015	1101	161336620	A MENOR	22/01/2020	DIVIDA ATIVA	1	RS17.761,66	RS2.443,13	RS20.204,79	RS8.265,90
	3014281385	10/10/2017	10/11/2015	1101	161646395	A MENOR	20/01/2020	DIVIDA ATIVA	1	RS956,84	RS67,92	RS1.024,76	RS297,16
	3011760457	30/07/2016	15/12/2015	1101	160970440	A MENOR	22/01/2020	DIVIDA ATIVA	1	RS5.567,92	RS1.113,58	RS6.681,50	RS3.709,64

Fonte: Elaborado pelo Autor, 2022.

5 MÉTODOS

Neste capítulo, apresentamos os métodos utilizados para realização da implantação, refatoração e manutenção do sistema de análise fiscal, são abordados pontos como: arquitetura do projeto, estratégia de refatoração, melhorias no consumo de disco nas bases de dados que fazem o intermédio das cargas, implementação da automação das cargas, política de limpeza das bases de dados Oracle e SQL Server.

5.1 Diagrama de caso de uso

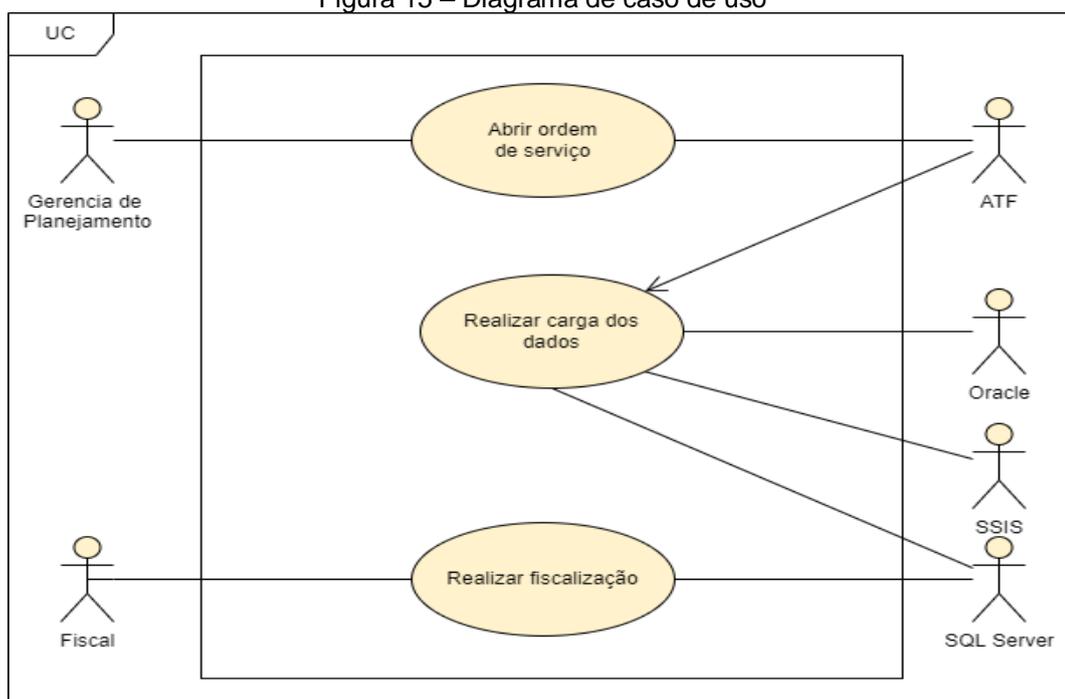
Com a finalidade de organizar todos os possíveis comportamentos do sistema, o diagrama de caso de uso exhibe esse fluxo a partir do uso de atores, funcionalidades e como eles se relacionam dentro do contexto do sistema. Em sua forma mais simples, um caso de uso identifica os atores envolvidos em uma interação e dá nome ao tipo de interação (SOMMERVILLE, 2011, p. 74).

Os elementos que compõem um diagrama de caso de uso são:

- **Atores:** Pessoas ou sistemas, representados por bonecos palitos
- **Classe de Iteração:** Funções ou ações dentro do sistema, representados por elipses.
- **Linhas:** Relacionam os atores com o sistema, podendo conter flechas para mostrar como essa interação se inicia.

O diagrama de caso de uso é um documento muito importante no tocante a documentação do sistema, porém a versão do BDFisc que nos foi passada não possuía nenhum tipo de documentação, desse modo foi necessário realizar uma atividade de engenharia reversa para identificar os atores e os demais componentes necessários para criação desse diagrama, a Figura 15 mostra o diagrama de caso de uso do BDFisc.

Figura 15 – Diagrama de caso de uso



Fonte: Elaborado pelo Autor, 2022.

O diagrama de caso de uso mostra uma visão geral do uso do projeto, os casos de uso descritos na imagem acima são detalhados no diagrama de atividades.

5.2 Diagrama de atividades

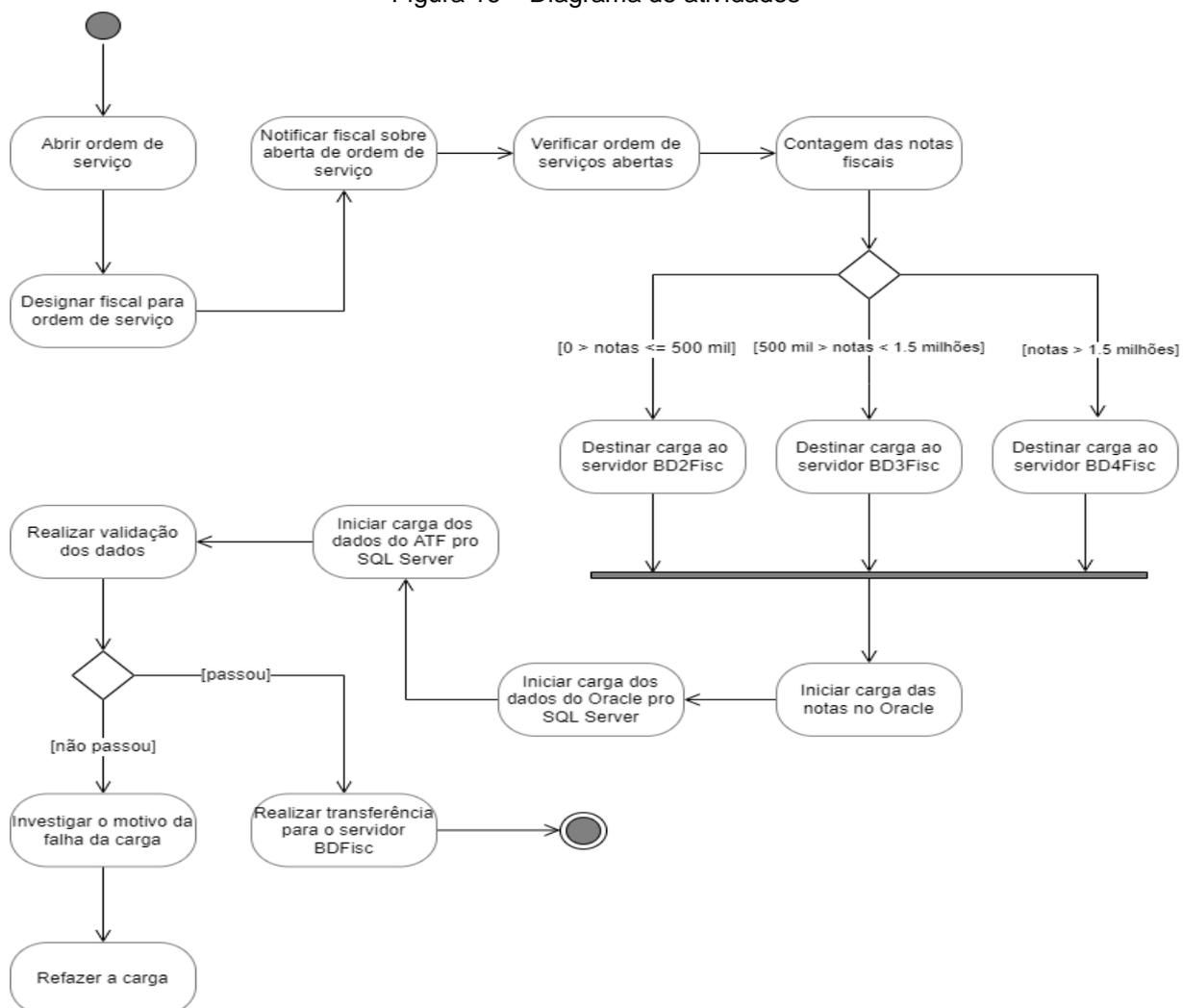
O diagrama de atividades tem como objetivo geral mostrar as atividades que compõem um processo de um sistema e o fluxo de controle de uma atividade para outra (SOMMERVILLE, 2011, p. 85).

Os elementos que compõem um diagrama de atividade são:

- **Círculo preenchido:** Início de um processo.
- **Círculo preenchido dentro de outro círculo:** Fim de um processo.
- **Retângulos com cantos arredondados:** Atividades.
- **Setas:** Fluxo de trabalho de uma atividade para outra.
- **Losango:** Decisão a ser tomada.

A Figura 16 mostra o diagrama que representa todas as atividades envolvidas no processo de carga dos dados.

Figura 16 – Diagrama de atividades



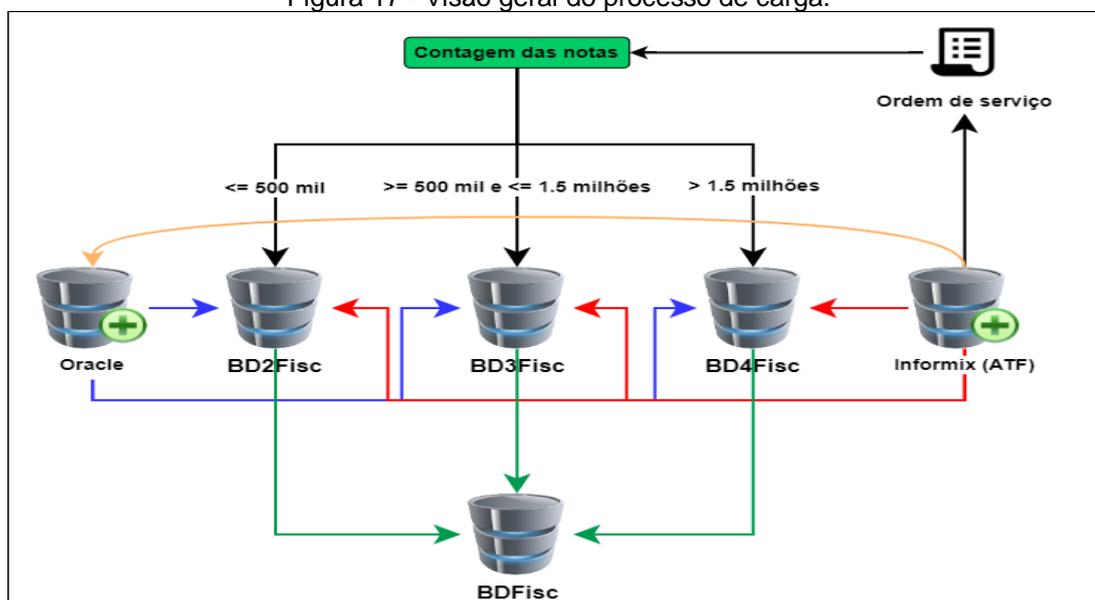
Fonte: Elaborado pelo Autor, 2022.

5.3 Visão geral sobre o fluxo das cargas

O projeto trabalha com seis bases de dados, duas são bases de produção onde são armazenadas as notas fiscais e informações dos contribuintes da Paraíba, uma Oracle e outra Informix (ATF). Na base Informix são abertas e armazenadas as ordens de serviço para fiscalização dos contribuintes e possui informações sobre os contribuintes e notas fiscais. O Oracle é uma base intermediária onde os dados são

enviados para serem carregados no SQL Server e apagados assim que os dados forem carregados. A Figura 17 relata o fluxo percorrido por uma carga.

Figura 17 - Visão geral do processo de carga.



Fonte: Elaborada pelo Autor, 2022.

O fluxo de carga acontece da seguinte maneira:

1. É aberto uma ordem de serviço (OS) no ATF;
2. O projeto SSIS identifica que foi aberta uma OS, faz a contagem das notas e insere os dados necessários para realização da carga no respectivo servidor;
3. É iniciada a carga das notas do ATF para o Oracle sinalizado na imagem com uma seta laranja, após a conclusão dessa carga é iniciada a carga do Oracle para o SQL Server sinalizado pela seta azul;
4. Logo após são carregados alguns documentos eletrônicos do ATF para o SQL Server sinalizado pela seta vermelha;
5. Quando a carga é finalizada do Oracle para o SQL Server é feito a validação das notas;
6. Após finalizada e validada a carga é inserida no BDFisc, onde os fiscais podem realizar o seu trabalho.
7. Posteriormente os dados referentes a carga realizada são apagados das bases do Oracle, tendo em vista que a função dela é ser uma área de “stage” onde os dados são armazenados temporariamente para serem transferidos para o SQL Server.

5.4 Refatoração e melhorias do projeto SSIS

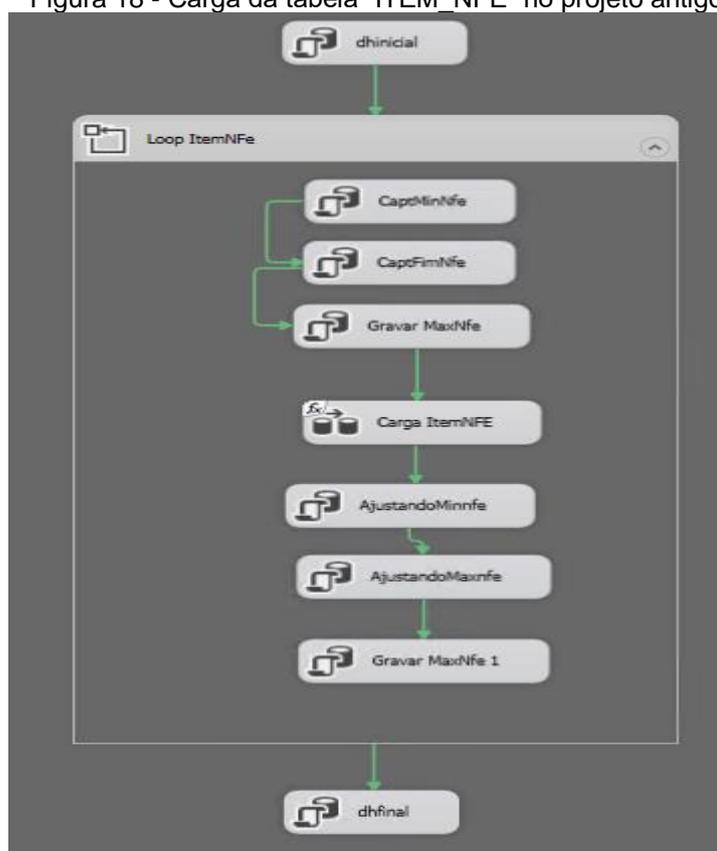
Houve diversas motivações no projeto antigo o que fez com que fosse criada uma versão do BDFisc chamada de BDFisc 2.0, dentre os principais motivos para refatoração do projeto podemos destacar que o projeto não funcionava corretamente, quando funcionava a validação das cargas não era confiável e as cargas levavam muito tempo para serem concluídas. Nesse ínterim os principais pontos trabalhados na parte de refatoração e melhorias no projeto SSIS foram:

1. Melhorias na estratégia de carga;
2. Melhorias na estratégia de validação das cargas;
3. Melhorias nos padrões do projeto;
4. Integração de um novo servidor para carga.

5.4.1 Melhorias na estratégia de carga

Assim que recebemos o projeto antigo foi feito um estudo para identificarmos os problemas que faziam com que o projeto não funcionasse. A princípio foi observado que o projeto continha vários recursos desnecessários para o funcionamento das cargas, tendo em vista que as cargas são realizadas de forma sequencial, essa estratégia foi adotada devido uma limitação de espaço em disco tanto no Oracle como no SQL Server. A Figura 18 demonstra como estava sendo realizada a carga para a tabela "ITEM_NFE" no projeto antigo.

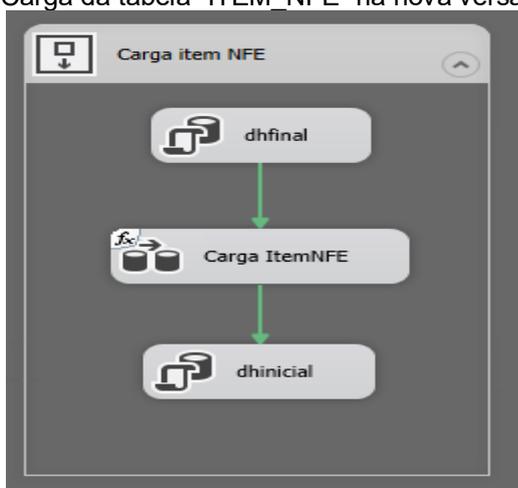
Figura 18 - Carga da tabela "ITEM_NFE" no projeto antigo.



Fonte: Elaborada pelo Autor, 2022.

Essa abordagem causava uma série de problemas, pois era realizada através de um laço de repetição controlada a partir do mínimo e do máximo da chave da nota fiscal, em termos simples era controlada a quantidade de itens carregados por vez. Essa abordagem se mostrou desnecessária, pois o próprio SSIS já realiza a carga dos itens de forma eficiente, basta que seja especificada na consulta a chave da nota fiscal para a qual os itens devem ser carregados, desse modo não é necessário se preocupar com a quantidade de itens que será carregado, pois a própria ferramenta já faz isso de forma rápida e eficiente. A Figura 19 mostra como ficou a carga da tabela "ITEM_NFE" na nova versão do projeto, onde foi retirado todo o laço de repetição e as consultas que capturavam os máximos e mínimos das chaves dos itens das notas fiscais, otimizando assim o tempo de carga.

Figura 19 - Carga da tabela "ITEM_NFE" na nova versão do projeto.

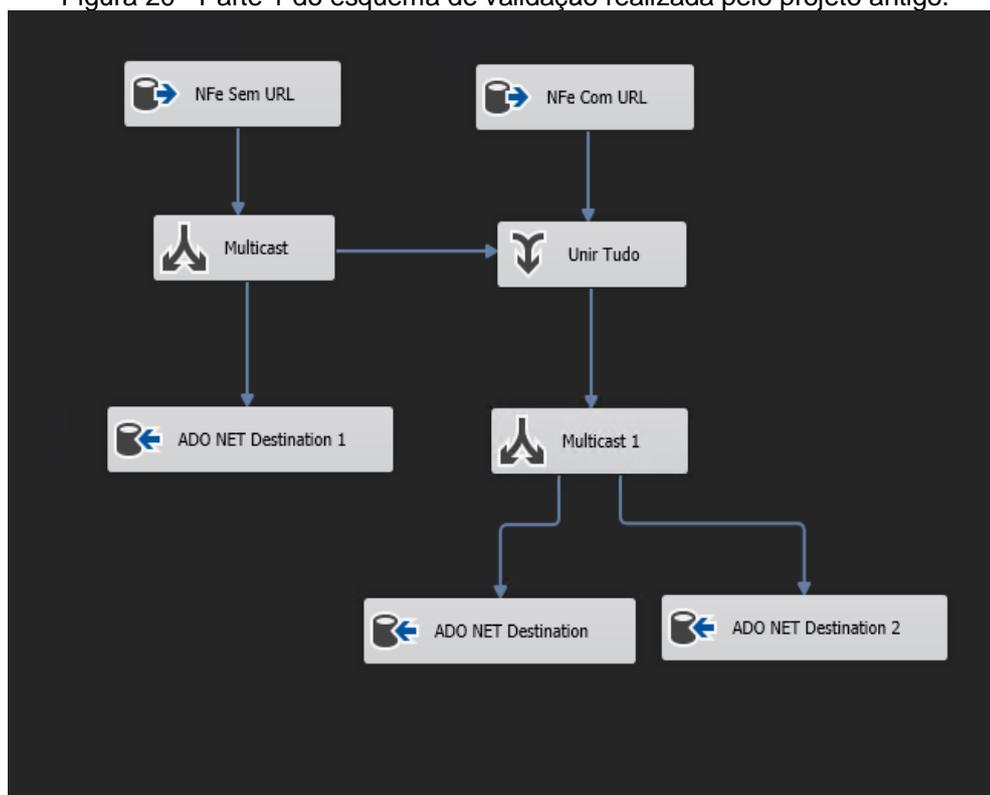


Fonte: Elaborada pelo Autor, 2022.

5.4.2 Melhorias na estratégia de validação das cargas

A versão antiga do projeto realizava as validações utilizando uma estratégia problemática que acabava fazendo com que os dados dessa não fossem confiáveis além disso acrescentava um tempo desnecessário nas cargas. Essa abordagem era dividida em duas partes. A primeira parte consistia em realizar a inserção de dados no momento da carga de uma determinada tabela, em uma tabela auxiliar usada para contagem das notas. Ou seja, suponhamos que vamos fazer a carga da tabela de notas fiscais chamada de "NFE" e que essa tabela possua 500 mil notas, além de inserção na própria tabela que irá ser utilizada para as análises fiscais era inserido dados em uma tabela chamada "sq_nota_validacao" que era utilizada para validação isso acrescentavam um tempo desnecessário na carga. A Figura 20 demonstra a parte 1 dessa validação no projeto antigo.

Figura 20 - Parte 1 do esquema de validação realizada pelo projeto antigo.



Fonte: Elaborada pelo Autor, 2022.

Como podemos ver na imagem os dados eram inseridos em duas tabelas adicionais além da tabela referente à carga, isso acrescentava um tempo desnecessário nas cargas, a parte 2 dessa validação era realizar a contagem na tabela “sq_nota_validacao” e inserir na tabela que de fato que é utilizada para checagem da validação das notas. A Figura 21 mostra o script de inserção utilizado na contagem.

Figura 21 - Parte 2 do esquema de validação realizada pelo projeto antigo

```

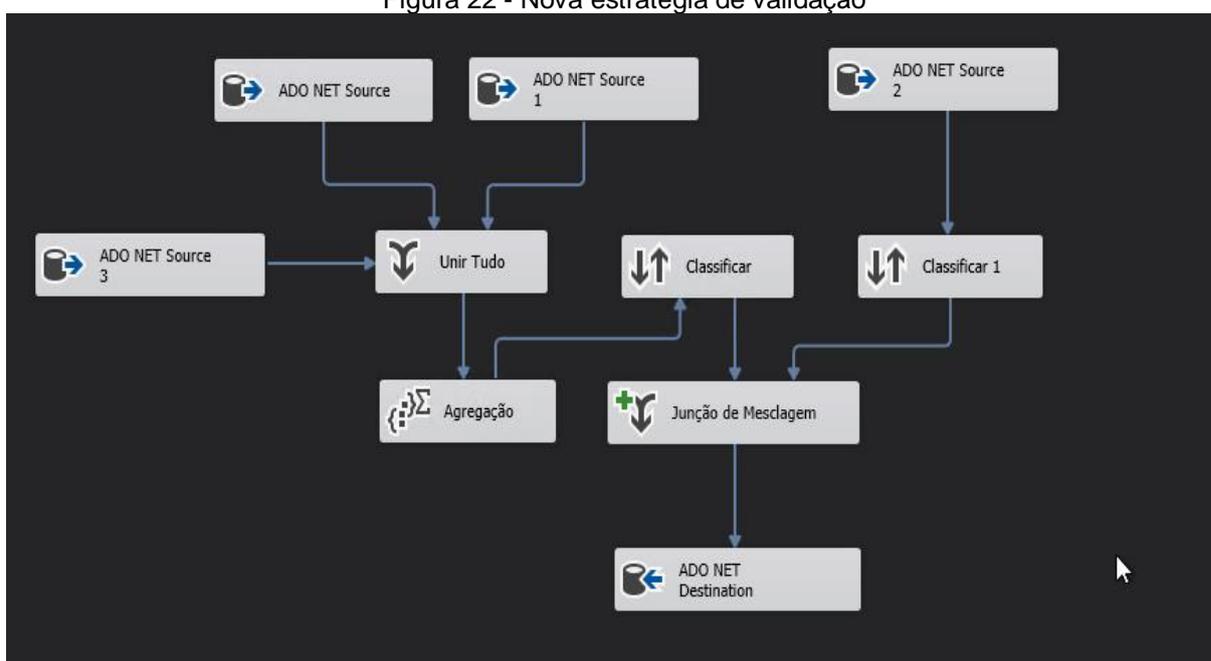
insert into sql_log_validacao(tabela,min_origem,max_origem,count_origem,
min_destino,max_destino,count_destino)
values ('NFE',(select isnull(min(sqnf),0) from sqnota_validacao where tabela = 'NFE'),
(select isnull(max(sqnf),0) from sqnota_validacao where tabela = 'NFE'),
(select isnull(count(sqnf),0) from sqnota_validacao where tabela = 'NFE'),
(select isnull(min(sqnf),0) from NFE), (select isnull(max(sqnf),0) from NFE),
(select isnull(count(sqnf),0) from NFE))
  
```

Fonte: Elaborada pelo Autor, 2022.

Dessa forma, a validação era feita para todas as tabelas do projeto e era analisada por quem realizou a carga, não possuía nenhum script para impedir que essa carga não fosse para o BDFisc caso a validação falhasse, por este motivo essa estratégia se mostrava ineficiente e não confiável.

Implementamos uma nova estratégia de validação que se mostrou mais eficiente e que consome menos tempo da carga. A técnica utilizada consiste em comparar a quantidade de notas presente no Oracle e a quantidade de notas carregadas no SQL Server. Realizamos a contagem e inserimos em uma tabela chamada “sq_log_validacao”, posteriormente é feita uma checagem nessa tabela e a carga só vai para o BDFisc caso passe na validação. A Figura 22 mostra a maneira que a validação é feita atualmente.

Figura 22 - Nova estratégia de validação



Fonte: Elaborada pelo Autor, 2022.

Primeiro é feita a contagem no Oracle referente a tabela em questão, como são 3 modelos de notas fiscais são realizadas 3 consultas para contagem dessas notas. Posteriormente, é feita uma união do resultado das consultas, depois uma agregação por soma, máximo e mínimo e depois uma ordenação, em seguida é feita a contagem das notas carregadas na tabela do SQL Server, logo após uma junção desses dados e uma inserção na tabela de validação já citada anteriormente. A estrutura dessa tabela de validação é mostrada na Tabela 1, onde a coluna “count_origem” refere-se às notas contadas no Oracle e “count_destino” as notas carregadas no SQL Server.

Tabela 1 - Estrutura da tabela de validação.

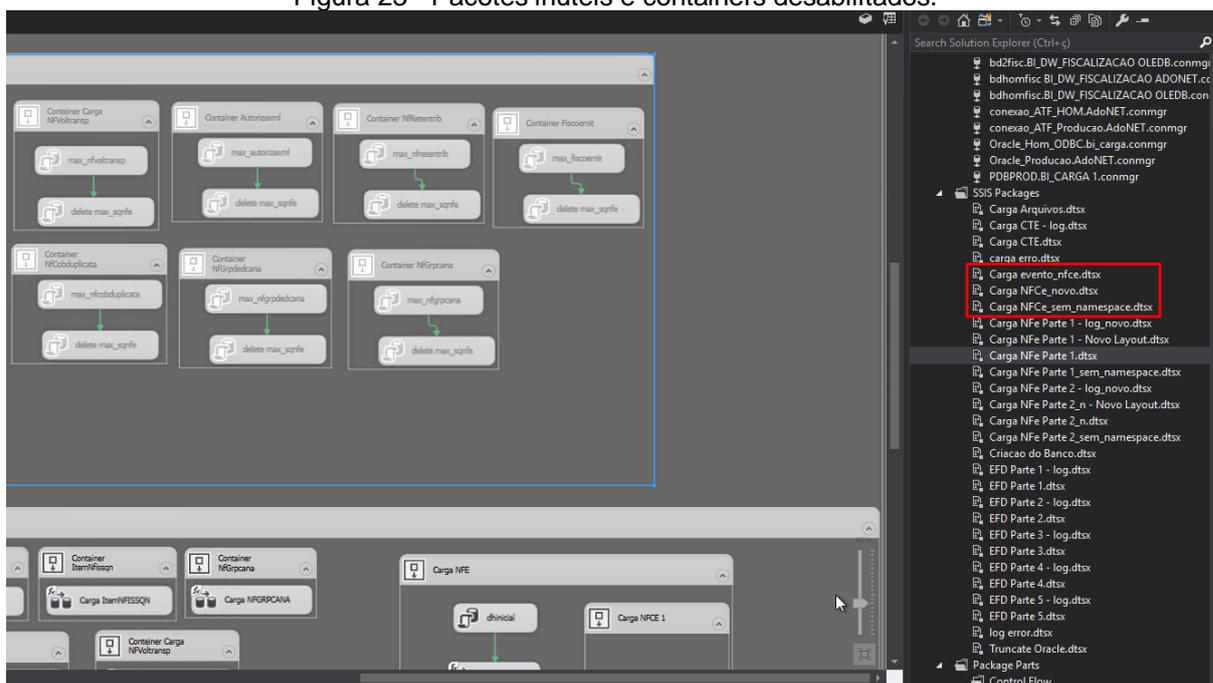
Nome_Tabela	Mínimo Notas	Máximo Notas
CTE_NFE	5301	5301
CTE_MUNIC	9138	9138
tbfis_logcarganfb	291482	291482
ITEM_NFEISSQN	65	65
NFE_NFCOBDUPLICATA	192343	192343
NFE_FISCOEMIT	7	7
NFE_RESPTEC	6751	6751
NFE_AUTORIZAXML	2128	2128
NFE	291482	291482
NFE_NFCOBFATURA	85401	85401
NFE_VOLTRANSP	286789	286789
NFE_OBSLIVRE	1146	1146
ITEM_NFE	1842866	1842866
EFD_k100	24	24
EFD_h020	8165	8165

Fonte: Elaborada pelo Autor, 2022.

5.4.3 Melhorias nos padrões do projeto

O projeto que nos foi passado estava totalmente defasado e sem nenhum padrão de projeto, continha vários pacotes que não são utilizados nas cargas, versões antigas de containers desabilitados, poluindo visualmente o projeto, laços de repetição desnecessários, entre outras coisas. A Figura 23 mostra um exemplo desses pacotes que existiam no projeto antigo e não era mais utilizados na carga e mostra os diversos containers desabilitados que poluíam visualmente o projeto, a figura foi capturada para o pacote “NFE Parte 1”.

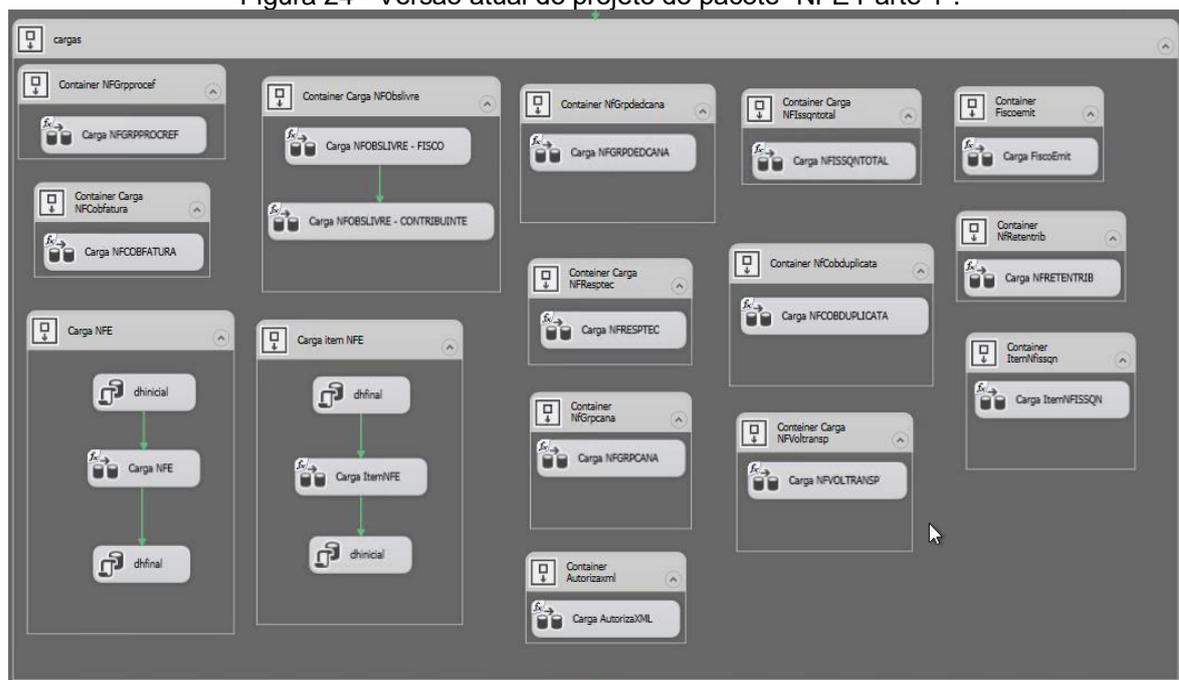
Figura 23 - Pacotes inúteis e containers desabilitados.



Fonte: Elaborada pelo Autor, 2022.

Foi feita uma limpeza no projeto removendo pacotes, containers, todo e qualquer tipo de estrutura que não era utilizado ou contribui de forma negativa para a realização das cargas, a Figura 24 mostra como está a versão atual do projeto para o pacote de carga “NFE Parte 1”.

Figura 24 - Versão atual do projeto do pacote “NFE Parte 1”.



Fonte: Elaborada pelo Autor, 2022.

5.4.4 Integração do novo servidor

Quando iniciamos as cargas do projeto piloto a partir de uma análise notamos que essas cargas iriam sobrecarregar o servidor BD2Fisc, pois no pacote inicial de 1.156 cargas que foram disponibilizadas para realizarmos. A grande maioria das cargas possuíam uma quantidade de notas inferior a 500 mil notas.

Nesse contexto, notamos que para cargas futuras seria necessária a criação de um novo servidor para rodar as cargas com quantidade de notas inferiores a 500 mil para dividir a carga de trabalho com o servidor BD2Fisc. Para suprir essa necessidade solicitamos ao setor de tecnologia da SEFAZ uma nova máquina idêntica ao BD2Fisc.

Com dois servidores para rodar cargas de perfis iguais fez-se necessário a realização de um estudo para separar as cargas que possuem quantidade de notas inferior a 500 mil nesses dois servidores, a Tabela 2 mostra como ficou a divisão para os 4 servidores, onde o novo servidor foi chamado de “BD22” e ficou responsável por realizar as cargas com quantidade de notas inferior a 300 mil notas.

Tabela 2 – Separação das cargas para os 4 servidores

Nome Servidor	Mínimo Notas	Máximo Notas
BD22	0	300.000
BD2	300.001	500.000
BD3	500.001	1.500.00
BD4	1.500.001	

Fonte: Elaborado pelo Autor, 2022.

5.5 Gerenciamento do consumo de disco

O ambiente onde os projetos são carregados possuem espaço limitado, portanto é de suma importância manter uma política de limpeza dos discos desses servidores, a SEFAZ não possui um administrador de banco de dados (DBA) para as bases SQL Server, portanto essa função também nos foi designada.

A princípio os servidores não possuíam uma política de limpeza das cargas e nem de gerenciamento dos logs gerados pelas mesmas, as cargas eram realizadas e enviadas para o BDFisc e ainda continuavam ocupando espaço no servidor. Desse modo desenvolvemos uma rotina de limpeza para cargas finalizadas. O script dessa rotina pode ser visto na Figura 25. Usamos um critério de limpeza de acordo com uma verificação que o script faz para ver se a carga finalizou sem problemas. Essa checagem é feita na variável chamada de “difCount”, se a validação deu certo e carga já foi para o BDFisc, o banco é apagado do servidor de produção.

Figura 25 - Script de limpeza das bases SQL Server.

```

:DECLARE @Lista_noBD TABLE(noBd varchar(100));
:DECLARE @noBD varchar(100);
:DECLARE @RowCnt INT;
:DECLARE @DROP nvarchar(100);
:DECLARE @RESULT nvarchar(200);

:INSERT INTO @Lista_noBD
:SELECT DISTINCT noBd
:FROM [BI_DW_FISCALIZACAO].[dbo].[controle_carga]
:WHERE difCount = 0 and dhFinal >= '2021-11-01';

:SET @RowCnt = (SELECT COUNT(*) AS RowCnt FROM @Lista_noBD);
:SET @DROP = 'DROP DATABASE IF EXISTS';

:WHILE @RowCnt <> 0
:BEGIN
:SET @noBD = (SELECT TOP 1 * FROM @Lista_noBD);
:SET @RESULT = (@DROP+' '+'['+@noBD+']');
:EXEC sp_executesql @RESULT;
:DELETE FROM @Lista_noBD WHERE noBd = @noBD;
:SET @RowCnt = (@RowCnt - 1);
:END

```

Fonte: Elaborada pelo Autor, 2022.

O servidor Oracle também sofre com o mesmo problema de espaço em disco, portanto essa atividade de limpeza dos dados também é realizada lá. Fizemos um estudo e comprovamos que a base suporta cerca de 11.5 milhões de registros nas tabelas, desse modo implementamos uma política de limpeza que ocorre nas quartas-feiras e sábados. Os dias foram escolhidos com base nas cargas de teste que realizamos e observamos que 2 dias de cargas seriam suficientes para ocupar próximo a lotação máxima suportada pelo Oracle sem prejudicar as cargas por estouro

de memória. A Figura 26 mostra o script que nos foi passado pela DBA do Oracle para realizar a limpeza das tabelas.

Figura 26 - Script de limpeza do Oracle

```
DECLARE  
  
var_retorno number:=null;  
  
begin  
    PRC_ESTRU_LOGCARG_TRUNC_TMP ('S', out_execucao=> var_retorno);  
    DBMS_OUTPUT.PUT_LINE(var_retorno);  
end;
```

Fonte: Elaborada pelo Autor, 2022.

T

5.6 Automação das cargas

A automação das cargas é uma parte muito importante do projeto, pois é através dela que as cargas acontecem com o mínimo de intervenção humana. Para realização dessa atividade foram utilizados os dois projetos do SSIS, o primeiro denominado de “Carga Temp” faz o monitoramento das ordens de serviço abertas no ATF e inclui os dados necessários para realização da carga das OS na tabela de controle “cargaTemp” no banco “BI_DW_FISCALIZACAO”, este banco é utilizado para controle das cargas.

A Tabela 3 mostra a estrutura da tabela “cargaTemp” e alguns campos como o número da ordem de serviço, o nome do banco, o período a ser fiscalizado e quantidade de notas, a variável “stbd” é quem controla o status da carga. Os possíveis valores para ela são: 1 quando foi identificado que uma OS foi aberta no ATF e os dados foram inseridos no SQL Server e a carga está pronta para ser

iniciada; 2 quando a carga das notas do ATF para o Oracle foi finalizada; 3 quando a carga do Oracle para o SQL Server foi iniciada, a carga permanece com o status 3 até que seja finalizada e a linha referente a OS seja apagada da tabela “cargaTemp”.

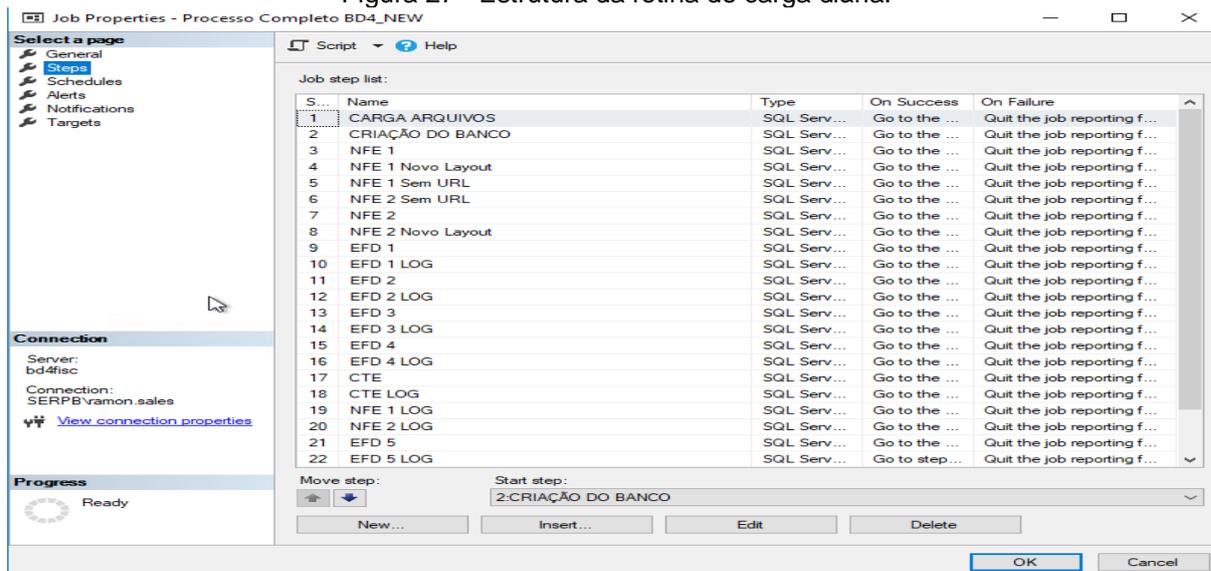
Tabela 3 – Estrutura da tabela “cargaTemp”.

numero_os	nome_banco	stbd	qtd_notas
193067	933000081200001929202186_20210423_093111	2	101
192197	933000081200001059202145_20210309_104635	2	4784
192007	933000081200000869202184_20210305_165842	2	38515
189933	933000081200006689202025_20201104_113757	2	4352653
192068	933000081200000930202193_20210305_165920	2	68926

Fonte: Elaborada pelo Autor, 2022.

Dentro do contexto citado anteriormente para resolver o problema da automação desenvolvemos quatro rotinas, dentre essas duas rodam diariamente que são as de verificação de ordens de serviços abertas no ATF o projeto “Carga Temp” e a de carga propriamente dita o “Processo Completo”. Essas rotinas apontam para os respectivos projetos do SSIS. A Figura 27 mostra a rotina implementada para execução das cargas e todos os passos que ela executa.

Figura 27 - Estrutura da rotina de carga diária.



Fonte: Elaborada pelo Autor, 2022.

As outras duas rotinas executados são referentes a limpeza e liberação de espaço nos servidores. A limpeza dos bancos de produção do SQL Server ocorre uma vez por semana no domingo, já a rotina de liberação de espaço do Oracle é executada nas quartas-feiras e sábados, os scripts que essas rotinas executam podem ser visualizados nas Figuras 25 e 26.

6 RESULTADO E DISCUSSÕES

Neste capítulo serão apresentados os resultados obtidos, a nova versão do BDFisc e o resultado de uma bateria de testes para validação das cargas, os dados que serão apresentados na seção 6.3 foram coletados através de uma bateria de cargas realizadas. Para realização desses testes foi criado um pacote com cerca de 1.156 ordens de serviços que foram abertas no ATF para que pudéssemos realizar as cargas que posteriormente foram validadas pelos fiscais da SEFAZ. Essa fase de testes do projeto foi batizada de “Projeto Piloto”.

6.1 BDFisc 2.0

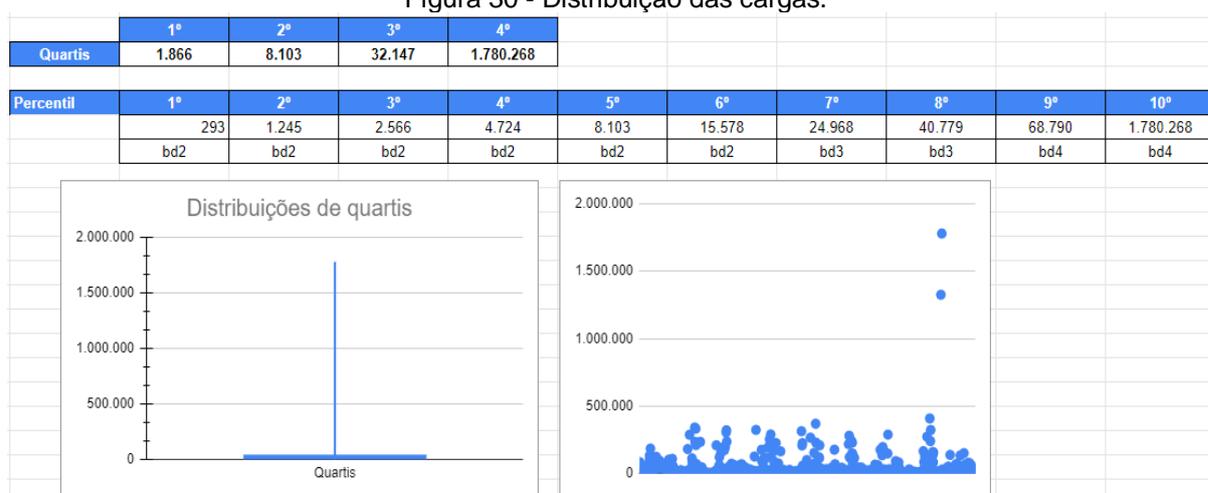
As diversas mudanças que foram realizadas no BDFisc e relatadas nas seções anteriores deram uma nova roupagem para o projeto trazendo diversos benefícios como uma melhora significativa no tempo de carga, uma validação confiável, remoção de coisas desnecessárias no projeto antigo, entre outras modificações significativas, essa nova versão do BDFisc reformulada foi apelidada de BDFisc 2.0. As Figuras 28 e 29 mostram um comparativo de como era o pacote de carga “NFE 1” no BDFisc e como ficou no BDFisc 2.0 no SSIS.

Para implementação dessa lógica foi adicionada uma nova variável na tabela “cargaTemp” chamada de “stbdcarga” essa variável é iniciada com o valor 1 no momento que a ordem de serviço chega no SQL Server e caso a carga não passe na validação o valor dela é mudado para 0 e a carga não vai para o BDFisc.

6.2 Peculiaridade das cargas

As ordens de serviços abertas no Projeto Piloto possuíam uma característica que fazia com que não fossem contempladas todas as bases de dados, pois cerca mais de 99% das cargas seriam direcionadas para o servidor BD2Fisc. De acordo com o critério de separação baseado na quantidade de notas, para driblar essa questão fizemos um estudo baseado na quantidade de notas que as 1.156 cargas possuíam e realizamos uma distribuição nos três servidores. A Figura 30 mostra como ficou a distribuição inicial nos três servidores. Posteriormente com intuito de agilizar as cargas, foram feitas novas distribuições. Para o BD2Fisc foram destinadas as cargas com notas entre 293 e 24.968, para o BD3Fisc as cargas com notas entre 24.969 e 40.779 e por fim o BD4Fisc ficou responsável pelas cargas com notas entre 40.780 e 1.780.268.

Figura 30 - Distribuição das cargas.



Fonte: Elaborada pelo Autor, 2022.

6.3 Resultados

Nesta seção serão apresentados métricas e resultados colhidos para cargas realizadas na antiga versão do BDFisc e as cargas do Projeto Piloto que foram executadas na versão BDFisc 2.0. Realizamos um comparativo com intuito de demonstrar a melhora obtida em relação ao tempo de carga na nova versão do projeto.

Posteriormente são apresentadas as métricas finais colhidas através das cargas do Projeto Piloto no servidor BDFisc. Os pontos que serão levados em consideração são: tempo médio de carga, quantidade de notas, entre outras informações.

6.3.1 *BDFisc 2.0 x BDFisc*

Levantamos duas amostras de cargas realizadas nas duas versões do projeto, na versão que nos foi passada assim que o projeto teve início e na nova versão obtida através das diversas modificações relatadas anteriormente.

O objetivo principal é demonstrar a melhora no tempo de execução das cargas de modo geral. Para realização do comparativo escolhemos 58 cargas que foram executadas nas duas versões do projeto. Devido a peculiaridade das cargas do Projeto, a amostra escolhida varia entre 50 mil e 100 mil notas.

A Tabela 4 mostra uma parte da tabela com os dados da amostra coletada das cargas executadas na versão BDFisc.

Tabela 4 – Amostra das cargas executadas na versão BDFisc.

sqparamcargafisc	periodo_inicial	periodo_final	duracao	qtd_notas
2606	2021-03-09 14:45:22.850	2021-03-09 16:56:46.240	02:11:24	63.946
2640	2021-04-02 18:43:53.410	2021-04-02 21:39:10.297	02:55:17	78.788
3400	2021-12-10 19:19:49.743	2021-12-10 19:44:59.480	00:25:10	61.902
4985	2021-12-21 06:13:47.587	2021-12-21 06:43:29.477	00:29:42	57.679
4986	2021-12-22 13:23:39.270	2021-12-22 13:45:13.987	00:21:34	72.092
4992	2021-12-22 15:36:49.690	2021-12-22 15:55:27.967	00:18:38	92.970
5011	2021-12-27 15:45:37.570	2021-12-27 16:01:13.337	00:15:36	72.620
5044	2021-12-28 10:14:37.737	2021-12-28 10:35:20.413	00:20:43	53.727
5047	2021-12-28 11:11:17.130	2021-12-28 11:27:39.473	00:16:22	79.158
5056	2021-12-28 22:55:42.450	2021-12-28 23:11:24.127	00:15:42	73.321
4995	2021-12-27 15:31:07.117	2021-12-27 15:44:30.027	00:13:23	53.383
5015	2021-12-29 04:10:11.460	2021-12-29 04:22:42.330	00:12:31	50.418
3094	2021-05-25 02:18:06.307	2021-05-25 03:49:26.807	01:31:20	65.999
3095	2021-05-25 05:00:05.130	2021-05-25 06:01:20.717	01:01:15	52.565

Fonte: Elaborada pelo Autor, 2022.

A Figura 31 resume os dados mensurados na Tabela 4, mostrando a média de notas da amostra e a duração média das cargas.

Figura 31 – Resumo das métricas do BDFisc



Fonte: Elaborada pelo Autor, 2022.

A Tabela 5 mostra uma parte da tabela com os dados da amostra coletada das cargas do Projeto Piloto que foram executadas na versão BDFisc 2.0.

Tabela 5 – Amostra das do Projeto Piloto executadas na versão BDFisc 2.0.

sqparamcargafisc	periodo_inicial	periodo_final	duracao	qtd_notas
5171	2022-06-18 00:34:20.323	2022-06-18 00:54:31.490	00:20:11	81334
5178	2022-06-18 22:28:13.477	2022-06-18 22:43:34.410	00:15:21	53799
5625	2022-06-25 15:23:48.983	2022-06-25 15:52:28.133	00:28:40	88726
6166	2022-06-25 15:36:47.913	2022-06-25 15:50:30.533	00:13:43	50073
5973	2022-06-25 08:17:44.007	2022-06-25 08:32:12.997	00:14:28	64079
5982	2022-06-25 11:19:09.540	2022-06-25 11:38:46.013	00:19:37	59423
6060	2022-06-25 12:50:44.567	2022-06-25 13:06:23.130	00:15:39	62502
6061	2022-06-25 13:11:02.523	2022-06-25 13:24:35.230	00:13:33	54448
6256	2022-06-25 18:43:21.800	2022-06-25 19:01:00.220	00:17:39	60042
6259	2022-06-25 19:36:19.520	2022-06-25 19:54:28.920	00:18:09	67970
6265	2022-06-25 21:24:55.333	2022-06-25 21:40:12.223	00:15:17	60606
5736	2022-06-25 21:35:56.967	2022-06-25 21:53:50.747	00:17:54	98068
6309	2022-06-26 04:33:08.770	2022-06-26 04:49:51.860	00:16:43	58705
6310	2022-06-26 04:55:17.980	2022-06-26 05:11:38.797	00:16:21	59360

Fonte: Elaborada pelo Autor, 2022.

A Figura 32 resume os dados mensurados na Tabela 5, mostrando a média de notas da amostra e a duração média das cargas.

Figura 31 – Resumo das métricas do BDFisc 2.0.



Fonte: Elaborada pelo Autor, 2022.

Observando as Figuras 31 e 32 podemos notar uma enorme diferença na duração média das cargas, esse resultado foi obtido devido às diversas mudanças implementadas no BDFisc 2.0, que fez com que o projeto não travasse as cargas igual a antiga versão.

6.3.2 BDFisc

Por fim nessa subseção serão apresentados os dados coletados para todas as 1.156 cargas do Projeto Piloto, como mencionado anteriormente o BDFisc é o servidor que serve como repositório central para armazenamento de todas as cargas e é lá

que os fiscais fazem as análises baseados nos dados carregados, dentre os dados coletados estão tempo médio de carga, a quantidade média de itens por carga, entre outras informações, a Tabela 6 mostra os números coletados para as cargas do Projeto Piloto.

Tabela 6 - Métricas sobre as cargas do BDFisc.

data_inicial	qtde_cargas	qtde_notas	qtde_itens	media_notas	media_itens	media_itens (nota)	media_tempo_carga (minutos)
18/06/2022	17	343021	1.719.214	20.177,71	101.130,24	5,0	7
19/06/2022	77	3777684	13.265.343	49.060,83	172.277,18	3,5	13
20/06/2022	128	1170521	10.106.560	9.144,70	78.957,50	8,6	6
21/06/2022	63	504808	2.225.570	8.012,83	35.326,51	4,4	6
23/06/2022	310	2645369	23.686.119	8.533,45	76.406,84	9,0	5
24/06/2022	270	6896707	57.480.420	25.543,36	212.890,44	8,3	10
25/06/2022	158	8706707	67.297.193	55.105,74	425.931,60	7,7	19
26/06/2022	63	7477723	59.314.448	118.694,02	941.499,17	7,9	38
27/06/2022	69	1593279	3.615.746	23.091,00	52.402,12	2,3	139
28/06/2022	1	1780268	2.040.273	1.780.268,00	2.040.273,00	1,1	10874

Fonte: Elaborada pelo Autor, 2022.

Os cartões mostrados na Figura 32, resumem os dados coletados e que foram apresentados na Tabela 6, além disso apresentam dados referentes ao tempo de execução das cargas.

Figura 32 – Resumo das métricas do BDFisc.



Fonte: Elaborada pelo Autor, 2022.

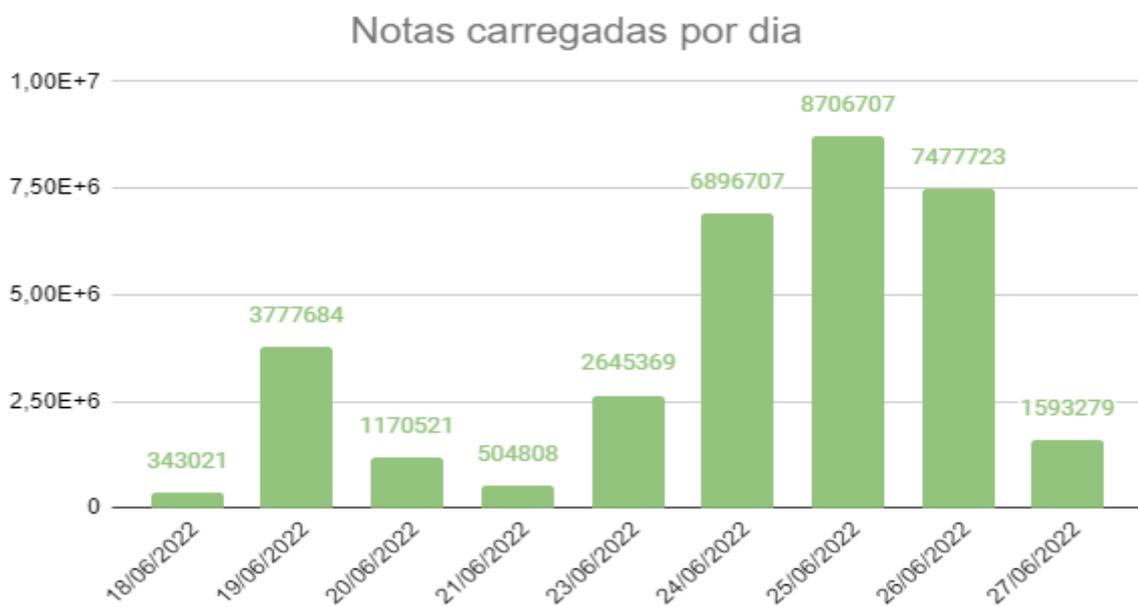
A seguir são apresentados os gráficos que demonstram a quantidade de cargas, notas e itens carregados por dia respectivamente.

Gráfico 1 – Cargas efetuadas por dia.



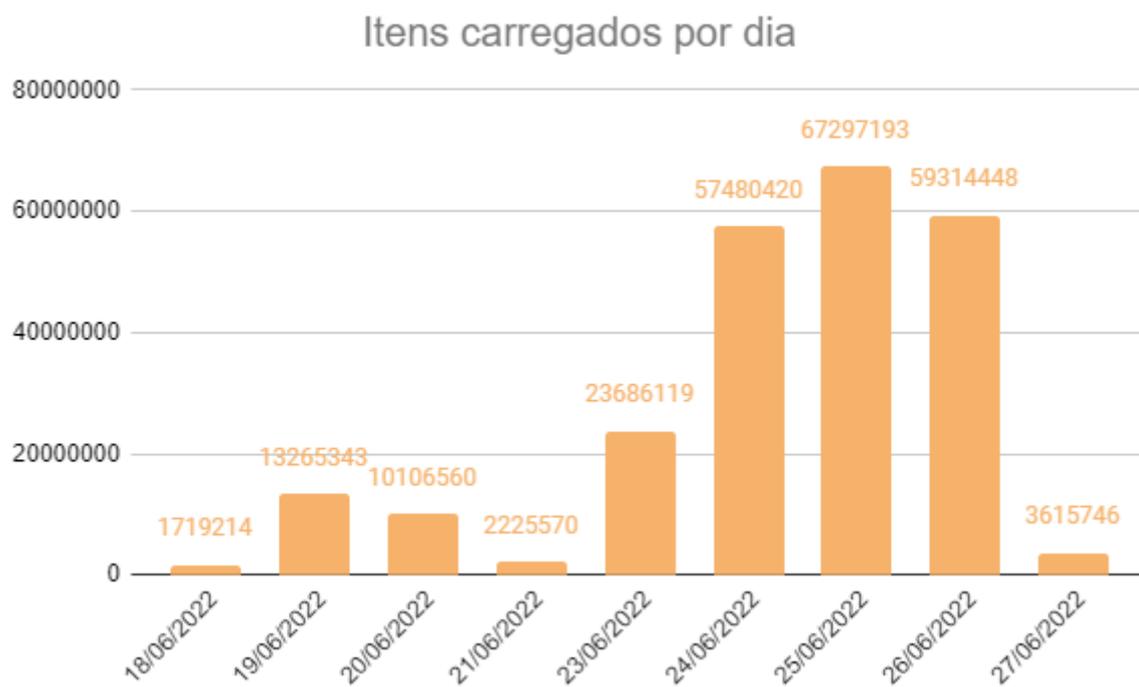
Fonte: Elaborada pelo Autor, 2022.

Gráfico 2 - Notas carregadas por dia.



Fonte: Elaborada pelo Autor, 2022.

Gráfico 3 - Itens carregados por dia.



Fonte: Elaborada pelo Autor, 2022.

7 CONCLUSÃO

Esse trabalho apresentou de forma geral todas as atividades que foram executadas para realização da implantação, refatoração e manutenção do sistema de análise fiscal para Secretária da Fazenda do Estado da Paraíba. A solução desenvolvida tem um papel fundamental no processo de análise fiscal e para que o trabalho fosse o mais fiel possível no atendimento das necessidades dos fiscais, trabalhamos em conjunto com o pessoal da fiscalização da SEFAZ e com a DBA do Oracle que nos auxiliou com as consultas para extração dos dados tanto no Oracle como também no Informix.

7.1 Trabalhos Futuros

Abaixo são listadas algumas atividades que podem ser exercidas no futuro com intuito de melhorar ainda mais o projeto.

- Versionamento do projeto no Git;
- Documentação detalhada de todo o processo de carga que o projeto executa;
- Estudo sobre a possibilidade de execução das cargas de modo paralelo para agilizar ainda mais o processo;
- Migração do projeto para outra plataforma que seja mais performática que a da Microsoft.

REFERÊNCIAS

AWS. **O que é um data lake?**. Disponível em: <https://aws.amazon.com/pt/big-data/datalakes-and-analytics/what-is-a-data-lake/>. Acesso em 30 jul. 2022.

CARDOSO G, CARDOSO V. **Linguagem SQL fundamentos e práticas**. São Paulo: Editora Saraiva, 2013. Disponível em: <https://books.google.com.br/books?hl=pt-BR&lr=&id=aYVnDwAAQBAJ&oi=fnd&pg=PT8&dq=linguagem+sql&ots=kC8Jk9hiPB&sig=NQQL9MrUKZmJoHJycbBbu8Bp-dU#v=onepage&q&f=false>. Acesso em: 08 jun. 2022.

CDES. **Indicadores de Equidade do Sistema Tributário Nacional**. 2009. Disponível em: https://www.ipea.gov.br/portal/images/stories/0906_Indicadores_de_Equidade_Sistema_TN_Relatorio_Observacao_01.pdf. Acesso em: 02 jul. 2022.

CDL (João Pessoa). **Índice de Inadimplência**. 2022. Disponível em: <https://cdljp.com.br/indice-de-inadimplencia>. Acesso em: 28 jun. 2022.

DATE, C. J. **Introdução a Sistemas de Banco de Dados**. 8.ed. Rio de Janeiro: Elsevier Editora Ltda, 2004.

DB-Engines Ranking. **DB-Engines**, 2022. Disponível em: <https://db-engines.com/en/ranking>. Acesso em: 11 jul 2022.

ENOTAS. **O que é e para que serve a SEFAZ?**. Disponível em: <https://enotas.com.br/blog/sefaz/>. Acesso em 28 jun. 2022.

IBM. **ETL (Extract, Transform, Load)**. Disponível em: <https://www.ibm.com/cloud/learn/etl>. Acesso em 31 jun. 2022.

IBM. **Data Mart**. Disponível em: <https://www.ibm.com/cloud/learn/data-mart>. Acesso em 31 jun. 2022.

INMON, WILLIAM. **Building the Data Warehouse**. 4.ed. Indianapolis: Wiley Publishing, Inc, 2005.

MICROSOFT. **O que é a integração de dados?**. Disponível em: <https://azure.microsoft.com/pt-br/overview/what-is-data-integration/>. Acesso em 10 jul. 2022.

MICROSOFT. **SQL Server Integration Services**. Disponível em: <https://docs.microsoft.com/pt-br/sql/integration-services/sql-server-integration-services?view=sql-server-ver16>. Acesso em 31 jul. 2022.

MICROSOFT. **O que é o SSMS (SQL Server Management Studio)?**. Disponível em: <https://docs.microsoft.com/pt-br/sql/ssms/sql-server-management-studio-ssms?view=sql-server-ver16>. Acesso em 31 jul. 2022.

MICROSOFT. **O que é o Analysis Services?**. Disponível em: <https://docs.microsoft.com/pt-br/analysis-services/analysis-services-overview?view=asallproducts-allversions>. Acesso em 31 jul. 2022.

MICROSOFT. **O que é o SQL Server Reporting Services (SSRS)?**. Disponível em: <https://docs.microsoft.com/pt-br/sql/reporting-services/create-deploy-and-manage-mobile-and-paginated-reports?view=sql-server-ver16>. Acesso em 31 jul. 2022.

NFE.IO. **O que é SEFAZ, como funciona e o que significa para empresas**. Disponível em: <https://nfe.io/blog/nota-fiscal/o-que-e-sefaz/#:~:text=O%20que%20C3%A9%20SEFAZ%2C%20como%20funciona%20e%20o%20que%20significa%20para%20empresas&text=SEFAZ%2C%20abreviatura%20para%20Secretaria%20de,que%20vendem%20produtos%20e%20servi%C3%A7os>. Acesso em 29 jul. 2022.

NIDA, FÁTIMA. **O que é um data warehouse - definição, exemplo e benefícios**. Disponível em: <https://www.astera.com/pt/tipo/blog/defini%C3%A7%C3%A3o-de-data-warehouse/>. 2022. Acesso em: 05 jul. 2022.

ORDILLE, J.; RAJARAMAN, A.; HALEVY A. **Data Integration: The Teenage Years**. Disponível em: https://www.cin.ufpe.br/~if696/referencias/integracao/_Data_Integration-The_Teenage_Years.pdf. Acesso em 30 jul. 2022.

PARAÍBA. **PORTARIA Nº 00061/2017/GSER**. Disponível em: <https://www.sefaz.pb.gov.br/legislacao/211-portarias/portarias-2017/4075-portaria-n-00061-2017-gser>. Acesso em 30 jul. 2022.

SOMMERVILLE, Ian. **Engenharia de Software**. 9. ed. São Paulo: Pearson Education do Brasil Ltda, 2011.