

**Paulo Henrique Oliveira**

**Automação de Extração e Análise de Indicadores de  
Sustentabilidade Fiscal do Brasil**

**Brasília**

**2020**

**Paulo Henrique Oliveira**

**Automação de Extração e Análise de Indicadores de  
Sustentabilidade Fiscal do Brasil**

Trabalho de conclusão do curso de pós-graduação lato sensu em Análise de Dados para o Controle realizado pela Escola Superior do Tribunal de Contas da União como requisito para a obtenção do título de especialista.

Orientador: Glauco Garcia Scandaroli, D. Sc.

**Brasília**

**2020**

## REFERÊNCIA BIBLIOGRÁFICA

Oliveira, Paulo Henrique. Automação de Extração e Análise de Indicadores de Sustentabilidade Fiscal do Brasil. Trabalho de Conclusão de Curso (Especialização em 2019) – Escola Superior do Tribunal de Contas da União, Instituto Serzedello Corrêa, Brasília DF.

## CESSÃO DE DIREITOS

NOME DO AUTOR: Paulo Henrique Oliveira

TÍTULO: Automação de Extração e Análise de Indicadores de Sustentabilidade Fiscal do Brasil

GRAU/ANO: Especialista/2020

É concedido ao Instituto Serzedello Corrêa (ISC) permissão para reproduzir cópias deste Trabalho de Conclusão de Curso e emprestar ou vender tais cópias somente para propósitos acadêmicos e científicos. Do mesmo modo, o ISC tem permissão para divulgar este documento em biblioteca virtual, em formato que permita o acesso via redes de comunicação e a reprodução de cópias, desde que protegida a integridade do conteúdo dessas cópias e proibido o acesso a partes isoladas desse conteúdo. O autor reserva outros direitos de publicação e nenhuma parte deste documento pode ser reproduzida sem a autorização por escrito do autor.

---

Paulo Henrique Oliveira

henriqueo@tcu.gov.br

Ficha catalográfica

A ficha de identificação é elaborada pelo próprio autor.

Orientações em:

<https://portal.tcu.gov.br/biblioteca-ministro-ruben-rosa/servicos/normalizacao-de-publicacoes.htm>

**Paulo Henrique Oliveira**

# **Automação de Extração e Análise de Indicadores de Sustentabilidade Fiscal do Brasil**

Trabalho de conclusão do curso de pós-graduação lato sensu em Análise de Dados para o Controle realizado pela Escola Superior do Tribunal de Contas da União como requisito para a obtenção do título de especialista.

Brasília, 31 de março de 2020.

## **Banca Examinadora:**

---

Glauco Garcia Scandaroli, D. Sc.  
Orientador

---

Prof. Joaquim Ramalho de Albuquerque, Me.

## RESUMO

O Tribunal de Contas da União, no desempenho de suas competências instituídas na Constituição Federal de 1988 e nas demais normas, realiza fiscalizações acerca da responsabilidade na gestão fiscal, entre as quais, destaca-se o acompanhamento quadrimestral do Relatório de Gestão Fiscal (RGF). Uma das extensas tarefas da equipe de fiscalização diz respeito às extrações de dados do sistema TesouroGerencial, que contém informações do Sistema Integrado de Administração Financeira (Siafi). A complexidade de tais extrações ocupa parte relevante da força de trabalho, devido ao tamanho das bases de dados pesquisadas. As consultas geram dados brutos, os quais precisam sofrer tratamento e organização, etapas que ocorrem em arquivos de planilhas eletrônicas em formato proprietário. A presente solução busca automatizar as etapas de extração, tratamento, organização e apresentação dos demonstrativos da Dívida Consolidada Líquida, das Garantias e Contragarantias de Valores e das Operações de Crédito. Por meio do desenvolvimento de códigos na linguagem de programação python, ocorre o acesso à ferramenta LabContas do TCU, que possui bases de dados internalizadas, entre as quais o DWTG, referente ao TesouroGerencial. Após estudo técnico sobre os normativos, conteúdo e formas dos demonstrativos, de acordo com a metodologia contida no arcabouço jurídico orçamentário, as informações do Relatório de Gestão Fiscal foram mapeadas e as consultas necessárias convertidas para que pudessem ser realizadas em Linguagem de Consulta Estruturada (SQL). Houve então a modelagem da estrutura de tratamento dos dados brutos obtidos pelas consultas, considerando que cada dado é utilizado como insumo para compor uma ou mais linhas de um demonstrativo. A execução dos códigos desenvolvidos na presente solução gera uma minuta de relatório assistido, com a tabela de cada demonstrativo preenchida de acordo com o formato utilizado pelo TCU nos relatórios de fiscalização do RGF, contemplando ainda a análise de cumprimento aos limites definidos na LRF e variações entre períodos. As principais conclusões da análise são geradas de forma automática em tabela e em textos, na forma de parágrafos pré-elaborados com os resultados das análises.

**Palavras-chave:** Controle externo. Gestão Fiscal. Mineração de dados. Tribunal de Contas da União. Limites de Despesas Públicas.

## ABSTRACT

The Federal Court of Accounts (TCU) audits Brazil's fiscal management and promotes accountability, pursuant to the country's 1988 Constitution and infra-constitutional legislation. The "Financial Management Report" (RGF) stands out as an eminent document on the government's accounts mandated by law. One of the cumbersome tasks regularly executed by TCU auditors concerns data extraction from the "TesouroGerencial" system, which contains information obtained from the "Integrated Financial Administration System" (Siafi). The complexity of these tasks requires attention from a significant part of the workforce, due to the size of the queried databases, as well as raw data transformation, which is currently performed using spreadsheet files. The present solution seeks to automate the stages of extraction, transformation, organization and presentation of the RGF's statements of Consolidated Net Debt, Securities and Counter-guarantees for Securities and Credit Operations. Through the development of coding in the Python programming language, the TCU's "LabContas" tool is accessed – this enables the direction of queries to databases internalized in LabContas, including the DWTG database, which refers to Treasury Management. After a technical study of the standards, content and forms of the statements, in accordance with the methodology contained in the budgetary legal framework, the information in the RGF has been mapped and the necessary consultations were converted to be carried out in Structured Query Language (SQL). Subsequently, the structure to treat the raw data obtained by the queries has been modeled, considering that each data constitutes input to compose one or more lines of a statement. The execution of the codes developed in this solution generates an assisted report draft, with the table of each statement filled in accordance with the format used by the TCU in RGF inspection reports, thereby carrying out the analysis of compliance with the limits defined in the Fiscal Responsibility Law (LRF), and of the evolution of results over time. The main conclusions of the analysis are generated automatically in a table and texts, in the form of standardized paragraphs.

**Keywords:** External control. Tax Management. Data mining. Federal Court of Accounts. Public Expenditure Limits.

## LISTA DE ILUSTRAÇÕES

Figura 1 – Fases do CRISP-DM.....	15
Figura 2 – Demonstrativo da Dívida Consolidada Líquida.....	22
Figura 3 – Exemplo de Metodologia de Apuração de Item da DCL.....	23
Figura 4 – Demonstrativo das Garantias e Contragarantias de Valores.....	25
Figura 5 – Exemplo de Metodologia de item do Demonstrativo das Garantias e Contragarantias de Valores.....	25
Figura 6 – Demonstrativo das Operações de Crédito.....	28
Figura 7 – Exemplo de Metodologia de item do Demonstrativo das Operações de Crédito.....	28
Figura 8 – Classes de Informação do Plano de Contas Aplicado ao Setor Público ...	30
Figura 9 – Código das Contas Contábeis do Plano de Contas Aplicado ao Setor Público.....	30
Figura 10 – Classes e Grupos do Plano de Contas Aplicado ao Setor Público.....	31
Figura 11 – Conceito de Conta Corrente utilizado pelo Sistema TesouroGerencial..	32
Figura 12 – Tipos de Administração constantes no Siafi.....	32
Figura 13 – Extrato do Template Base para o Demonstrativo da Dívida Consolidada Líquida.....	36
Figura 14 – Extrato do Template Base para o Demonstrativo das Garantias e Contragarantias de Valores.....	37
Figura 15 – Extrato do Template Base para o Demonstrativo das Operações de Crédito.....	38
Figura 16 – Exemplo de Conclusão e Encaminhamento da Minuta de Relatório Assistido.....	39
Figura 17 – Extrato de Minuta de Relatório Assistido do Demonstrativo da DCL ...	40
Figura 18 – Extrato de Minuta de Relatório Assistido do Demonstrativo das Garantias e Contragarantias de Valores.....	41
Figura 19 – Extrato de Minuta de Relatório Assistido do Demonstrativo das Operações de Crédito.....	42

## **LISTA DE ABREVIATURAS E SIGLAS**

CF/1988 – Constituição Federal de 1988

DCL – Dívida Consolidada Líquida

DWTG – Data Warehouse do Tesouro Gerencial

GND – Grupo de Natureza de Despesa

LC – Lei Complementar

LRF – Lei de Responsabilidade Fiscal

MCASP – Manual de Contabilidade Aplicada ao Setor Público

MDF – Manual de Demonstrativos fiscais

PCASP – Plano de Contas Aplicado ao Setor Público

RCL – Receita Consolidada Líquida

RGF – Relatório de Gestão Fiscal

Segecex – Secretaria-Geral de Controle Externo

Semag – Secretaria de Macroavaliação Governamental

Siafi – Sistema Integrado de Administração Financeira

Siconfi – Sistema de Informações Contábeis e Fiscais do Setor Público Brasileiro

Siop – Sistema Integrado de Planejamento e Orçamento

SQL – Structured Query Language

STN – Secretaria do Tesouro Nacional

TCU – Tribunal de Contas da União



## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO .....</b>	<b>11</b>
1.1	Objetivos .....	13
<b>2</b>	<b>METODOLOGIA .....</b>	<b>14</b>
<b>3</b>	<b>ENTENDIMENTO DO NEGÓCIO .....</b>	<b>15</b>
3.1	Determinando os objetivos do negócio .....	15
3.2	Verificando a situação atual .....	16
3.3	Objetivos de mineração de dados .....	18
3.4	Plano do projeto.....	18
<b>4</b>	<b>ENTENDIMENTO DOS DADOS .....</b>	<b>19</b>
4.1	Histórico .....	19
4.2	Coletando dados iniciais.....	20
4.2.1	Demonstrativo da Dívida Consolidada Líquida .....	20
4.2.2	Demonstrativo das Garantias e Contragarantias de valores .....	23
4.2.3	Demonstrativo das Operações de Crédito .....	26
4.3	Descrevendo os dados .....	29
<b>5</b>	<b>PREPARAÇÃO DOS DADOS.....</b>	<b>34</b>
5.1	Selecionando dados .....	34
5.2	Tratando dados .....	34
5.3	Construindo novos dados .....	35
5.4	Integrando dados .....	35
<b>6</b>	<b>MODELAGEM .....</b>	<b>35</b>
<b>7</b>	<b>AVALIAÇÃO .....</b>	<b>39</b>
<b>8</b>	<b>CONCLUSÃO .....</b>	<b>43</b>
8.1	Trabalhos Futuros.....	44
	<b>REFERÊNCIAS .....</b>	<b>46</b>

## 1 INTRODUÇÃO

O controle externo no Brasil é de titularidade do Congresso Nacional, que o exerce com auxílio do Tribunal de Contas da União (TCU), ao qual compete, nos termos do art. 71:

I - apreciar as contas prestadas anualmente pelo Presidente da República, mediante parecer prévio que deverá ser elaborado em sessenta dias a contar de seu recebimento;

II - julgar as contas dos administradores e demais responsáveis por dinheiros, bens e valores públicos da administração direta e indireta, incluídas as fundações e sociedades instituídas e mantidas pelo Poder Público federal, e as contas daqueles que derem causa a perda, extravio ou outra irregularidade de que resulte prejuízo ao erário público;

(..)

VIII - aplicar aos responsáveis, em caso de ilegalidade de despesa ou irregularidade de contas, as sanções previstas em lei, que estabelecerá, entre outras cominações, multa proporcional ao dano causado ao erário;

A Lei Complementar (LC) 101/2000 – Lei de Responsabilidade Fiscal (LRF) estabelece “normas de finanças públicas voltadas para a responsabilidade na gestão fiscal”, conforme art. 1º, *caput*.

O § 1º do art. 1º define contornos para a responsabilidade na gestão fiscal, a qual pressupõe a ação planejada e transparente, com vistas à prevenção de riscos e à correção de desvios que podem afetar o equilíbrio das contas públicas. Citam-se os instrumentos cumprimento de metas, obediência a limites e condições para a renúncia de receita, geração de despesas com pessoal, da seguridade social, dívida consolidada e mobiliária, operações de crédito, concessão de garantias e restos a pagar.

Os instrumentos de transparência da gestão fiscal estão previstos, de forma não exaustiva, no art. 48 da LRF, sendo um deles, o Relatório de Gestão Fiscal (RGF).

O art. 54 da LRF trata da periodicidade (quadrimestral) e da responsabilidade por assinar o RGF, a qual incide sobre os titulares dos Poderes e órgãos constantes do art. 20 dessa lei.

Já o art. 55 apresenta a composição do RGF, que contém demonstrativos sobre: despesa com pessoal, dívida consolidada e mobiliária, concessão de garantias, operações de crédito. No último quadrimestre de cada exercício, são elaborados ainda os demonstrativos do montante da disponibilidade de caixa em 31 de dezembro e da inscrição em restos a pagar.

A Lei 8.443/1992 – Lei Orgânica do TCU (LOTUCU), no art. 1º, detalha as competências do TCU, entre as quais, destacam-se: apreciar as contas prestadas pelo Presidente da Re-

pública; acompanhar a arrecadação de receitas federais, realizar fiscalização contábil, financeira, orçamentária, operacional e patrimonial das unidades dos poderes da União e das demais que lidem com dinheiros, bens e valores públicos federais.

Conforme o Regimento Interno do TCU (aprovado pela Resolução TCU 155/2002 e alterações), o relatório que acompanha o parecer prévio sobre as contas prestadas pelo Presidente da República deve conter informações sobre o cumprimento dos limites e parâmetros estabelecidos pela LRF.

A Resolução-TCU 142/2001 dispõe sobre as competências atribuídas ao TCU pela LC 101/2000. O art. 1 discorre que compete ao TCU:

- I - verificar o cálculo dos limites da despesa total com pessoal na esfera federal, compreendendo:
  - (...)
  - II - alertar aos Poderes ou órgãos referidos no inciso I, quando constatar:
    - (...)
    - b) que o montante da despesa total com pessoal ultrapassou noventa por cento do limite;
    - c) que os montantes das dívidas consolidada e mobiliária, das operações de crédito e da concessão de garantia encontram-se acima de noventa por cento dos respectivos limites;

A utilização do instrumento do alerta tem origem no § 1º do art. 59 da LRF.

Art. 59 (...)

§ 1º Os Tribunais de Contas alertarão os Poderes ou órgãos referidos no art. 20 quando constatarem:

- I - a possibilidade de ocorrência das situações previstas no inciso II do art. 4º e no art. 9º;
- II - que o montante da despesa total com pessoal ultrapassou 90% (noventa por cento) do limite;
- III - que os montantes das dívidas consolidada e mobiliária, das operações de crédito e da concessão de garantia se encontram acima de 90% (noventa por cento) dos respectivos limites;
- IV - que os gastos com inativos e pensionistas se encontram acima do limite definido em lei;
- V - fatos que comprometam os custos ou os resultados dos programas ou índices de irregularidades na gestão orçamentária.

No âmbito do TCU, cabe à Secretaria-Geral de Controle Externo (Segecex) gerenciar as atividades de controle externo. A Secretaria de Macroavaliação Governamental (Semag), unidade integrante da Segecex, é responsável por realizar fiscalização e controle do cumprimento das normas estabelecidas pela LRF, nos termos da Resolução-TCU 305, de 28/12/2018, art. 48, inciso XI. À Semag compete ainda sistematizar as ações de controle e realizar fiscalização da responsabilidade fiscal (inciso XII).

Dessa forma, no desempenho de sua missão constitucional, o TCU, por meio da Semag, deve realizar trabalhos para assegurar a obediência aos limites e condições estabelecidas na LRF, notadamente no que se refere à observância da responsabilidade na gestão fiscal.

Para tanto, o TCU desenvolve fiscalização periódica e contínua sobre o RGF publicado pelo Poder Executivo federal, tarefa essencial no acompanhamento da sustentabilidade fiscal da União.

A automatização desenvolvida neste trabalho tornará as análises do TCU mais eficientes, notadamente devido à otimização de parte relevante das atividades desenvolvidas pela equipe de fiscalização, liberando força de trabalho especializada para que o Tribunal aprofunde suas ações de controle em temas relacionados à dívida pública que demandem efetivamente a atuação dos auditores.

### **1.1 Objetivos**

A presente solução visa automatizar procedimentos relacionados à extração, tratamento, análise e apresentação de informações relativas aos Relatórios de Gestão Fiscal do Poder Executivo federal – com ênfase nos demonstrativos da Dívida Consolidada Líquida, das Garantias e Contragarantias de Valores e das Operações de Crédito.

O RGF, publicado quadrimestralmente pelo Poder Executivo federal, é analisado por uma equipe de auditores federais de controle externo (AUFCs) do TCU, que verifica se os limites e condições estabelecidos na legislação foram observados e elabora um relatório de acompanhamento, o qual subsidia o acórdão a ser emitido pelo Plenário da Corte de Contas.

Para realização do trabalho, a equipe de fiscalização realiza consultas no sistema TesouroGerencial, o qual armazena informações do Sistema Integrado de Administração Financeira (Siafi), principal instrumento de registro, acompanhamento e controle da execução orçamentária, financeira e patrimonial do Governo Federal.

No âmbito do plano estratégico do TCU válido para 2019-2025, um dos objetivos estratégicos priorizados é “Atuar pela sustentabilidade fiscal da União”.

A esse objetivo estão associadas, no plano operacional da Semag (aprovado para o período de abril de 2019 a março de 2021), as ações de acompanhamento dos RGFs e a de automatização das consultas e análises do RGF.

Para o exercício de 2020, constam as fiscalizações de acompanhamento do RGF para cada um dos três bimestres. A força de trabalho destinada à realização dessas ações no TCU é

de 190 HDs (homens-dia alocados em instrução de processos) por ano, de acordo com o planejamento da Semag homologado pelo TCU.

Acerca da automatização das consultas e análises do RGF, registra-se que há uma equipe da Semag desenvolvendo parte da tarefa, com foco nos demonstrativos da despesa com pessoal e da disponibilidade de caixa e restos a pagar.

Por meio da solução desenvolvida neste trabalho, pretende-se automatizar tarefas que consomem parte relevante do trabalho realizado pela equipe de fiscalização.

As consultas necessárias serão realizadas no sistema TesouroGerencial de forma automática, os dados serão tratados e organizados para alimentarem as tabelas e os parágrafos das minutas de instrução assistida de cada demonstrativo.

Além disso, serão verificados se os limites foram cumpridos, se os resultados ficaram entre 90 e 100% dos limites (o que enseja a emissão de alerta, nos termos da LRF), ou se foram ultrapassados, com a aposição de parágrafo de conclusão contemplando cada uma das hipóteses e os consequentes encaminhamentos estabelecidos na Lei de Responsabilidade Fiscal.

O detalhamento das etapas realizadas consta dos itens Modelagem e Avaliação do presente relatório.

Pertencem ao escopo deste trabalho a automatização e tarefas referentes aos demonstrativos da dívida consolidada líquida (DCL), das garantias e contragarantias de valores, e das operações de crédito.

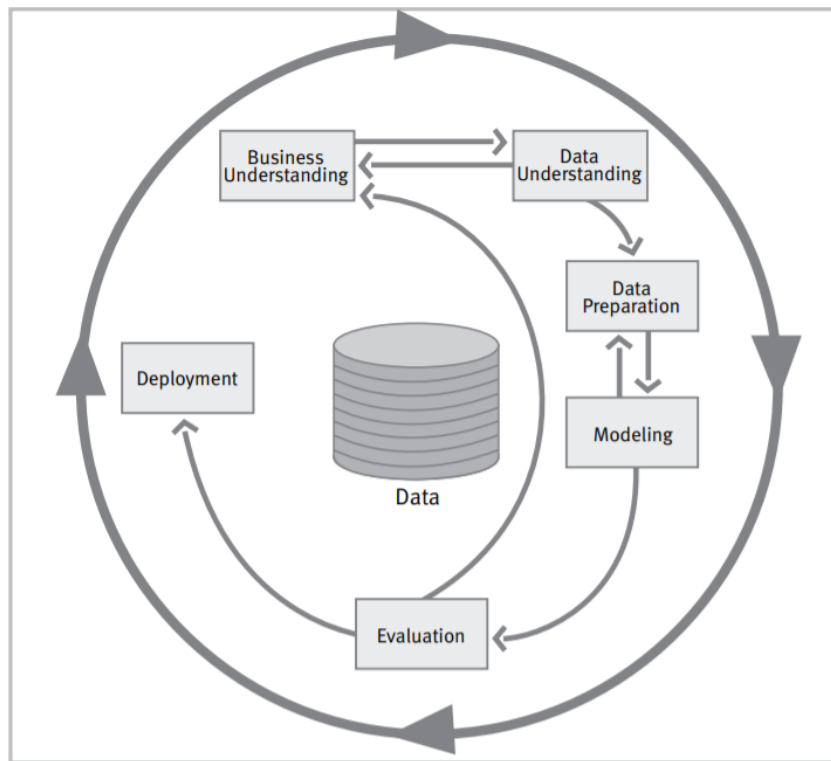
## 2 METODOLOGIA

Para desenvolver o presente trabalho, será utilizado a metodologia CRISP-DM (*Cross Industry Standard Process for Data Mining*).

O CRISP-DM fornece uma visão geral do projeto de mineração de dados, com as fases, tarefas a serem desenvolvidas. A metodologia consiste em seis etapas: entendimento do negócio; entendimento dos dados; preparação dos dados; modelagem; avaliação dos resultados; implantação.

A **Erro! Fonte de referência não encontrada.** apresenta as fases estabelecidas pela metodologia CRISP-DM e suas dependências.

Figura 1 – Fases do CRISP-DM



Fonte: CRISP-DM 1.0 Step-by-step data mining guide.

A seqüência de frases não é rígida, e as setas indicam apenas as dependências entre fases. O círculo externo mostra o caráter cíclico da mineração de dados, que não se encerra após a implementação da solução.

### 3 ENTENDIMENTO DO NEGÓCIO

Serão apresentados nesta seção os objetivos do projeto, explorando o problema, a situação atual e as ferramentas disponíveis, os objetivos e os critérios de sucesso da mineração de dados e o plano do projeto.

#### 3.1 Determinando os objetivos do negócio

O Poder Executivo federal publica o Relatório de Gestão Fiscal até trinta dias após o encerramento do respectivo quadrimestre, inclusive por meio eletrônico, nos termos do art. 55 da Lei de Responsabilidade Fiscal.

Atualmente, além da publicação no Diário Oficial da União, o RGF é disponibilizado no Sistema de Informações Contábeis e Fiscais do Setor Público Brasileiro (Siconfi).

O trabalho realizado pelo TCU inclui verificar se os valores publicados estão de acordo com a metodologia descrita no Manual de Demonstrativos Fiscais (MDF), publicado pela Secretaria do Tesouro Nacional (STN), em consonância com o disposto na LRF.

Para realizar a citada verificação, é necessário que a equipe de fiscalização da Semag realize extrações de dados por meio de consultas ao sistema TesouroGerencial. Cada extração gera uma informação que é insumo para alimentação dos campos de cada anexo do RGF, de modo que, para verificar a exatidão dos números publicados, a Semag realiza diversas consultas, além de tratamento e organização dos dados, etapas que consomem parte relevante da força de trabalho dos auditores.

O objetivo deste trabalho é a automatização das etapas de extração, tratamento, análise e apresentação das informações constantes nos demonstrativos da Dívida Consolidada Líquida (DCL), das Garantias e Contragarantias de Valores e das Operações de Crédito.

Uma vez implementada a solução ora desenvolvida, estima-se um período de transição, a fim de atestar que a ferramenta funciona de forma segura e perene para os períodos seguintes. Nessa fase, o método anterior segue sendo realizado, de forma a não haver risco de descontinuidade ou atraso na conclusão dos trabalhos decorrentes de eventuais inconsistências.

Encerrada a fase de validação da solução, espera-se que haja redução da força de trabalho necessária para realização da tarefa no TCU, de modo que os auditores possam se concentrar em atividades que demandem efetiva análise do auditor.

### **3.2 Verificando a situação atual**

Os anexos do Relatório de Gestão Fiscal apresentam informações que sintetizam, de forma padronizada, transações contábeis, as quais são registradas no Sistema Integrado de Administração Financeira do Governo Federal (Siafi). A organização das informações é definida pela metodologia constante do MDF, manual publicado pela STN.

Em que pese a maturidade da metodologia e dos dados constantes do Siafi, não se pode presumir que os números estão isentos de erros, fazendo-se necessária a atuação do TCU no sentido de assegurar a confiabilidade dos dados apresentados, em obediência aos normativos que disciplinam as competências do Tribunal quanto à responsabilidade na gestão fiscal.

Atualmente, a Semag realiza consultas diversas ao TesouroGerencial, as quais resultam em informações que são exportadas para planilha eletrônica (formato *xlsx*). Em seguida, desenvolvem-se etapas de cruzamento e manipulação de dados, até que as informações estejam em condições de serem confrontadas com os demonstrativos que foram publicados.

Devido ao grande volume de informações, as consultas realizadas no sistema TesouroGerencial podem ser demoradas. Além disso, as extrações resultam em dados relativamente brutos, e a manipulação necessária, além de consumir força de trabalho, representa risco de erro humano, impondo ao trabalho etapas de revisão e, eventualmente, retrabalho.

Isso representa um fator que, por si só, justifica o desenvolvimento da presente solução.

Para assegurar a tempestividade do alcance dos objetivos propostos neste trabalho, a automatização das extrações deverá ser concluída até março de 2020, para que os resultados possam ser utilizados na elaboração do Relatório e Parecer Prévio sobre as Contas do Presidente de 2019, bem como no relatório de acompanhamento que analisa diretamente o RGF do 3º quadrimestre de 2019.

Para obtenção dos dados, faz-se necessário acesso ao banco de dados DWTG (Data Warehouse do sistema TesouroGerencial), disponível no Laboratório de Informações de Controle (LabContas), plataforma do TCU que possui bases de dados internalizadas e utiliza a Linguagem de Consulta Estruturada (SQL) como interface para seu sistema de gerenciamento de banco de dados (SGBD).

A utilização da presente solução por outros servidores do TCU requer prévio acesso ao LabContas, bem como ao Siafi e ao TesouroGerencial, a fim de que eventuais detalhes técnicos das consultas possam ser explorados.

Entre os riscos do projeto, registra-se a necessidade de atualização das bases de dados do TesouroGerencial no LabContas. Caso a base não seja atualizada periodicamente, a solução ficará inviabilizada para os quadrimestres futuros.

Outro risco é eventual alteração na metodologia de um dos anexos do RGF, o que demandaria ajuste nos critérios da(s) consulta(s) realizada(s) via SGBD, e a depender da relevância da alteração, pode ser necessário adequar o código desenvolvido na linguagem de programação python, a fim de manter a exatidão das informações geradas.

Visando à mitigação do risco de futura alteração de metodologia pelo MDF, os códigos python estão desenvolvidos para que a substituição de consultas no SGBD ou a adequação da organização dos demonstrativos seja feita de modo simples, para que seja realizado com a menor complexidade possível, de modo que a manutenção da ferramenta possa ser realizada por servidores sem experiência em linguagens de programação.



Pretende-se utilizar a presente solução ao longo dos anos seguintes, razão pela qual a construção das linhas de código buscou redação clara, objetiva, direta e padronizada, aplicando conceitos de programação defensiva para, de forma proativa, evitar quebras de código futuras decorrentes de eventuais alterações nas estruturas e conceitos das bases de dados (Kuznetsov, 2010).

A estrutura da linguagem de programação ora desenvolvida poderá ainda ser utilizada como ponto de partida para automatizar outras atividades semelhantes no âmbito da Semag, sobretudo as periódicas que demandam consultas a sistemas e elaboração de tabelas, liberando força de trabalho para aprofundamento em temas afetos à unidade técnica.

### **3.3 Objetivos de mineração de dados**

O objetivo de mineração de dados deste trabalho é o desenvolvimento de solução para automatização do processo de extração, tratamento, organização e apresentação de dados constantes de anexos do RGF. Conforme definido na metodologia CRISP-DM, o tipo de problema de mineração abordado é o de descrição e sumarização de dados.

Para atingir o objetivo de mineração de dados, os objetivos secundários são: mapeamento das informações constantes dos anexos do RGF a partir metodologia de extração de cada linha de cada demonstrativo analisado, conforme MDF, publicado pela STN; tratamento dos dados obtidos; produção de atributos derivados a partir de dados brutos extraídos; cruzamentos e validações de informações de cada demonstrativo; preenchimento e disponibilização de tabelas exportadas em formato “docx”, prontas para utilização pelo TCU nos relatórios de acompanhamento do RGF; elaboração de minuta de relatório assistido, a partir das principais informações e conclusões de cada tabela referente ao seu respectivo anexo do RGF; e exportação de tabelas em formato de planilha eletrônica “xlsx” de cada demonstrativo analisado.

Para ser considerada bem-sucedida, a presente solução deverá automatizar as etapas as etapas de extração, tratamento, análise e apresentação das informações conforme modelo proposto, que corresponde ao documento já utilizado pelo TCU na instrução referente ao relatório de acompanhamento do RGF.

### **3.4 Plano do projeto**

O trabalho tem início com a leitura dos demonstrativos ora analisados do Relatório de Gestão Fiscal, notadamente das tabelas e da seção que descreve a metodologia. Em seguida, para o correto entendimento das linhas de cada demonstrativo, será estudado o Manual de Demonstrativos Fiscais, publicado pela Secretaria do Tesouro Nacional. O RGF é elaborado e publicado de acordo com a metodologia definida no MDF.

O próximo passo é identificar, junto à equipe de auditoria que realiza periodicamente a análise do RGF, de que forma os dados são obtidos no TesouroGerencial, para fins de mapear as consultas necessárias nesse sistema e traduzi-las para que possam ser realizadas via SGBD.

De posse das consultas via SGBD, deve ser desenvolvido código na linguagem de programação python para, por meio de acesso utilizando a plataforma LabContas do TCU, realizar de forma automática todas as consultas (SGBD) necessárias.

Uma vez obtidos os dados, será construída a estrutura de montagem de cada anexo, o que requer o mapeamento das operações necessárias para o preenchimento de cada campo da tabela. Alguns campos são o resultado direto de uma extração no TesouroGerencial, outros são o resultado de operações aritméticas entre consultas, e há ainda campos que mostram percentuais de determinados valores em relação à receita corrente líquida (RCL) e aos limites estabelecidos pela Lei de Responsabilidade Fiscal.

Entre os produtos finais da solução estão os arquivos em formato “docx” com as tabelas de cada anexo devidamente preenchidas e parágrafos pré-elaborados que descrevem os principais pontos da análise a partir dos resultados obtidos, para fins de entregar à equipe de fiscalização do TCU uma minuta das partes do relatório referentes à cada anexo.

Além disso, planilhas eletrônicas em formato “xlsx” serão exportadas para cada anexo, com a finalidade de eventuais conferências por parte da equipe de fiscalização.

Como resultado final, será evidenciado se cada anexo cumpriu os limites estabelecidos pela RGF, se o limite de alerta foi atingido (90% do limite), ou se algum limite foi ultrapassado, com a proposição de encaminhamento de acordo com cada situação verificada.

## **4 ENTENDIMENTO DOS DADOS**

Esta seção apresentará informações relativas ao entendimento dos dados, fase prévia necessária para subsidiar o desenvolvimento da solução de automatização pretendida.

### **4.1 Histórico**

A apresentação do MDF traz um histórico das finanças públicas a partir da década de 80 do século passado, quando houve a criação da STN em 1986, a implantação do Siafi em 1987 e a instituição da Conta Única do Tesouro Nacional em 1988, mesmo ano de publicação da CF/1988.

Em 2000, a publicação da LRF impôs ao país novas regras fiscais; em 2008 iniciou-se o processo de convergência das práticas de contabilidade do setor público brasileiro às normas internacionais; e em 2009 a Lei Complementar 131 (Lei da Transparência) acrescentou normas à LRF, determinando a disponibilização, em tempo real, de informações sobre a execução orçamentária e financeira.

Já em 2016, com a publicação da Lei Complementar 156, todos os Poderes e órgãos passaram a ter que enviar os demonstrativos fiscais (a exemplo do RGF) ao Sistema de Informações Contábeis e Fiscais do Setor Público Brasileiro (Siconfi), dando amplo acesso público.

## **4.2 Coletando dados iniciais**

A parte IV do Manual de Demonstrações Fiscais válido para o exercício de 2020, publicado pela Secretaria Nacional, dispõe sobre o Relatório de Gestão Fiscal.

Os demonstrativos tratados neste trabalho são: da Dívida Consolidada Líquida, das Garantias e Contragarantias de Valores, e das Operações de Crédito.

### **4.2.1 Demonstrativo da Dívida Consolidada Líquida**

Nos termos do item 04.02.00 do MDF, o demonstrativo da DCL compõe apenas o RGF do Poder Executivo e abrange todos os Poderes da União, dos estados, do Distrito Federal e dos municípios.

Nele há informações sobre a dívida consolidada, dívida mobiliária, dívida contratual, precatórios posteriores a 5/5/2000 e outras dívidas, deduções, dívida consolidada líquida e percentual em relação à receita corrente líquida. São evidenciados ainda precatórios, passivo atuarial e insuficiências financeiras, que não estão incluídos no conceito de dívida consolidada mas causam impacto em sua situação econômico-financeira.

De acordo com o MDF, o detalhamento, a forma e a metodologia de apuração da DCL buscam assegurar a transparência das obrigações contraídas e verificar os limites de endividamento estabelecidos nos normativos.

Os conceitos de dívida pública consolidada ou fundada e de dívida pública mobiliária estão definidos na LRF:

Art. 29. Para os efeitos desta Lei Complementar, são adotadas as seguintes definições:

I - dívida pública consolidada ou fundada: montante total, apurado sem duplicidade, das obrigações financeiras do ente da Federação, assumidas em virtude de leis, contratos, convênios ou tratados e da realização de operações de crédito, para amortização em prazo superior a doze meses;

II - dívida pública mobiliária: dívida pública representada por títulos emitidos pela União, inclusive os do Banco Central do Brasil, Estados e Municípios;

A dívida consolidada líquida (DCL), segundo o MDF, é obtida após deduzidas da dívida consolidada (DC) as disponibilidades de caixa, as aplicações financeiras e os demais haveres financeiros.

A CF/1988 atribui competência privativa ao Senado Federal para fixar, por proposta do Presidente da República, limites globais para o montante da dívida consolidada da União, conforme art. 52, inciso VI.

Já a competência para dispor sobre o montante da dívida mobiliária federal pertence ao Congresso Nacional, com a sanção do Presidente da República (art. 48, inciso XIV da CF/1988).

Ocorre que referidos limites não foram definidos até o momento. Tanto o Projeto de Resolução do Senado 84/2007, que previa o limite de 350% da RCL para a dívida consolidada líquida, como o Projeto de Lei da Câmara 54/2009, que trazia o limite de 650% da RCL para a dívida mobiliária, foram arquivados, de modo que tais limites, por não constarem em normas aprovadas, não são parâmetros considerados pelo TCU na análise de cumprimento aos limites e condições da LRF, como enfatizou o relatório da Semag que embasou a emissão do Acórdão 47/2020-TCU-Plenário, o qual tratou do acompanhamento do RGF do 2º quadrimestre de 2019.

O item 04.02.05.03 do MDF apresenta a tabela do demonstrativo da DCL válido para o Poder Executivo federal, com base no art. 55, inciso I, alínea “b” da LRF:

Figura 2 – Demonstrativo da Dívida Consolidada Líquida

RGF - ANEXO 2 (LRF, art. 55, inciso I, alínea "b")		R\$ milhares		
CÁLCULO DA DÍVIDA CONSOLIDADA LÍQUIDA	SALDO DO EXERCÍCIO ANTERIOR	SALDO DO EXERCÍCIO DE <EXERCÍCIO>		
		Até o 1º Quadrimestre	Até o 2º Quadrimestre	Até o 3º Quadrimestre
DÍVIDA CONSOLIDADA - DC (I)				
Dívida Mobiliária				
Dívida Mobiliária do TN Interna (em mercado)				
(-) Aplicações em Títulos Públicos				
Dívida Mobiliária do TN Interna (em carteira BCB)				
Dívida Securitizada				
Dívida Mobiliária Externa				
Operações de Equalização Cambial - Relacionamento TN/ BCB (Lei nº 11.803/08)				
Dívida Contratual				
Precatórios posteriores a 05/05/2000 (inclusive)				
Dívida Assumida pela União (Lei nº 8.727/93)				
Outras Dívidas				
Passivos reconhecidos com insuficiência de créditos / recursos				
Demais Dívidas				
DEDUÇÕES (II)				
Ativo Disponível				
Depósitos do TN no BCB				
Depósitos à Vista				
Arrecadação a Recolher				
Haveres Financeiros				
Aplicações Financeiras				
Disponibilidades do FAT				
Aplicações de Fundos Diversos Junto ao Setor Privado				
Recursos da Reserva Monetária				
Renegociação de Dívidas de Entes da Federação				
Dívida Renegociada Estados e Municípios (Lei nº 9.496/97 e MP nº 2.185/01)				
Créditos da Lei nº 8.727/93				
Dívida Externa Renegociada (Aviso MF nº 30 e outros)				
Demais Dívidas Renegociadas				
Ajustes para Perdas				
Demais Ativos Financeiros				
Haveres Externos (Garantias)				
Outros Créditos Bancários				
Ajustes para Perdas				
(-) Restos a Pagar Processados				
DÍVIDA CONSOLIDADA LÍQUIDA (DCL) (III) = (I - II)				
RECEITA CORRENTE LÍQUIDA - RCL				
% da DC sobre a RCL (I/RCL)				
% da DCL sobre a RCL (III/RCL)				
LIMITE DEFINIDO POR RESOLUÇÃO DO SENADO FEDERAL - <%>				

Fonte: Manual de Demonstrativos Fiscais válido para 2020, publicado pela Secretaria do Tesouro Nacional.

Em seguida, o MDF detalha a metodologia para preenchimento de cada campo do demonstrativo.

Referida metodologia é traduzida em classificadores utilizados pelo Siafi em item próprio do Relatório de Gestão Fiscal, publicado pelo Poder Executivo federal.

Cada campo é o resultado de operações aritméticas entre contas contábeis do Siafi, aplicando-se, em cada caso, determinados filtros, a exemplo de conta corrente, tipo de administração, órgão, indicador de superávit financeiro (ISF), órgão, unidade gestora e ação de governo.

Apresentam-se a seguir os critérios para a obtenção do valor que deve preencher a linha “Dívida Externa Renegociada (Aviso MF nº 30 e outros)” do demonstrativo da DCL, como forma de exemplificar a metodologia.

Figura 3 – Exemplo de Metodologia de Apuração de Item da DCL

Dívida Externa Renegociada (Aviso MF nº 30 e outros)	Código	Descrição
	11241.01.00	Empréstimos Concedidos a Receber
	11241.04.01	Encargos sobre Empréstimos Concedidos a Receber – Exceto FAT
	11241.06.00	Resíduo de Empréstimos Concedidos
	11244.01.00	Empréstimos Concedidos a Receber – Inter Estados
	11244.04.01	Encargos sobre Empréstimos Concedidos a Receber – Inter Estados
	11244.06.00	Resíduo de Empréstimos Concedidos – Inter Estados
	11245.01.00	Empréstimos Concedidos a Receber – Inter Municípios
	11245.04.01	Encargos sobre Empréstimos Concedidos a Receber – Inter Municípios
	11245.06.00	Resíduo de Empréstimos Concedidos – Inter Municípios
	12111.03.01	Empréstimos Concedidos a Receber
	12111.03.18	Resíduo de Empréstimos Concedidos
	12114.03.01	Empréstimos Concedidos a Receber – Estados
	12114.03.18	Resíduo de Empréstimos Concedidos – Estados
	12115.03.01	Empréstimos Concedidos a Receber – Municípios
	12115.03.18	Resíduo de Empréstimos Concedidos – Municípios
	12121.98.18	Créditos Sub-Rogados
	12124.98.18	Créditos Sub-Rogados – Estados
	11381.42.00	Créditos Sub-Rogados
	11384.42.00	Créditos Sub-Rogados – Estados
	11385.42.00	Créditos Sub-Rogados – Município
	<b>Critérios</b>	
	Unidade Gestora	170512 – "Coordenação-Geral de Haveres Financeiros (COAFI)".
	Apenas os saldos identificados em nível de conta corrente relacionados à Dívida Externa Renegociada (Aviso MF nº 30 e outros).	
	Identificados pela Entidade no Conta Corrente que contenham os seguintes termos: PF1705104; PF1705117; PF1705521; PF1705534; PF1705116; PF1705531; PF1705532; PF1705113; PF1701536; PF1705520; PF1705533; PF1705464; PF1705534; PF1705119; BIB; PF1705114; MF 030; FRANÇA; EXTER; FRANÇA; DMLP; PF1705384.	

Fonte: Manual de Demonstrativos Fiscais válido para 2020.

As demais linhas do demonstrativo são alimentadas por informações produzidas segundo critério semelhante, qual seja, contas contábeis e filtros de classificadores do Sifafi.

Para a conferência de cada campo do demonstrativo, o TCU realiza uma ou mais consultas no TesouroGerencial. Uma tarefa inicial do processo de automatização da extração dos dados é identificar como cada consulta é feita no sistema TesouroGerencial, de modo que tal consulta possa ser realizada por meio de SGBD via LabContas, utilizando o banco de dados DWTG.

Essa fase do trabalho contou com avanços prévios realizados por auditores da Semag, tanto da equipe que realiza periodicamente a análise do RGF quanto da equipe encarregada de automatizar etapas referentes a determinados anexos do RGF.

#### 4.2.2 Demonstrativo das Garantias e Contragarantias de valores

O item 04.03.00 do MDF discorre sobre o Demonstrativo das Garantias e Contragarantias de Valores, o qual compõe apenas o RGF do Poder Executivo.

São informados os valores das garantias concedidas pelo ente na contratação de operações de créditos realizadas pelo garantidos, sendo divididas em garantias aos estados, aos municípios, às entidades controladas e por meio de fundos. Há ainda um detalhamento, em cada item, em operações de crédito externas e internas.

Os objetivos declarados no MDF são dar transparência às garantias oferecidas a terceiros pelo ente e verificar o cumprimento dos limites da LRF, bem como das contragarantias vinculadas, as quais decorrem de operações de crédito internas e externas.

Segue conceito de garantia dado pelo MDF:

04.03.02 CONCEITO

04.03.02.01 Garantia

Garantia é o compromisso de adimplência de obrigação financeira ou contratual assumida pelo ente da Federação por ocasião da realização de operações de crédito por suas estatais não dependentes ou por outros entes da federação. São fianças e avais concedidos pelo ente federativo, em operações de crédito, inclusive com recursos de fundos de aval, a assunção de risco creditício em linhas de crédito, o seguro de crédito à exportação e outras garantias de natureza semelhante que representem compromisso de adimplência de obrigação financeira ou contratual.

Referido conceito está alinhado ao disposto no art. 9º, § 1º da Resolução 48/2007 do Senado Federal:

§ 1º Consideram-se garantia concedida, para os efeitos deste artigo, as fianças e avais concedidos direta ou indiretamente pela União, em operações de crédito, inclusive com recursos de fundos de aval, a assunção de risco creditício em linhas de crédito, o seguro de crédito à exportação e outras garantias de natureza semelhante que representem compromisso de adimplência de obrigação financeira ou contratual.

A mesma Resolução 48/2007 estabelece que o montante das garantias concedidas pela União está limitado a 60% da RCL.

O item 04.03.05 do MDF apresenta a tabela referente ao anexo 3 do RGF, que trata do Demonstrativo das Garantias e Contragarantias de Valores:

Figura 4 – Demonstrativo das Garantias e Contragarantias de Valores

RGF - ANEXO 3 (LRF, art. 55, inciso I, alínea "c" e art. 40, § 1º)

GARANTIAS CONCEDIDAS	SALDO DO EXERCÍCIO ANTERIOR	SALDOS DO EXERCÍCIO DE <EXERCÍCIO>		
		Até o 1º Quadrimestre	Até o 2º Quadrimestre	Até o 3º Quadrimestre
<b>AOS ESTADOS (I)</b>				
Em Operações de Crédito Externas				
Em Operações de Crédito Internas				
<b>AOS MUNICÍPIOS (II)</b>				
Em Operações de Crédito Externas				
Em Operações de Crédito Internas				
<b>ÀS ENTIDADES CONTROLADAS (III)</b>				
Em Operações de Crédito Externas				
Em Operações de Crédito Internas				
<b>POR MEIO DE FUNDOS E PROGRAMAS (IV)</b>				
<b>TOTAL GARANTIAS CONCEDIDAS (V) = (I + II + III + IV)</b>				
<b>RECEITA CORRENTE LÍQUIDA - RCL (VI)</b>				
(-) Transferências obrigatórias da União relativas às emendas individuais (art. 166-A, § 1º, da CF) (VII)				
<b>RECEITA CORRENTE LÍQUIDA AJUSTADA PARA CÁLCULO DOS LIMITES DE ENDIVIDAMENTO (VIII) = (VI - VII)</b>				
% do TOTAL DAS GARANTIAS sobre a RCL AJUSTADA (V/VIII)				
LIMITE DEFINIDO POR RESOLUÇÃO DO SENADO FEDERAL - <%>				
LIMITE DE ALERTA (inciso III do §1º do art. 59 da LRF) - <%>				
<b>CONTRAGARANTIAS RECEBIDAS</b>	<b>SALDO DO EXERCÍCIO ANTERIOR</b>	<b>SALDOS DO EXERCÍCIO DE &lt;EXERCÍCIO&gt;</b>		
		Até o 1º Quadrimestre	Até o 2º Quadrimestre	Até o 3º Quadrimestre
<b>DOS ESTADOS (IX)</b>				
Em Garantia às operações de Crédito Externas				
Em Garantia às operações de Crédito Internas				
<b>DOS MUNICÍPIOS (X)</b>				
Em Garantia às operações de Crédito Externas				
Em Garantia às operações de Crédito Internas				
<b>DAS ENTIDADES CONTROLADAS (XI)</b>				
Em Garantia às operações de Crédito Externas				
Em Garantia às operações de Crédito Internas				
<b>EM GARANTIAS POR MEIO DE FUNDOS E PROGRAMAS (XII)</b>				
<b>TOTAL CONTRAGARANTIAS RECEBIDAS (XIII) = (IX + X + XI + XII)</b>				
<b>MEDIDAS CORRETIVAS:</b>				

Fonte: Manual de Demonstrativos Fiscais válido para 2020.

Em seguida, o MDF discorre sobre como deve ser preenchido cada campo do demonstrativo.

Assim como ocorre com o demonstrativo da DCL, o RGF publicado pelo Poder Executivo federal apresenta a metodologia de cálculo de cada campo em contas contábeis e classificadores do Siafi.

A metodologia consiste basicamente em critérios relacionados a contas contábeis e contas correntes, conforme o seguinte extrato:

Figura 5 – Exemplo de Metodologia de item do Demonstrativo das Garantias e Contragarantias de Valores

GARANTIAS CONCEDIDAS	
<b>AOS ESTADOS (I)</b>	Soma das Operações Internas e Externas.
Em Operações de Crédito Externas	Conta Contábil 81211.02.04 – Fianças a Executar Conta Corrente CG0000069
Em Operações de Crédito Internas	Conta Contábil 81211.01.04 – Fianças a Executar Conta Corrente CG0000064
<b>AOS MUNICÍPIOS (II)</b>	Soma das Operações Internas e Externas.
Em Operações de Crédito Externas	Conta Contábil 81211.02.04 – Fianças a Executar Conta Corrente CG0000070
Em Operações de Crédito Internas	Conta Contábil 81211.01.04 – Fianças a Executar Conta Corrente CG0000065
<b>ÀS ENTIDADES CONTROLADAS (III)</b>	Soma das Operações Internas e Externas.
Em Operações de Crédito Externas	Conta Contábil 81211.02.04 – Fianças a Executar Conta Corrente CG0000071 CG0000072
Em Operações de Crédito Internas	Conta Contábil 81211.01.04 – Fianças a Executar Conta Corrente CG0000066 CG0000067
<b>POR MEIO DE FUNDOS E PROGRAMAS (IV)</b>	Conta Contábil 81211.01.04 – Fianças a Executar Conta Corrente CG0000068;CGFSCAIRB;CGLEI8036;CGPPRONAF;CGPRACAU
<b>TOTAL GARANTIAS CONCEDIDAS (V) = (I + II + III + IV)</b>	



Fonte: Manual de Demonstrativos Fiscais válido para 2020.

De posse desses critérios, são realizadas consultas ao sistema TesouroGerencial para obter as informações, de acordo com o período estabelecido. A fase de automatização da extração se propõe a realizar, por meio de linhas de programação na linguagem python, todas as consultas necessárias no TesouroGerencial, com acesso via LabContas ao banco de dados DWTG.

### 4.2.3 Demonstrativo das Operações de Crédito

O demonstrativo das operações de crédito está descrito no item 04.04.00 do MDF, e também é publicado apenas no RGF do Poder Executivo, abrangendo as operações de crédito internas e externas, inclusive por antecipação da receita.

O referido demonstrativo apresenta ainda informações sobre o percentual de operações de crédito em relação à RCL, operações vedadas, antecipação da receita orçamentária e os limites máximo e de alerta constante nos normativos.

Evidenciam-se ainda operações que, embora integrem a dívida consolidada, não são computadas para apuração dos limites de operações de crédito, a exemplo dos parcelamentos de dívidas e operações de reestruturação e recomposição do principal de dívidas. As operações registradas no demonstrativo independem do ingresso de receitas orçamentárias nos cofres públicos.

O art. 29 da LRF define o conceito de operação de crédito:

Art. 29. Para os efeitos desta Lei Complementar, são adotadas as seguintes definições:

(...)

III - operação de crédito: compromisso financeiro assumido em razão de mútuo, abertura de crédito, emissão e aceite de título, aquisição financiada de bens, recebimento antecipado de valores provenientes da venda a termo de bens e serviços, arrendamento mercantil e outras operações assemelhadas, inclusive com o uso de derivativos financeiros;

De acordo com o MDF, as operações de crédito possuem, pelo menos, uma das seguintes características:

- Envolvem o reconhecimento, por parte do setor público, de um passivo, que equivale a um aumento do endividamento público com impactos no montante da dívida pública e na capacidade de endividamento do ente;
- Pressupõem a existência de risco de não adimplemento de obrigações que, em geral, materializa-se na forma de cobrança de juros explícitos ou implícitos, deságio e demais encargos financeiros, tendo como consequência uma redução do Patrimônio Líquido do ente que equivale a um aumento do valor original da dívida; e
- Diferimento no tempo, uma vez que, em regra, as operações de crédito envolvem o recebimento de recursos financeiros, bens, ou prestação de serviços, os quais terão como contrapartida a incorporação de uma dívida a ser quitada em momento futuro.

A LRF e o MDF apresentam ainda vedações relacionadas às operações de crédito. Nos termos dos art. 35 a 37 da LC 101/2000:

Art. 35. É vedada a realização de operação de crédito entre um ente da Federação, diretamente ou por intermédio de fundo, autarquia, fundação ou empresa estatal dependente, e outro, inclusive suas entidades da administração indireta, ainda que sob a forma de novação, refinanciamento ou postergação de dívida contraída anteriormente.

§ 1º Excetuam-se da vedação a que se refere o caput as operações entre instituição financeira estatal e outro ente da Federação, inclusive suas entidades da administração indireta, que não se destinem a:

I - financiar, direta ou indiretamente, despesas correntes;

II - refinarciar dívidas não contraídas junto à própria instituição concedente.

§ 2º O disposto no caput não impede Estados e Municípios de comprar títulos da dívida da União como aplicação de suas disponibilidades.

Art. 36. É proibida a operação de crédito entre uma instituição financeira estatal e o ente da Federação que a controle, na qualidade de beneficiário do empréstimo.

Parágrafo único. O disposto no caput não proíbe instituição financeira controlada de adquirir, no mercado, títulos da dívida pública para atender investimento de seus clientes, ou títulos da dívida de emissão da União para aplicação de recursos próprios.

Art. 37. Equiparam-se a operações de crédito e estão vedados:

I - captação de recursos a título de antecipação de receita de tributo ou contribuição cujo fato gerador ainda não tenha ocorrido, sem prejuízo do disposto no § 7º do art. 150 da Constituição;

II - recebimento antecipado de valores de empresa em que o Poder Público detenha, direta ou indiretamente, a maioria do capital social com direito a voto, salvo lucros e dividendos, na forma da legislação;

III - assunção direta de compromisso, confissão de dívida ou operação assemelhada, com fornecedor de bens, mercadorias ou serviços, mediante emissão, aceite ou aval de título de crédito, não se aplicando esta vedação a empresas estatais dependentes;

IV - assunção de obrigação, sem autorização orçamentária, com fornecedores para pagamento a posteriori de bens e serviços.

A Constituição Federal de 1988 atribuiu competência privativa ao Senado Federal para dispor sobre limites globais e condições para as operações de crédito externo e interno da União, conforme art. 52, inciso VII.

A definição do limite foi feita pela Resolução 48/2007 do Senado Federal, em seu art. 7º, inciso I, qual seja, 60% da RCL. Considerando que a Resolução 48/2007 não estabelece limites para as operações de crédito por antecipação da receita orçamentária, as linhas que se referem a tal ponto não devem ser preenchidas.

Segue tabela modelo para preenchimento do demonstrativo de operações de crédito da União, nos termos do MDF:

Figura 6 – Demonstrativo das Operações de Crédito

OPERÇÕES DE CRÉDITO	VALOR REALIZADO	
	No Quadrimestre de Referência	Até o Quadrimestre de Referência (a)
<b>OPERÇÕES DE CRÉDITO (I)</b> Mobiliária Interna Refinanciamento Assunção, Reconhecimento e Confissão de Dívidas (LRF, art. 29, § 1º) Outras Internas - Orçamentárias Outras Internas - Extraorçamentárias Aporte Bacen Lei nº 11.803/2008 Aportes em Empresas Trocas e Demais Operações Internas Externa Refinanciamento Assunção, Reconhecimento e Confissão de Dívidas (LRF, art. 29, § 1º) Outras Operações Mobiliárias Externas Contratual Interna Abertura de Crédito Assunção, Reconhecimento e Confissão de Dívidas (LRF, art. 29, § 1º) Outras Operações Contratuais Internas Externa Abertura de Crédito - Orçamentárias Abertura de Crédito - Extraorçamentárias Assunção, Reconhecimento e Confissão de Dívidas (LRF, art. 29, § 1º) Outras Operações Contratuais Externas		
<b>APURAÇÃO DO CUMPRIMENTO DOS LIMITES</b>	<b>VALOR</b>	<b>% SOBRE A RCL</b>
RECEITA CORRENTE LÍQUIDA – RCL		
OPERÇÕES VEDADAS (II)		
<b>OPERÇÕES DEDUZIDAS DO LIMITE (III)</b> Amortização/Refinanciamento do Principal de Dívidas Cancelamento de títulos aceitos em leilões de trocas Aporte Bacen Lei 11.803/2008 Concessão de Garantias		
TOTAL CONSIDERADO PARA FINS DA APURAÇÃO DO CUMPRIMENTO DO LIMITE (IV) = (Ia + II - III)		
LIMITE GERAL DEFINIDO POR RESOLUÇÃO DO SENADO FEDERAL PARA AS OPERÇÕES DE CRÉDITO INTERNAS E EXTERNAS		
OPERÇÕES DE CRÉDITO POR ANTECIPAÇÃO DA RECEITA ORÇAMENTÁRIA		
LIMITE DEFINIDO POR RESOLUÇÃO DO SENADO FEDERAL PARA AS OPERÇÕES DE CRÉDITO POR ANTECIPAÇÃO DA RECEITA		

Fonte: Manual de Demonstrativos Fiscais válido para 2020.

O MDF discorre sobre o que deve conter cada campo desse demonstrativo, e o Relatório de Gestão Fiscal publicado pelo Poder Executivo federal converte as disposições do MDF referentes à metodologia de preenchimento do demonstrativo em contas contábeis e demais instrumentos e classificadores constantes do Siafi.

Além das contas contábeis, são utilizados como critérios para os campos os classificadores “natureza de receita” e “grupo de despesa”.

A seguir, exemplifica-se a situação descrita com os critérios para preenchimento do campo “refinanciamento”, constante das operações de crédito mobiliárias internas:

Figura 7 – Exemplo de Metodologia de item do Demonstrativo das Operações de Crédito

DEMONSTRATIVO DAS OPERAÇÕES DE CRÉDITO	
<b>OPERÇÕES DE CRÉDITO (I)</b>	Soma das Operações Mobiliárias e Contratuais.
<b>Mobiliária</b>	Soma das Operações Internas e Externas.
<b>Interna</b>	Soma dos itens abaixo.
<b>Refinanciamento</b>	62131.00.00 Restituições 62132.00.00 Retificações 62133.00.00 Compensações 62134.00.00 Incentivos Fiscais 62139.00.00 Outras Deduções da Receita Orçamentária 62120.00.00 Receita Realizada
<b>Crítérios</b>	
Natureza de Receita	Começa com: 2111002
Natureza de Receita	Começa com: 8111002

Fonte: Manual de Demonstrativos Fiscais.

Com o mapeamento dos critérios para preenchimento de cada campo do demonstrativo, o próximo passo é realizar as extrações no sistema TesouroGerencial, o que poderá ocorrer de forma automática e para todas as consultas necessárias com a execução do código elaborado

na linguagem de programação python, o qual acessa o banco de dados DWTG por meio da ferramenta LabContas.

### 4.3 Descrevendo os dados

Serão detalhados conceitos de itens apresentados na seção anterior, com a finalidade de contribuir para o pleno entendimento de cada campo dos demonstrativos analisados neste trabalho, com ênfase no atributo conta contábil, principal componente das consultas necessárias para preenchimento de cada campo dos anexos do RGF.

O Manual de Contabilidade Aplicada ao Setor Público (MCASP) disciplina, no âmbito nacional, procedimentos que permitem e regulamentam o registro da aprovação e da execução do orçamento, resgatam o patrimônio como objeto da contabilidade, e buscam a convergência aos padrões internacionais, respeitando a legislação nacional vigente e os princípios da ciência contábil.

O MCASP válido para 2019 em diante foi aprovado pelas seguintes portarias:

- Portaria STN nº 877, de 18/12/2018, que aprova as Partes Geral, II – Procedimentos Contábeis Patrimoniais, III – Procedimentos Contábeis Específicos, IV – Plano de Contas Aplicado ao Setor Público e V – Demonstrações Contábeis Aplicadas ao Setor Público;
- Portaria Conjunta STN/SOF nº 6, de 18/12/2018, que aprova a Parte I - Procedimentos Contábeis Orçamentários;
- Portaria Conjunta STN/SPREV nº 7, de 18/12/2018, que aprova a Parte III - Procedimentos Contábeis Específicos: Capítulo 4 – Regime Próprio de Previdência Social (RPPS).

Integra o MCASP o Plano de Contas Aplicado ao Setor Público (PCASP), que é uma estrutura básica de escrituração contábil que permite o registro de atos e fatos de forma padronizada e sistematizada, a partir do método das partidas dobradas. É formado por uma relação padronizada de contas contábeis e permite a elaboração de relatórios gerenciais e de demonstrações contábeis úteis aos usuários, a exemplo do Relatório de Gestão Fiscal.

A conta contábil, elemento basilar do PCASP, permite expressar qualitativa e quantitativamente fatos de mesma natureza, evidenciando a composição, variação e estado do patrimônio, e ainda os bens, direitos obrigações e situações que possam afetá-lo direta ou indiretamente.

O PCASP está estruturado em oito classes, de acordo com a natureza da informação, conforme se demonstra a seguir:

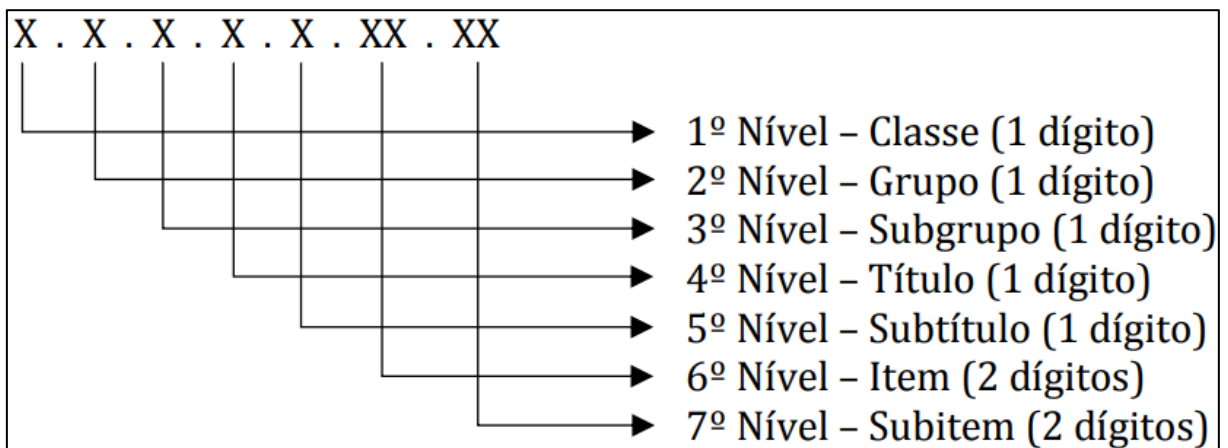
Figura 8 – Classes de Informação do Plano de Contas Aplicado ao Setor Público

PCASP		
Natureza da informação	Classes	
Patrimonial	1. Ativo	2. Passivo
	3. Variações Patrimoniais Diminutivas	4. Variações Patrimoniais Aumentativas
Orçamentária	5. Controles da Aprovação do Planejamento e Orçamento	6. Controles da Execução do Planejamento e Orçamento
Controle	7. Controles Devedores	8. Controles Credores

Fonte: Manual de Contabilidade Aplicada ao Setor Público válido a partir de 2019.

O código das contas contábeis do PCASP possui 7 níveis de desdobramento e 9 dígitos:

Figura 9 – Código das Contas Contábeis do Plano de Contas Aplicado ao Setor Público



Fonte: Manual de Contabilidade Aplicada ao Setor Público válido a partir de 2019.

A figura a seguir detalha os 2 primeiros níveis, classe e grupo:

Figura 10 – Classes e Grupos do Plano de Contas Aplicado ao Setor Público

<b>PCASP</b>	
<b>1 – Ativo</b> 1.1 - Ativo Circulante 1.2 - Ativo Não Circulante	<b>2 – Passivo e Patrimônio Líquido</b> 2.1 - Passivo Circulante 2.2 - Passivo Não Circulante 2.3 - Patrimônio Líquido
<b>3 – Variação Patrimonial Diminutiva</b> 3.1 - Pessoal e Encargos 3.2 - Benefícios Previdenciários e Assistenciais 3.3 - Uso De Bens, Serviços e Consumo de Capital Fixo 3.4 - Variações Patrimoniais Diminutivas Financeiras 3.5 - Transferências e Delegações Concedidas 3.6 - Desvalorização e Perda De Ativos e Incorporação de Passivos 3.7 - Tributárias 3.8 - Custo das Mercadorias Vendidas, dos Produtos Vendidos e dos Serviços Prestados 3.9 - Outras Variações Patrimoniais Diminutivas	<b>4 – Variação Patrimonial Aumentativa</b> 4.1 - Impostos, Taxas e Contribuições de Melhoria 4.2 - Contribuições 4.3 - Exploração e venda de bens, serviços e direitos 4.4 - Variações Patrimoniais Aumentativas Financeiras 4.5 - Transferências e Delegações Recebidas 4.6 - Valorização e Ganhos Com Ativos e Desincorporação de Passivos 4.9 - Outras Variações Patrimoniais Aumentativas
<b>5 – Controles da Aprovação do Planejamento e Orçamento</b> 5.1 - Planejamento Aprovado 5.2 - Orçamento Aprovado 5.3 - Inscrição de Restos a Pagar	<b>6 – Controles da Execução do Planejamento e Orçamento</b> 6.1 - Execução do Planejamento 6.2 - Execução do Orçamento 6.3 - Execução de Restos a Pagar
<b>7 – Controles Devedores</b> 7.1 - Atos Potenciais 7.2 - Administração Financeira 7.3 - Dívida Ativa 7.4 - Riscos Fiscais 7.5 - Consórcios Públicos 7.8 - Custos 7.9 - Outros Controles	<b>8 – Controles Credores</b> 8.1 - Execução dos Atos Potenciais 8.2 - Execução da Administração Financeira 8.3 - Execução da Dívida Ativa 8.4 - Execução dos Riscos Fiscais 8.5 - Execução dos Consórcios Públicos 8.8 - Apuração de Custos 8.9 - Outros Controles

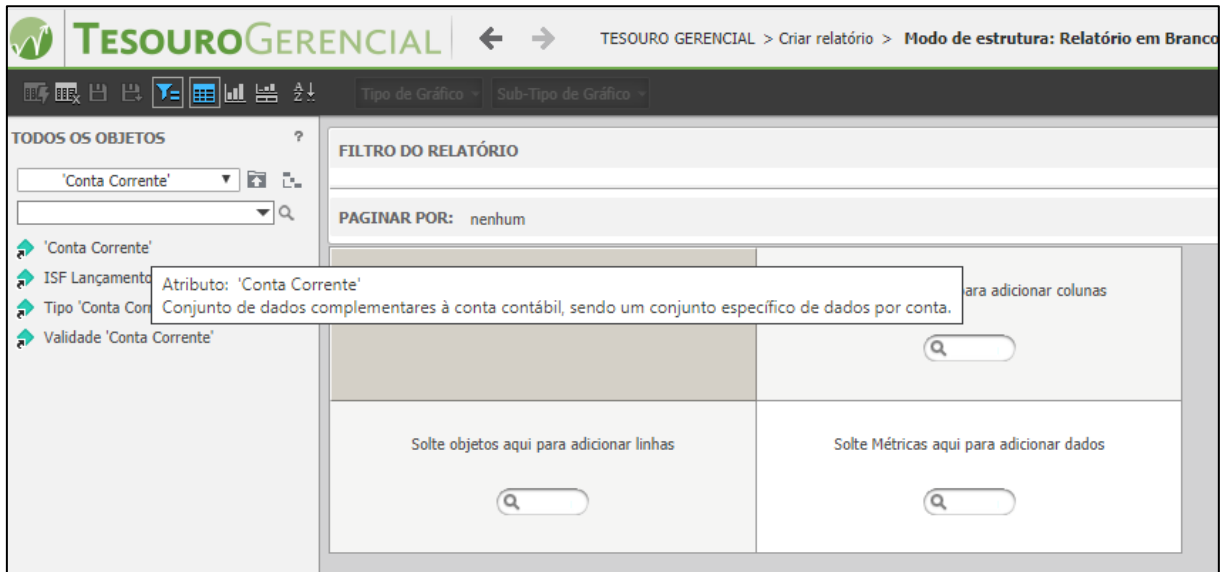
Fonte: Manual de Contabilidade Aplicada ao Setor Público válido a partir de 2019.

No anexo 2 do RGF, que trata do demonstrativo da DCL, são utilizadas contas contábeis pertencentes às classes 1 – Ativo, 2 – Passivo e Patrimônio Líquido, 5 – Controles da Aprovação do Planejamento e Orçamento, 6 – Controles da Execução do Planejamento e Orçamento, e 8 – Controles Credores.

Os demais classificadores utilizados no demonstrativo da DCL serão expostos a seguir.

O atributo “Conta Corrente” é um conjunto de dados complementares à conta contábil, sendo um conjunto específico de dados por conta, conforme se extrai do TesouroGerencial:

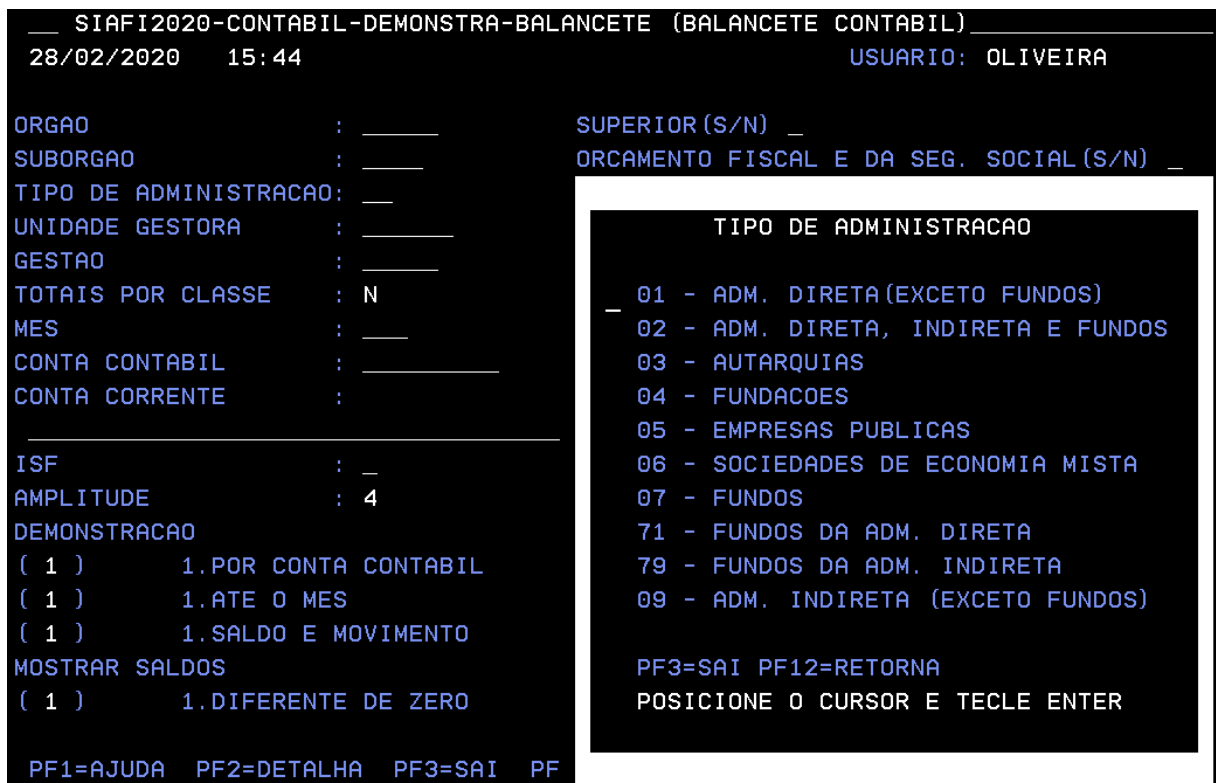
Figura 11 – Conceito de Conta Corrente utilizado pelo Sistema TesouroGerencial



Fonte: Sistema TesouroGerencial.

Já o “Tipo de Administração” é um classificador referente ao aspecto institucional que permite as seguintes opções, de acordo com o Siafi:

Figura 12 – Tipos de Administração constantes no Siafi



Fonte: Sistema Integrado de Administração Financeira.

O MCASP conceitua “Órgão Orçamentário” como o agrupamento de unidades orçamentárias. Já a definição de unidade orçamentária consta no art. 14 da Lei 4.320/1964, qual

seja, “agrupamento de serviços subordinados ao mesmo órgão ou repartição a que serão consignadas dotações próprias”.

A definição de “Unidade Gestora” é citada no Manual do Siafi (item 010400 – Glossário) como sendo:

Unidade Orçamentária ou Administrativa que realiza atos de gestão orçamentária, financeira e/ou patrimonial, cujo titular, em consequência, está sujeito a tomada de contas anual na conformidade do disposto nos artigos 81 e 82 do Decreto-lei nr. 200, de 25 de fevereiro de 1967.

Segundo o MCASP, a “Ação” é uma operação da qual resultam bens ou serviços que contribuem para o objetivo de um programa. Também estão incluídas no conceito de ação as transferências a entes da federação e a pessoas físicas ou jurídicas.

Por fim, “ISF Lançamento” é o Indicador de Superávit Financeiro, assume os valores “N – Não se aplica”, “F – Financeiro”, “P – Permanente” ou “X – Financeiro ou Permanente”, conforme o Tesouro Gerencial.

Com relação ao anexo 3 do RGF – Demonstrativo das Garantias e Contragarantias de Valores, os principais critérios usados nas consultas são conta contábil e conta corrente. Ambos os classificadores já foram conceituados nesta seção.

Quanto ao anexo 4 – Demonstrativo das Operações de Crédito, além do classificador conta contábil, são utilizados como filtros o “Grupo de Despesa” e a “Natureza de Receita”.

O MCASP define grupo de natureza de despesa (GND) como sendo um agregador de elementos de despesa orçamentária cujo objeto de gasto possui as mesmas características. Os GNDs são 1 – Pessoal e Encargos Sociais; 2 – Juros e Encargos da Dívida; 3 – Outras Despesas Correntes; 4 – Investimentos; 5 – Inversões Financeiras; e 6 – Amortização da Dívida.

A natureza de receita encontra amparo legal na Lei 4.320/1964:

Art. 8º A discriminação da receita geral e da despesa de cada órgão do Governo ou unidade administrativa, a que se refere o artigo 2º, § 1º, incisos III e IV obedecerá à forma do Anexo nº 2.

§ 1º Os itens da discriminação da receita e da despesa, mencionados nos artigos 11, § 4º, e 13, serão identificados por números de códigos decimal, na forma dos Anexos nºs 3 e 4.

(...)

Art. 11 - A receita classificar-se-á nas seguintes categorias econômicas: Receitas Correntes e Receitas de Capital.

§ 1º - São Receitas Correntes as receitas tributária, de contribuições, patrimonial, agropecuária, industrial, de serviços e outras e, ainda, as provenientes de recursos financeiros recebidos de outras pessoas de direito público ou privado, quando destinadas a atender despesas classificáveis em Despesas Correntes.



§ 2º - São Receitas de Capital as provenientes da realização de recursos financeiros oriundos de constituição de dívidas; da conversão, em espécie, de bens e direitos; os recursos recebidos de outras pessoas de direito público ou privado, destinados a atender despesas classificáveis em Despesas de Capital e, ainda, o superávit do Orçamento Corrente.

Nos termos do MCASP, “natureza de receita é a menor célula de informação no contexto orçamentário para as receitas públicas”, auxiliando a elaboração de análises da atuação estatal no aspecto econômico-financeiro.

## 5 PREPARAÇÃO DOS DADOS

A seguir, serão descritas as etapas de seleção, tratamento e construção de novos dados.

### 5.1 Selecionando dados

Os dados necessários ao presente trabalho são obtidos por meio de consultas ao banco de dados DWTG via ferramenta LabContas, após identificação e mapeamento dos critérios e fontes utilizadas para elaboração de cada demonstrativo ora analisado, conforme metodologia constante do Manual de Demonstrativos Fiscais, publicado pela Secretaria do Tesouro Nacional, documento que detalha e padroniza a estrutura de cada anexo do Relatório de Gestão Fiscal.

Para o anexo 2 – Demonstrativo da Dívida Consolidada Líquida, são realizadas 27 extrações por período via SGBD no LabContas, por meio de código de linguagem de programação python.

No anexo 3 – Demonstrativo das Garantias e Contragarantias de Valores, são necessárias 8 consultas por período.

Já o anexo 4 – Demonstrativo das Operações de Crédito possui 17 consultas por período.

Cada consulta gera um dado de saída que é utilizado como insumo para alimentação de um ou mais campos de cada demonstrativo.

### 5.2 Tratando dados

Observou-se que algumas consultas geram resultado zero, fazendo com que seja exibido “*none*” como dado de saída. Nesses casos, os códigos de programação seguintes retornam mensagem de erro.

Para sanar o erro, os resultados de saída “*none*” foram substituídos por zero, a fim de possibilitar a execução dos cálculos seguintes, necessários à realização da solução ora desenvolvida.

### 5.3 Construindo novos dados

Considerando que os resultados das consultas realizadas no banco de dados DWTG via LabContas são parciais, faz-se necessário realizar operações via linguagem de programação python para obtenção das informações organizadas e formatadas conforme cada campo de cada demonstrativo do RGF.

Os novos dados foram construídos a partir de operações aritméticas entre os resultados obtidos pelas consultas ao TesouroGerencial, soma/subtração de itens, variação percentual entre o resultado de dois períodos.

Os dados gerados são utilizados tanto nas tabelas de cada demonstrativo quanto nos parágrafos constantes da minuta de relatório assistido referente à cada anexo analisado.

### 5.4 Integrando dados

Em todos os demonstrativos ora analisados, há campos que se referem a percentuais em relação à receita corrente líquida (RCL). Para tanto, foi importado via código de linguagem de programação python tabela em formato “xlsx” com os valores referentes à RCL em cada quadrimestre, a partir do exercício de 2011.

Dessa forma, montaram-se dataframes que integram os resultados tratados das consultas ao TesouroGerencial com os montantes de RCL obtidos por período, para fins de utilização nas etapas posteriores deste trabalho.

## 6 MODELAGEM

Para um dos produtos finais de cada demonstrativo, foi utilizado um modelo de documento em formato “docx”. Por meio do pacote docx-template, disponível na linguagem de programação python, criaram-se referências para que cada campo de cada tabela fosse alimentado por um resultado, de acordo com a estrutura definida para os demonstrativos.

A estrutura de automatização das consultas pelo período desejado, assim como a identificação, mapeamento, limpeza, formatação, organização e tratamento dos dados foi elaborada de forma padronizada para que fosse replicada para os demais demonstrativos de forma relativamente simples, inteligível e cuja utilização e manutenção sejam possíveis de serem realizadas por auditores que não são especialistas em linguagens de programação.

Sempre que possível, utilizou-se o artifício das estruturas de repetição para que apenas um código pudesse ser utilizado para diversas consultas, resultados e períodos.

Cada consulta ao TesouroGerencial via SGBD foi tratada com o nome “query(i)”, com (i) variando de 1 até o número total de consultas. A conversão dos resultados no formato de valor desejado transforma query(i) em query(i)v, e cada linha do demonstrativo foi tratada como “L1”, “L2”, “L3” etc. Quando houve necessidade de referenciar o campo completo, isto é, linha e coluna de cada tabela, utilizou-se “L1C1”, “L1C2” etc.

Para subsidiar a minuta de relatório assistido, os dados necessários foram tratados e formatados para alimentar os campos devidos, utilizando-se códigos no formato “Z1\_1”, “Z1\_2”, “Z2\_1”, “Z2\_2” etc.

Seguem extratos dos templates utilizados.

Figura 13 – Extrato do Template Base para o Demonstrativo da Dívida Consolidada Líquida

<b>Tabela xx – Demonstrativo da Dívida Consolidada Líquida</b>							
(LRF, art. 55, inciso I, alínea "b")							
R\$ milhares							
ESPECIFICAÇÃO	SALDO {{ano_anterior}}	Até o 1º Quadr	Var Perc	Até o 2º Quadr	Var Perc	Até o 3º Quadr	Var Perc
<b>DÍVIDA CONSOLIDADA - DC (I)</b>	{{L1_C1}}	{{L1_C2}}	{{L1_C3}}	{{L1_C4}}	{{L1_C5}}	{{L1_C6}}	{{L1_C7}}
<i>Dívida Mobiliária do Tesouro Nacional</i>	{{L2_C1}}	{{L2_C2}}	{{L2_C3}}	{{L2_C4}}	{{L2_C5}}	{{L2_C6}}	{{L2_C7}}
<i>Oper de Eq Camb - Rel TN/BCB</i>	{{L3_C1}}	{{L3_C2}}	{{L3_C3}}	{{L3_C4}}	{{L3_C5}}	{{L3_C6}}	{{L3_C7}}
<i>Dívida Contratual</i>	{{L4_C1}}	{{L4_C2}}	{{L4_C3}}	{{L4_C4}}	{{L4_C5}}	{{L4_C6}}	{{L4_C7}}
<i>Precatórios posteriores a 5/5/2000</i>	{{L5_C1}}	{{L5_C2}}	{{L5_C3}}	{{L5_C4}}	{{L5_C5}}	{{L5_C6}}	{{L5_C7}}
<i>Div Assumida pela União (Lei 8.727/93)</i>	{{L6_C1}}	{{L6_C2}}	{{L6_C3}}	{{L6_C4}}	{{L6_C5}}	{{L6_C6}}	{{L6_C7}}
<i>Pass reconh insufcréd/recursos</i>	{{L7_C1}}	{{L7_C2}}	{{L7_C3}}	{{L7_C4}}	{{L7_C5}}	{{L7_C6}}	{{L7_C7}}
<b>DEDUÇÕES (II)</b>	{{L8_C1}}	{{L8_C2}}	{{L8_C3}}	{{L8_C4}}	{{L8_C5}}	{{L8_C6}}	{{L8_C7}}
<i>Ativo Disponível</i>	{{L9_C1}}	{{L9_C2}}	{{L9_C3}}	{{L9_C4}}	{{L9_C5}}	{{L9_C6}}	{{L9_C7}}
<i>Haveres Financeiros</i>	{{L10_C1}}	{{L10_C2}}	{{L10_C3}}	{{L10_C4}}	{{L10_C5}}	{{L10_C6}}	{{L10_C7}}
<i>(-) Restos a Pagar Processados*</i>	{{L11_C1}}	{{L11_C2}}	{{L11_C3}}	{{L11_C4}}	{{L11_C5}}	{{L11_C6}}	{{L11_C7}}
<b>DIV CONS LIQ-DCL (III) = (I - II)</b>	{{L12_C1}}	{{L12_C2}}	{{L12_C3}}	{{L12_C4}}	{{L12_C5}}	{{L12_C6}}	{{L12_C7}}
<b>RECEITA CORR LIQUIDA - RCL</b>	{{L13_C1}}	{{L13_C2}}	{{L13_C3}}	{{L13_C4}}	{{L13_C5}}	{{L13_C6}}	{{L13_C7}}
<b>% da DC sobre a RCL</b>	{{L14_C1}}	{{L14_C2}}	{{L14_C3}}	{{L14_C4}}	{{L14_C5}}	{{L14_C6}}	{{L14_C7}}
<b>% da DCL sobre a RCL</b>	{{L15_C1}}	{{L15_C2}}	{{L15_C3}}	{{L15_C4}}	{{L15_C5}}	{{L15_C6}}	{{L15_C7}}
<b>% da DM* sobre a RCL (DM*/ RCL)</b>	{{L16_C1}}	{{L16_C2}}	{{L16_C3}}	{{L16_C4}}	{{L16_C5}}	{{L16_C6}}	{{L16_C7}}
Fonte: Relatório de Gestão Fiscal do Poder Executivo Federal do {{quadrimestre}}º quadrimestre de {{ano_atual}}.							
5. A Dívida Consolidada Líquida {{Z12_5}} {{Z12_3}} ({{Z12_4}}%) no {{quadrimestre}}º quadrimestre de {{ano_atual}} em relação ao quadrimestre anterior. Enquanto isso, a RCL da União, principal parâmetro de avaliação dos indicadores fiscais, passou de {{Z13_2}} para {{Z13_1}}, apresentando {{Z13_6}} {{Z13_4}}% no mesmo período, conforme dados consignados nos registros do presente RGF.							

Fonte: Elaboração própria a partir da estrutura utilizada pelo TCU nas instruções de fiscalização.

Figura 14 – Extrato do Template Base para o Demonstrativo das Garantias e Contragarantias de Valores

<b>Tabela xx – Demonstrativo das Garantias e Contragarantias de Valores</b>				
(LRF, arts. 40, § 1º, e 55, inciso I, alínea "c")				
R\$ milhares				
GARANTIAS CONCEDIDAS	{{ano_anti or}}	SALDOS DO EXERCÍCIO DE {{ano_atual}}		
		Até o 1º Quad	Até o 2º Quad	Até o 3º Quad
AOS ESTADOS (I)	{{L1_C1}}	{{L1_C2}}	{{L1_C3}}	{{L1_C4}}
AOS MUNICÍPIOS (II)	{{L2_C1}}	{{L2_C2}}	{{L2_C3}}	{{L2_C4}}
ÀS ENTIDADES CONTROLADAS (III)	{{L3_C1}}	{{L3_C2}}	{{L3_C3}}	{{L3_C4}}
POR MEIO DE FUNDOS E PROGRAMAS (IV)	{{L4_C1}}	{{L4_C2}}	{{L4_C3}}	{{L4_C4}}
TOTAL GARANTIAS CONCEDIDAS (V) = (I + II + III + IV)	{{L5_C1}}	{{L5_C2}}	{{L5_C3}}	{{L5_C4}}
RECEITA CORRENTE LÍQUIDA - RCL (VI)	{{L6_C1}}	{{L6_C2}}	{{L6_C3}}	{{L6_C4}}
% do TOTAL DAS GARANTIAS sobre a RCL	{{L7_C1}}	{{L7_C2}}	{{L7_C3}}	{{L7_C4}}
LIMITE DEFINIDO POR RESOLUÇÃO DO SENADO FEDERAL - 60%	{{L8_C1}}	{{L8_C2}}	{{L8_C3}}	{{L8_C4}}
LIMITE DE ALERTA (inc III do §1º do art. 59 da LRF) - (0,9 x 60%)	{{L9_C1}}	{{L9_C2}}	{{L9_C3}}	{{L9_C4}}
CONTRAGARANTIAS RECEBIDAS	{{ano_anti or}}	SALDOS DO EXERCÍCIO DE {{ano_atual}}		
		Até o 1º Quad	Até o 2º Quad	Até o 3º Quad
DOS ESTADOS (VII)	{{L10_C1}}	{{L10_C2}}	{{L10_C3}}	{{L10_C4}}
DOS MUNICÍPIOS (VIII)	{{L11_C1}}	{{L11_C2}}	{{L11_C3}}	{{L11_C4}}
DAS ENTIDADES CONTROLADAS (IX)	{{L12_C1}}	{{L12_C2}}	{{L12_C3}}	{{L12_C4}}
EM GARANTIAS POR MEIO DE FUNDOS E PROGRAMAS (X)	{{L13_C1}}	{{L13_C2}}	{{L13_C3}}	{{L13_C4}}
TOTAL CONTRAGARANTIAS RECEBIDAS (XI) = (VII + VIII + IX + X)	{{L14_C1}}	{{L14_C2}}	{{L14_C3}}	{{L14_C4}}

Fonte: RGF do Poder Executivo Federal do {{quadrimestre}}º quadrimestre de {{ano\_atual}}.

7. No período objeto desta análise, o saldo das garantias concedidas pela União aos estados, municípios, estatais federais e entidades controladas e por meio de fundos e programas passou de {{Z5\_2}} no quadrimestre anterior para {{Z5\_1}} no {{quadrimestre}}º quadrimestre de {{ano\_atual}}, o que representa {{Z5\_6}} de {{Z5\_3}} (ou {{Z5\_4}}%). {{Z\_garant\_conclusao}}

8. Já o saldo das contragarantias totalizou {{Z14\_1}} ({{Z\_aux1}}% da RCL), o que representa {{Z\_aux2}}% das garantias concedidas até o quadrimestre em análise.

Fonte: Elaboração própria a partir da estrutura utilizada pelo TCU nas instruções de fiscalização.

Figura 15 – Extrato do Template Base para o Demonstrativo das Operações de Crédito

<b>Tabela xx – Demonstrativo das Operações de Crédito</b> (LRF, art. 55, inciso I, alínea "d")		
		R\$ milhares
OPERAÇÕES DE CRÉDITO	VALOR	
	No {{quadrimestre}}° quadrimestre	Até o {{quadrimestre}}° quadrimestre
OPERAÇÕES DE CRÉDITO (I)	{{L1_C1}}	{{L1_C2}}
Mobiliária	{{L2_C1}}	{{L2_C2}}
Interna	{{L3_C1}}	{{L3_C2}}
Externa	{{L4_C1}}	{{L4_C2}}
Contratual	{{L5_C1}}	{{L5_C2}}
Externa	{{L6_C1}}	{{L6_C2}}
APURAÇÃO DO CUMPRIMENTO DOS LIMITES	VALOR	% SOBRE A RCL
RECEITA CORRENTE LÍQUIDA	{{L7C1}}	-
OPERAÇÕES VEDADAS (II)	{{L8C1}}	{{L8C2}}
OUTRAS OPERAÇÕES DEDUZIDAS DO LIMITE (III)	{{L9C1}}	{{L9C2}}
Amortização/Refinanciamento do Principal de Dívidas	{{L10C1}}	{{L10C2}}
Cancelamento de títulos aceitos em leilões de trocas	{{L11C1}}	{{L11C2}}
Aporte Bacen Lei 11.803/2008 <sup>1</sup>	{{L12C1}}	{{L12C2}}
Concessão de Garantias <sup>2</sup>	{{L13C1}}	{{L13C2}}
TOTAL CONSIDERADO P/APURAÇÃO DO CUMPR DO LIMITE (IV) = (Ia) + (II) – (III)	{{L14C1}}	{{L14C2}}
LIMITE GERAL DEF P/RESOL DO SENADO P/AS OPER DE CRÊD EXT E INT	{{L15C1}}	60,00%

Fonte: RGF do Poder Executivo Federal do {{quadrimestre}}° quadrimestre de {{ano\_atual}}.

5. Conforme evidenciado na Tabela xx, o valor bruto das operações de crédito contratadas ficou em {{Z1\_1}}. As do tipo mobiliária totalizaram {{Z2\_1}} ({{Z3\_1}} interna e {{Z4\_1}} externa), enquanto as contratuais somaram {{Z5\_1}}.

6. Foram deduzidos valores no total de {{Z9\_1}} a título de amortização e refinanciamento da dívida ({{Z10\_1}}) e de aporte do Bacen ({{Z12\_1}}).

7. {{Z\_opcred conclusao}}

Fonte: Elaboração própria a partir da estrutura utilizada pelo TCU nas instruções de fiscalização.

Segue exemplo do código utilizado para elaboração do parágrafo de conclusão e encaminhamento (exemplo refere-se ao demonstrativo das Garantias e Contragarantias de Valores):

Figura 16 – Exemplo de Conclusão e Encaminhamento da Minuta de Relatório Assistido

```

### Visualização do % do total das garantias em relação à RCL
df_assistido.iloc[6][0]

34.505316868552676

### Texto para conclusão da análise, a partir do resultado do resultado acima (Z7_1, ou df_assistido.iloc[6][0])
if df_assistido.iloc[6][0] < (60 *0.9):
    Z_garant_conclusao = str('0 percentual das Garantias em relação à RCL foi de '+str(template_dict_z['Z7_1'])+', portanto, inferior a 90% do limite de referência, que é 60% da RCL.')
else:
    if df_assistido.iloc[6][0] <= 60:
        Z_garant_conclusao = str('0 percentual da DCL em relação à RCL foi de '+str(template_dict_z['Z7_1'])+', portanto, ficou entre 90 e 100% do limite de referência. Nessa situação, em atenção ao disposto no art. 59, § 1º da LRF, propõe-se a emissão de alerta ao Poder Executivo, registrando que, nos termos do art. 40, § 5º da LRF, é nula a garantia concedida acima dos limites fixados pelo Senado Federal.')
    else:
        Z_garant_conclusao = str('0 percentual da DCL em relação à RCL foi de '+str(template_dict_z['Z7_1'])+', portanto, superior ao limite de referência. Em virtude disso, propõe-se determinar ao Poder Executivo a adoção de medidas com vistas a adequação ao disposto no art. 40, § 5º da LRF, que diz que é nula a garantia concedida acima dos limites fixados pelo Senado Federal.')

Z_garant_conclusao

'0 percentual das Garantias em relação à RCL foi de 34,51, portanto, inferior a 90% do limite de referência, que é 60% da RCL.'

```

Fonte: Elaboração própria.

## 7 AVALIAÇÃO

Verificou-se que a solução desenvolvida neste trabalho logrou êxito em realizar, de forma automática, todas as consultas necessárias no banco de dados DWTG, por meio da ferramenta LabContas, para conferência dos anexos dos demonstrativos da Dívida Consolidada Líquida, das Garantias e Contragarantias de Valores e das Operações de Crédito do Relatório de Gestão Fiscal do Poder Executivo federal.

Após executar as linhas de programação referentes a cada anexo do RGF, as consultas foram realizadas para os períodos determinados, os dados brutos obtidos foram tratados e analisados de acordo com a metodologia constante no Manual de Demonstrativos Fiscais.

Os produtos foram gerados adequadamente para os demonstrativos da Dívida Consolidada Líquida, das Garantias e Contragarantias e das Operações de Crédito, com as tabelas devidamente preenchidas no formato e nas condições utilizadas nos relatórios elaborados pela equipe de fiscalização do TCU.

Os dados gerados a partir das informações de cada demonstrativo foram utilizados para alimentar os parágrafos da minuta de relatório assistido de cada tópico, de modo que as principais conclusões das análises são geradas automaticamente em tabela e em texto.

Seguem extratos dos arquivos gerados após a execução dos códigos referentes a cada demonstrativo no 3º quadrimestre de 2019:

Figura 17 – Extrato de Minuta de Relatório Assistido do Demonstrativo da DCL

<b>Tabela xx – Demonstrativo da Dívida Consolidada Líquida</b>							
(LRF, art. 55, inciso I, alínea "b")							
R\$ milhares							
ESPECIFICAÇÃO	SALDO 2018.	Até o 1º Quadr	Var Perc	Até o 2º Quadr	Var Perc	Até o 3º Quadr	Var Perc
<b>DÍVIDA CONSOLIDADA - DC (I)</b>	5.764.560.886	5.781.968.233	0,30	5.943.381.534	2,79	6.200.607.290	4,33
<i>Dívida Mobiliária do Tesouro Nacional</i>	5.684.014.621	5.735.956.128	0,91	5.893.481.036	2,75	6.146.316.245	4,29
<i>Oper de Eq Camb - Rel TN/BCB</i>	27.186.388	1.049.927	-96,14	5.618.827	435,16	6.705.687	19,34
<i>Dívida Contratual</i>	44.552.710	38.409.100	-13,79	38.450.968	0,11	40.492.803	5,31
<i>Precatórios posteriores a 5/5/2000</i>	411.567	2.614.105	535,16	1.963.128	-24,90	557.030	-71,63
<i>Dív Assumida pela União (Lei 8.727/93)</i>	8.395.601	3.938.972	-53,08	3.867.575	-1,81	6.535.525	68,98
<i>Pass reconh insuf créd/recursos</i>	0	0	0,00	0	0,00	0	0,00
<b>DEDUÇÕES (II)</b>	2.579.392.494	2.470.973.830	-4,20	2.456.118.483	-0,60	2.616.523.454	6,53
<i>Ativo Disponível</i>	1.276.823.195	1.173.014.572	-8,13	1.217.631.121	3,80	1.438.785.413	18,16
<i>Haveres Financeiros</i>	1.363.020.187	1.363.678.504	0,05	1.324.122.277	-2,90	1.251.837.095	-5,46
<i>(-) Restos a Pagar Processados*</i>	-60.450.888	-65.719.246	8,72	-85.634.915	30,30	-74.099.054	-13,47
<b>DÍV CONS LIQ-DCL (III) = (I - II)</b>	3.185.168.392	3.310.994.402	3,95	3.487.263.051	5,32	3.584.083.837	2,78
<b>RECEITA CORR LIQUIDA - RCL</b>	805.348.403	818.616.620	1,65	829.157.292	1,29	905.658.590	9,23
<b>% da DC sobre a RCL</b>	716	706	-1,32	717	1,48	685	-4,48
<b>% da DCL sobre a RCL</b>	396	404	2,27	421	3,98	396	-5,91
<b>% da DM* sobre a RCL (DM*/ RCL)</b>	706	701	-0,72	711	1,44	679	-4,52

Fonte: Relatório de Gestão Fiscal do Poder Executivo Federal do 3º quadrimestre de 2019.

5. A Dívida Consolidada Líquida aumentou R\$ 96,82 bilhões (2,78%) no 3º quadrimestre de 2019 em relação ao quadrimestre anterior. Enquanto isso, a RCL da União, principal parâmetro de avaliação dos indicadores fiscais, passou de R\$ 829,16 bilhões para R\$ 905,66 bilhões, apresentando crescimento de 9,23% no mesmo período, conforme dados consignados nos registros do presente RGF.

Fonte: Elaboração própria.

Figura 18 – Extrato de Minuta de Relatório Assistido do Demonstrativo das Garantias e Contragarantias de Valores

**Tabela xx – Demonstrativo das Garantias e Contragarantias de Valores**

(LRF, arts. 40, § 1º, e 55, inciso I, alínea "c")

GARANTIAS CONCEDIDAS	2018	R\$ milhares SALDOS DO EXERCÍCIO DE 2019		
		Até o 1º Quad	Até o 2º Quad	Até o 3º Quad
		AOS ESTADOS (I)	197.186.182	197.113.572
AOS MUNICÍPIOS (II)	16.739.529	17.279.045	18.215.762	19.455.577
ÀS ENTIDADES CONTROLADAS (III)	41.099.979	40.600.834	40.137.918	35.912.373
POR MEIO DE FUNDOS E PROGRAMAS (IV)	22.862.328	22.902.323	22.893.464	23.674.573
<b>TOTAL GARANTIAS CONCEDIDAS (V) = (I + II + III + IV)</b>	<b>277.888.019</b>	<b>277.895.773</b>	<b>284.429.879</b>	<b>279.506.012</b>
RECEITA CORRENTE LÍQUIDA - RCL (VI)	805.348.403	818.616.620	829.157.292	905.658.590
% do TOTAL DAS GARANTIAS sobre a RCL	34,51	33,95	34,30	30,86
LIMITE DEFINIDO POR RESOLUÇÃO DO SENADO FEDERAL - 60%	483.209.042	491.169.972	497.494.375	543.395.154
LIMITE DE ALERTA (inc III do §1º do art. 59 da LRF) - (0,9 x 60%)	434.888.138	442.052.975	447.744.938	489.055.638
CONTRAGARANTIAS RECEBIDAS	2018	SALDOS DO EXERCÍCIO DE 2019		
		Até o 1º Quad	Até o 2º Quad	Até o 3º Quad
		DOS ESTADOS (VII)	197.186.182	199.093.740
DOS MUNICÍPIOS (VIII)	16.739.529	17.332.999	18.215.762	19.455.577
DAS ENTIDADES CONTROLADAS (IX)	16.381.671	16.801.162	16.122.785	15.550.349
EM GARANTIAS POR MEIO DE FUNDOS E PROGRAMAS (X)	0	0	0	0
<b>TOTAL CONTRAGARANTIAS RECEBIDAS (XI) = (VII + VIII + IX + X)</b>	<b>230.307.382</b>	<b>233.227.901</b>	<b>237.521.282</b>	<b>235.469.416</b>

Fonte: RGF do Poder Executivo Federal do 3º quadrimestre de 2019.

7. No período objeto desta análise, o saldo das garantias concedidas pela União aos estados, municípios, estatais federais e entidades controladas e por meio de fundos e programas passou de R\$ 284,43 bilhões no quadrimestre anterior para R\$ 279,51 bilhões no 3º quadrimestre de 2019, o que representa redução de de R\$ 4,92 bilhões (ou -1,73%). O percentual das Garantias em relação à RCL foi de 30,86, portanto, inferior a 90% do limite de referência, que é 60% da RCL.

8. Já o saldo das contragarantias totalizou R\$ 235,47 bilhões (26,00% da RCL), o que representa 84,24% das garantias concedidas até o quadrimestre em análise.

Fonte: elaboração própria.



Figura 199 – Extrato de Minuta de Relatório Assistido do Demonstrativo das Operações de Crédito

<b>Tabela xx – Demonstrativo das Operações de Crédito</b> (LRF, art. 55, inciso I, alínea "d")		
R\$ milhares		
OPERAÇÕES DE CRÉDITO	VALOR	
	No 3º quadrimestre	Até o 3º quadrimestre
OPERAÇÕES DE CRÉDITO (I)	285.595.248	979.205.026
Mobiliária	283.669.041	976.645.259
Interna	270.966.710	957.988.228
Externa	12.702.331	18.657.031
Contratual	1.926.207	2.559.767
Externa	1.926.207	2.559.767
APURAÇÃO DO CUMPRIMENTO DOS LIMITES	VALOR	% SOBRE A RCL
RECEITA CORRENTE LÍQUIDA	905.658.590	-
OPERAÇÕES VEDADAS (II)	0	0,00
OUTRAS OPERAÇÕES DEDUZIDAS DO LIMITE (III)	753.556.725	83,21
Amortização/Refinanciamento do Principal de Dívidas	752.534.506	83,09
Cancelamento de títulos aceitos em leilões de trocas	0	0,00
Aporte Bacen Lei 11.803/2008 <sup>1</sup>	1.022.219	0,11
Concessão de Garantias <sup>2</sup>	0	0,00
TOTAL CONSIDERADO P/APURAÇÃO DO CUMPR DO LIMITE (IV) = (Ia) + (II) – (III)	225.648.300	24,92
LIMITE GERAL DEF P/RESOL DO SENADO P/AS OPER DE CRÉD EXT E INT	543.395.154	60,00%

Fonte: RGF do Poder Executivo Federal do 3º quadrimestre de 2019.

5. Conforme evidenciado na Tabela xx, o valor bruto das operações de crédito contratadas ficou em R\$ 979,21 bilhões. As do tipo mobiliária totalizaram R\$ 976,65 bilhões (R\$ 957,99 bilhões interna e R\$ 18,66 bilhões externa), enquanto as contratuais somaram R\$ 2,56 bilhões.

6. Foram deduzidos valores no total de R\$ 753,56 bilhões a título de amortização e refinanciamento da dívida (R\$ 752,53 bilhões) e de aporte do Bacen (R\$ 1,02 bilhão).

7. O percentual das Operações de Crédito em relação à RCL foi de 24,92, portanto, inferior a 90% do limite de referência, que é 60% da RCL.

Fonte: elaboração própria.

Nas simulações realizadas conforme figura 16 deste trabalho, as conclusões e encaminhamentos possíveis para cada demonstrativo, quais sejam, cumprimento de limite (menor que 90% do limite), enquadramento no limite de alerta (entre 90 e 100%) e ultrapassagem do limite (maior que 100%) foram inseridas adequadamente em cada minuta de relatório assistido.

Além disso, os arquivos em formato de planilha eletrônica “xlsx”, gerados e salvos automaticamente no formato de cada demonstrativo, permitem eventuais análises adicionais que se mostrarem necessárias, a depender de especificidades de cada período.

Referidos produtos têm o potencial de permitir a otimização dos trabalhos do TCU referentes ao acompanhamento quadrimestral do Relatório de Gestão Fiscal, reduzindo a força de trabalho necessária para realizar parte das tarefas, permitindo que o Tribunal possa direcionar a equipe de auditores para atividades que efetivamente demandem o conhecimento especializado dos servidores.

## **8 CONCLUSÃO**

Uma das principais competências do Tribunal de Contas da União (TCU) é a fiscalização das normas da Lei de Responsabilidade Fiscal (LRF) atinentes à responsabilidade da gestão fiscal.

Nesse sentido, cabe ao TCU acompanhar o cumprimento dos limites referentes aos demonstrativos constantes do Relatório de Gestão Fiscal (RGF) do Poder Executivo federal.

A Secretaria de Macroavaliação Governamental (Semag) tem como uma de suas atribuições realizar o acompanhamento periódico e contínuo das publicações quadrimestrais referentes ao RGF.

Para tanto, a Semag tem em seu planejamento anual ações de fiscalização que possuem como objeto as publicações referentes aos RGFs quadrimestrais. Parte da força de trabalho da equipe de auditores consiste em realizar extrações no sistema TesouroGerencial, as quais geram dados de saída bruto, que são tratados de modo que resultem nas informações que alimentam os campos de cada demonstrativo do RGF.

Por meio da solução ora desenvolvida, buscou-se automatizar as etapas de extração, tratamento, organização e apresentação das informações constantes de cada demonstrativo, com a finalidade de otimizar a ação de fiscalização do TCU, reduzindo a força de trabalho necessária para as tarefas atualmente realizadas e permitindo que a equipe possa se dedicar a desempenhar análises que efetivamente exigem a atuação do auditor federal de controle externo, no que diz respeito à sua expertise.

A elaboração da presente solução teve início com o levantamento do fluxo de trabalho desenvolvido pela equipe de fiscalização da Semag, estudo dos normativos e documentos que regulam o Relatório de Gestão Fiscal, e posterior mapeamento das etapas necessárias para automatizar as fases de extração, tratamento, organização e apresentação dos demonstrativos da

Dívida Consolidada Líquida, das Garantias e Contragarantias de Valores, e das Operações de Crédito.

Em seguida, todas as consultas ao sistema TesouroGerencial necessárias à elaboração dos demonstrativos foram obtidas na forma de Linguagem de Consulta Estruturada (SQL), por meio do qual o acesso ao banco de dados DWTG pode ser realizado via plataforma do TCU LabContas.

De posse dos dados brutos extraídos referentes à cada período, desenvolveu-se a estrutura de tratamento das informações para se chegar a cada campo de cada demonstrativo, etapa que envolve a seleção, limpeza, construção e integração de dados.

O resultado parcial foi a obtenção de dataframes referentes à cada demonstrativo, os quais foram utilizados como subsídio para obtenção dos produtos finais da presente solução.

O primeiro produto consiste na tabela utilizada pela Semag nas instruções de acompanhamento do RGF. Por meio do pacote docx-template, disponível na linguagem de programação python, desenvolveu-se código que, uma vez executado, geram-se tabelas no formato desejado devidamente preenchida, com a análise de cumprimento aos limites estabelecidos pela LRF.

As tabelas são complementadas por parágrafos que buscam oferecer à equipe de fiscalização minuta de instrução referente à cada demonstrativo, na qual os principais pontos da análise são abordados de forma automática.

Outro produto refere-se à exportação de tabelas em formato “xlsx” referentes à cada demonstrativo, disponível de acordo com o quadrimestre analisado.

Ao final, a automatização das etapas de extração, tratamento, organização e apresentação dos demonstrativos da Dívida Consolidada Líquida, das Garantias e Contragarantias de Valores e das Operações de crédito mostrou-se bem-sucedida.

### **8.1 Trabalhos Futuros**

O processo de automatização realizado neste trabalho pode ser replicado para outras ações de fiscalização realizadas pela Secretaria de Macroavaliação Governamental.

A estrutura construída é passível de replicação nos casos em que, de forma periódica, a Semag realize consultas a sistemas de informações relacionadas a finanças públicas.

Como exemplo de fiscalizações que podem se valer da lógica de automatização ora desenvolvida, cita-se o acompanhamento bimestralmente realizado sobre as receitas primárias, despesas primárias impactantes, resultado primário e contingenciamento.

Há tópicos elaborados no Relatório e Parecer Prévio sobre as Contas do Presidente que podem igualmente ter parte da tarefa automatizada, não para substituir a análise do auditor, mas para otimizar a extração, tratamento, organização e apresentação dos dados. Os tópicos referentes à receita orçamentária e à despesa orçamentária, que possuem caráter de contextualização por sua natureza descritiva, são exemplos de possibilidades futuras de aproveitamento de parte relevante da estrutura aqui desenvolvida.

Outro trabalho que pode ser objeto de automatização em condições semelhantes ao ora desenvolvido é a análise anualmente realizada sobre a regularidade da abertura de créditos suplementares em relação ao art. 4º da Lei Orçamentária Anual (LOA). Para tanto, faz-se necessário que as bases de dados referentes ao Sistema Integrado de Planejamento e Orçamento (Siop) estejam disponíveis na plataforma LabContas do TCU.

## REFERÊNCIAS

- . **Brasil. 1988.** Constituição Federal de 1988. *Constituição da República Federativa do Brasil*. 1988.
- . **2000.** Lei Complementar nº 101 de 4 de maio de 20. *Lei de Responsabilidade Fiscal*. 04 de maio de 2000.
- . **1992.** Lei nº 8.443/1992. *Lei Orgânica do Tribunal de Contas da União*. 16 de julho de 1992.
- União, Tribunal de Contas da.** Resolução-TCU nº 155/2002. *Aprova o Regimento Interno do Tribunal de Contas da União*. 04 de dezembro de 2002.
- União, Tribunal de Contas da.** Resolução-TCU nº 142/2001. *Dispõe sobre as competências atribuídas ao Tribunal de Contas da União pela Lei Complementar nº 101, de 4 de maio de 2000, denominada Lei de Responsabilidade Fiscal - LRF*. 30 de maio de 2001
- União, Tribunal de Contas da.** Resolução-TCU nº 305/2018. *Define a estrutura, as competências e a distribuição das funções de confiança das unidades da Secretaria do Tribunal de Contas da União*. de 28/12/2018. 28 de dezembro de 2018.
- União, Tribunal de Contas da.** Portaria-Segecex nº 5/2017. *Dispõe sobre as competências, a estrutura e a alocação de funções de confiança da Secretaria-Geral de Controle Externo*. 07 de março de 2017.
- Nacional, Secretaria do Tesouro, 2020.** Manual de Demonstrativos Fiscais.
- Nacional, Secretaria do Tesouro. 2019.** Manual de Contabilidade Aplicada ao Setor Público.
- Kuznetsov, Alex, 2010.** Defensive Database Programming with SQL Server. Disponível em <http://assets.red-gate.com/community/books/defensive-database-programming.pdf>. Acesso em março de 2020.
- Pete Chapman, Julian Clinton, Randy Kerber, Thomas Khabaza, Thomas Reinartz, Colin Shearer e Rüdiger Wirth. 2000.** CRISP-DM 1.0 Step-by-step data mining guide. Disponível em <ftp://ftp.software.ibm.com/software/analytics/spss/support/Modeler/Documentation/14/UserManual/CRISP-DM.pdf>. Acesso em março de 2020.
- AGARWAL, Sundeep. 2016.** Python Basics - Introduction to Python - Syntax, working with Shell commands, Files, Text Processing, and more. Disponível em [https://github.com/learnbyexample/Python\\_Basics?utm\\_source=devfreebooks&utm\\_medium=medium&utm\\_campaign=DevFreeBooks](https://github.com/learnbyexample/Python_Basics?utm_source=devfreebooks&utm_medium=medium&utm_campaign=DevFreeBooks). Acesso em março de 2020.

In [ ]:

```
ANEXO I - DÍVIDA CONSOLIDADA LÍQUIDA
```

In [443]:

```
### Importação de bibliotecas e definições iniciais
import docxtp1
import pandas as pd
import pyodbc
import locale

conn = pyodbc.connect('DSN=LABCONTAS')
quad_to_mes = {1:4,2:8,3:12}
locale.setlocale( locale.LC_MONETARY , 'pt_BR.UTF-8' )
'{} {}'.format('one', 'two')
```

Out[443]:

```
'one two'
```

In [444]:

```
### DEFINIÇÃO DO PERÍODO OBJETO DA ANÁLISE --- ALTERE AQUI!!!
ano_atual = 2019
quadrimestre = 3

### Definições fixas
ano_anterior = ano_atual - 1
quad1 = 1
quad2 = 2
quad3 = 3
```

In [445]:

### *Queries de acordo com as consultas do sistema TesouroGerencial convertidas em SQL*

###L2+ *Dívida Mobiliária TN Interna*

```
query1 = ""select          sum((a11.VA_MOVIMENTO_LIQUIDO * a13.PE_TAXA))  SDCC
from    BD_DWTG.DBO.WF_LANCAMENTO_EP20  a11
        join    BD_DWTG.DBO.WD_MOEDA      a12
            on   (a11.ID_MOEDA_UG_EXEC_H = a12.ID_MOEDA)
        join    BD_DWTG.DBO.WD_TAXA_CAMBIO_MENSAL  a13
            on   (a12.ID_MOEDA = a13.ID_MOEDA_ORIGEM)
        join    BD_DWTG.DBO.WA_MES_ACUM  a14
            on   (a11.ID_ANO_LANC = a14.ID_ANO_ACUM_ANO_SALDO and
a11.ID_MES_LANC = a14.ID_MES_ACUM_ANO_SALDO)
        join    BD_DWTG.DBO.WD_CONTA_CONTABIL_EXERCICIO  a15
            on   (a11.ID_ANO_LANC = a15.ID_ANO and
a11.ID_CONTA_CONTABIL_LANC = a15.ID_CONTA_CONTABIL)
        join    BD_DWTG.DBO.WD_MES      a16
            on   (a14.ID_ANO = a16.ID_ANO and
a14.ID_MES = a16.ID_MES)
where    (a11.ID_TP_ENTIDADE_CCOR = 'IN'
and a11.ID_ENTIDADE_CCOR in ('DP1000001', 'DP1400001', 'DP1500001', 'DP1700001', 'DP18
00001', 'DP2000001', 'DP2300007', 'DP2400001', 'DP2600001', 'DP2800001', 'DP3000001',
'DP3400001', 'DP5000001', 'DP5500001', 'DP5800001', 'DP6100001', 'DP6200001', 'DP63000
01', 'DP6600001', 'DP7000001', 'DP8000001', 'DP9000001', 'DP1200001')
and a11.ID_ORCA_FISCAL_ORGAO_EXEC_H in (0)
and (a14.ID_ANO = {} and a14.ID_MES = {}))
and a15.ID_CONTA_CONTABIL_DESTINO in (899913900, 899913901, 899913902, 899913903, 8999
13904, 899913905, 899913906)
and a13.ID_ANO = a16.ID_ANO
and a13.ID_MES = a16.ID_MES)""
```

### *and cast(a19.ID\_CONTA\_CONTABIL as varchar) like '1111150%' tiramos o coringa do fina  
L*

###L2- *Aplicação em Títulos Públicos*

```
query2 = ""select          sum((a11.VA_MOVIMENTO_LIQUIDO * a13.PE_TAXA))  SDCC
from    BD_DWTG.DBO.WF_LANCAMENTO_EP02  a11
        join    BD_DWTG.DBO.WD_MOEDA      a12
            on   (a11.ID_MOEDA_UG_EXEC_H = a12.ID_MOEDA)
        join    BD_DWTG.DBO.WD_TAXA_CAMBIO_MENSAL  a13
            on   (a12.ID_MOEDA = a13.ID_MOEDA_ORIGEM)
        join    BD_DWTG.DBO.WA_MES_ACUM  a14
            on   (a11.ID_ANO_LANC = a14.ID_ANO_ACUM_ANO_SALDO and
a11.ID_MES_LANC = a14.ID_MES_ACUM_ANO_SALDO)
        join    BD_DWTG.DBO.WD_CONTA_CONTABIL_EXERCICIO  a15
            on   (a11.ID_ANO_LANC = a15.ID_ANO and
a11.ID_CONTA_CONTABIL_LANC = a15.ID_CONTA_CONTABIL)
        join    BD_DWTG.DBO.WD_UG_EXERCICIO      a16
            on   (a11.ID_ANO_LANC = a16.ID_ANO and
a11.ID_UG_EXEC = a16.ID_UG)
        join    BD_DWTG.DBO.WD_MES      a17
            on   (a14.ID_ANO = a17.ID_ANO and
a14.ID_MES = a17.ID_MES)
        join    BD_DWTG.DBO.WD_ORGAO      a18
            on   (a16.ID_ORGAO_UG = a18.ID_ORGAO)
        join    BD_DWTG.DBO.WD_CONTA_CONTABIL  a19
            on   (a15.ID_CONTA_CONTABIL_DESTINO = a19.ID_CONTA_CONTABIL)
where    (a11.ID_ORCA_FISCAL_ORGAO_EXEC_H in (0)
and (a14.ID_ANO = {} and a14.ID_MES = {}))
and a18.ID_TP_ADMIN_ORGAO in (3, 4, 5, 6, 8)
```

```

and cast(a19.ID_CONTA_CONTABIL as varchar) like '1111150%'
and a15.ID_CONTA_CONTABIL_DESTINO not in (111115005, 111115011, 111115012)
and a16.ID_ORGAO_UG not in (25901)
and a13.ID_ANO = a17.ID_ANO
and a13.ID_MES = a17.ID_MES)""

```

### L2+ Dívida Mobiliária do TN (em BCB)

```

query3 = ""select          sum((a11.VA_MOVIMENTO_LIQUIDO * a13.PE_TAXA))  SDCC
from    BD_DWTG.DBO.WF_LANCAMENTO_EP20  a11
        join    BD_DWTG.DBO.WD_MOEDA      a12
              on    (a11.ID_MOEDA_UG_EXEC_H = a12.ID_MOEDA)
        join    BD_DWTG.DBO.WD_TAXA_CAMBIO_MENSAL      a13
              on    (a12.ID_MOEDA = a13.ID_MOEDA_ORIGEM)
        join    BD_DWTG.DBO.WA_MES_ACUM  a14
              on    (a11.ID_ANO_LANC = a14.ID_ANO_ACUM_ANO_SALDO and
a11.ID_MES_LANC = a14.ID_MES_ACUM_ANO_SALDO)
        join    BD_DWTG.DBO.WD_CONTA_CONTABIL_EXERCICIO a15
              on    (a11.ID_ANO_LANC = a15.ID_ANO and
a11.ID_CONTA_CONTABIL_LANC = a15.ID_CONTA_CONTABIL)
        join    BD_DWTG.DBO.WD_MES      a16
              on    (a14.ID_ANO = a16.ID_ANO and
a14.ID_MES = a16.ID_MES)
where    (a11.ID_ORCA_FISCAL_ORGAO_EXEC_H in (0)
and (a14.ID_ANO = {} and a14.ID_MES = {}))
and a15.ID_CONTA_CONTABIL_DESTINO in (899913901, 899913902, 899913907, 899913908)
and a11.ID_ENTIDADE_CCOR in ('DP1500010', 'DP1700010', 'DP1800010', 'DP2300010', 'DP55
00010', 'DP7000010', 'DP9000010')
and a13.ID_ANO = a16.ID_ANO
and a13.ID_MES = a16.ID_MES)""

```

### L2+ Dívida Securitizada

```

query4 = ""select          sum((a11.VA_MOVIMENTO_LIQUIDO * a13.PE_TAXA))  SDCC
from    BD_DWTG.DBO.WF_LANCAMENTO_EP20  a11
        join    BD_DWTG.DBO.WD_MOEDA      a12
              on    (a11.ID_MOEDA_UG_EXEC_H = a12.ID_MOEDA)
        join    BD_DWTG.DBO.WD_TAXA_CAMBIO_MENSAL      a13
              on    (a12.ID_MOEDA = a13.ID_MOEDA_ORIGEM)
        join    BD_DWTG.DBO.WA_MES_ACUM  a14
              on    (a11.ID_ANO_LANC = a14.ID_ANO_ACUM_ANO_SALDO and
a11.ID_MES_LANC = a14.ID_MES_ACUM_ANO_SALDO)
        join    BD_DWTG.DBO.WD_CONTA_CONTABIL_EXERCICIO a15
              on    (a11.ID_ANO_LANC = a15.ID_ANO and
a11.ID_CONTA_CONTABIL_LANC = a15.ID_CONTA_CONTABIL)
        join    BD_DWTG.DBO.WD_MES      a16
              on    (a14.ID_ANO = a16.ID_ANO and
a14.ID_MES = a16.ID_MES)
where    (a11.ID_ORCA_FISCAL_ORGAO_EXEC_H in (0)
and (a14.ID_ANO = {} and a14.ID_MES = {}))
and a13.ID_ANO = a16.ID_ANO
and a13.ID_MES = a16.ID_MES
and ((a15.ID_CONTA_CONTABIL_DESTINO in (899913900, 899913901, 899913902, 899913903, 89
9913904, 899913905, 899913906)
and a11.ID_ENTIDADE_CCOR in ('DP3100001', 'DP3200001', 'DP3200002', 'DP3201031', 'DP32
01032', 'DP3201059', 'DP3201077', 'DP3201078', 'DP3201080', 'DP3201081', 'DP3201145',
'DP3201202', 'DP3201222', 'DP3201228', 'DP3201233', 'DP3201250', 'DP3201256', 'DP32012
57', 'DP3201258', 'DP3201259', 'DP3201260', 'DP3201262', 'DP3201271', 'DP3201272', 'DP3
201275', 'DP3201276', 'DP3201277', 'DP3201280', 'DP3201281', 'DP3201296', 'DP3201299',

```



```
'DP3201362', 'DP3201368', 'DP3201378', 'DP3201390'))  
or a15.ID_CONTA_CONTABIL_DESTINO in (212110202, 222110102))''''
```

### ###L2+ Dívida Mobiliária Externa

```
query5 = ""select          sum((a11.VA_MOVIMENTO_LIQUIDO * a13.PE_TAXA))  SDCC  
from      BD_DWTG.DBO.WF_LANCAMENTO_EP02  a11  
          join      BD_DWTG.DBO.WD_MOEDA    a12  
                on  (a11.ID_MOEDA_UG_EXEC_H = a12.ID_MOEDA)  
          join      BD_DWTG.DBO.WD_TAXA_CAMBIO_MENSAL      a13  
                on  (a12.ID_MOEDA = a13.ID_MOEDA_ORIGEM)  
          join      BD_DWTG.DBO.WA_MES_ACUM  a14  
                on  (a11.ID_ANO_LANC = a14.ID_ANO_ACUM_ANO_SALDO and  
a11.ID_MES_LANC = a14.ID_MES_ACUM_ANO_SALDO)  
          join      BD_DWTG.DBO.WD_CONTA_CONTABIL_EXERCICIO a15  
                on  (a11.ID_ANO_LANC = a15.ID_ANO and  
a11.ID_CONTA_CONTABIL_LANC = a15.ID_CONTA_CONTABIL)  
          join      BD_DWTG.DBO.WD_MES      a16  
                on  (a14.ID_ANO = a16.ID_ANO and  
a14.ID_MES = a16.ID_MES)  
where      (a11.ID_ORCA_FISCAL_ORGAO_EXEC_H in (0)  
and (a14.ID_ANO = {} and a14.ID_MES = {})  
and a13.ID_ANO = a16.ID_ANO  
and a13.ID_MES = a16.ID_MES  
and a15.ID_CONTA_CONTABIL_DESTINO in (899913903, 899913904))''''
```

### ###L3+ Operações de Equalização Cambial

```
query6 = ""select          sum((a11.VA_MOVIMENTO_LIQUIDO * a13.PE_TAXA))  SDCC  
from      BD_DWTG.DBO.WF_LANCAMENTO_EP02  a11  
          join      BD_DWTG.DBO.WD_MOEDA    a12  
                on  (a11.ID_MOEDA_UG_EXEC_H = a12.ID_MOEDA)  
          join      BD_DWTG.DBO.WD_TAXA_CAMBIO_MENSAL      a13  
                on  (a12.ID_MOEDA = a13.ID_MOEDA_ORIGEM)  
          join      BD_DWTG.DBO.WA_MES_ACUM  a14  
                on  (a11.ID_ANO_LANC = a14.ID_ANO_ACUM_ANO_SALDO and  
a11.ID_MES_LANC = a14.ID_MES_ACUM_ANO_SALDO)  
          join      BD_DWTG.DBO.WD_CONTA_CONTABIL_EXERCICIO a15  
                on  (a11.ID_ANO_LANC = a15.ID_ANO and  
a11.ID_CONTA_CONTABIL_LANC = a15.ID_CONTA_CONTABIL)  
          join      BD_DWTG.DBO.WD_MES      a16  
                on  (a14.ID_ANO = a16.ID_ANO and  
a14.ID_MES = a16.ID_MES)  
where      (a11.ID_ORCA_FISCAL_ORGAO_EXEC_H in (0)  
and (a14.ID_ANO = {} and a14.ID_MES = {})  
and a13.ID_ANO = a16.ID_ANO  
and a13.ID_MES = a16.ID_MES  
and a15.ID_CONTA_CONTABIL_DESTINO in (218912902, 218942902, 218952902, 218912901))''''
```

### ###L4+ Demais Dívidas Contratuais

```
query7 = ""select          sum((a11.VA_MOVIMENTO_LIQUIDO * a13.PE_TAXA))  SDCC  
from      BD_DWTG.DBO.WF_LANCAMENTO_EP20  a11  
          join      BD_DWTG.DBO.WD_MOEDA    a12  
                on  (a11.ID_MOEDA_UG_EXEC_H = a12.ID_MOEDA)  
          join      BD_DWTG.DBO.WD_TAXA_CAMBIO_MENSAL      a13  
                on  (a12.ID_MOEDA = a13.ID_MOEDA_ORIGEM)  
          join      BD_DWTG.DBO.WA_MES_ACUM  a14
```

```

on (a11.ID_ANO_LANC = a14.ID_ANO_ACUM_ANO_SALDO and
a11.ID_MES_LANC = a14.ID_MES_ACUM_ANO_SALDO)
join BD_DWTG.DBO.WD_CONTA_CONTABIL_EXERCICIO a15
on (a11.ID_ANO_LANC = a15.ID_ANO and
a11.ID_CONTA_CONTABIL_LANC = a15.ID_CONTA_CONTABIL)
join BD_DWTG.DBO.WD_UG_EXERCICIO a16
on (a11.ID_ANO_LANC = a16.ID_ANO and
a11.ID_UG_EXEC = a16.ID_UG)
join BD_DWTG.DBO.WD_MES a17
on (a14.ID_ANO = a17.ID_ANO and
a14.ID_MES = a17.ID_MES)
where (a11.ID_ORCA_FISCAL_ORGAO_EXEC_H in (0)
and (a14.ID_ANO = {} and a14.ID_MES = {}))
and a11.IN_ISF_LANC in ('P')
and (a15.ID_CONTA_CONTABIL_DESTINO in (222210200, 212210300, 222110200, 212110301, 212
110303, 212510103, 212140303, 212150303, 212540103, 212550103, 212140301, 212150301, 21
7310301, 217310602, 217350402, 227310301, 212110700, 212210601, 212310201, 212310202, 2
12410201, 222310101, 222310102, 222410101, 217710101, 227710101)
or (a15.ID_CONTA_CONTABIL_DESTINO in (227310401)
and a16.ID_UG not in (170512)))
and a13.ID_ANO = a17.ID_ANO
and a13.ID_MES = a17.ID_MES)""""

```

###L5+ *Precatórios Posteriores a 05/05/2000*

```

query8 = """"select sum((a11.VA_MOVIMENTO_LIQUIDO * a13.PE_TAXA)) SDCC
from BD_DWTG.DBO.WF_LANCAMENTO_EP03 a11
join BD_DWTG.DBO.WD_MOEDA a12
on (a11.ID_MOEDA_UG_EXEC_H = a12.ID_MOEDA)
join BD_DWTG.DBO.WD_TAXA_CAMBIO_MENSAL a13
on (a12.ID_MOEDA = a13.ID_MOEDA_ORIGEM)
join BD_DWTG.DBO.WA_MES_ACUM a14
on (a11.ID_ANO_LANC = a14.ID_ANO_ACUM_ANO_SALDO and
a11.ID_MES_LANC = a14.ID_MES_ACUM_ANO_SALDO)
join BD_DWTG.DBO.WD_CONTA_CONTABIL_EXERCICIO a15
on (a11.ID_ANO_LANC = a15.ID_ANO and
a11.ID_CONTA_CONTABIL_LANC = a15.ID_CONTA_CONTABIL)
join BD_DWTG.DBO.WD_MES a16
on (a14.ID_ANO = a16.ID_ANO and
a14.ID_MES = a16.ID_MES)
where (a11.ID_ORCA_FISCAL_ORGAO_EXEC_H in (0)
and (a14.ID_ANO = {} and a14.ID_MES = {}))
and a11.ID_ACAO_PT in ('0005')
and a15.ID_CONTA_CONTABIL_DESTINO in (631100000, 631200000, 631300000, 631510000, 6315
20000, 632100000, 522110101, 522110201, 522110209, 522120101, 522120103, 522120201, 522
120202, 522120203, 522120301, 522120302, 522120303, 522190100, 522190200, 522190300, 52
2190400, 522190101, 522190109, 522190201, 522190209, 522190301, 522190309)
and a13.ID_ANO = a16.ID_ANO
and a13.ID_MES = a16.ID_MES)""""

```

###L6+ *Dívida Assumida pela União (L8727)*

```

query9 = """"select sum((a11.VA_MOVIMENTO_LIQUIDO * a13.PE_TAXA)) SDCC
from BD_DWTG.DBO.WF_LANCAMENTO_EP20 a11
join BD_DWTG.DBO.WD_MOEDA a12
on (a11.ID_MOEDA_UG_EXEC_H = a12.ID_MOEDA)
join BD_DWTG.DBO.WD_TAXA_CAMBIO_MENSAL a13
on (a12.ID_MOEDA = a13.ID_MOEDA_ORIGEM)
join BD_DWTG.DBO.WA_MES_ACUM a14

```

```

on (a11.ID_ANO_LANC = a14.ID_ANO_ACUM_ANO_SALDO and
a11.ID_MES_LANC = a14.ID_MES_ACUM_ANO_SALDO)
join BD_DWTG.DBO.WD_CONTA_CONTABIL_EXERCICIO a15
on (a11.ID_ANO_LANC = a15.ID_ANO and
a11.ID_CONTA_CONTABIL_LANC = a15.ID_CONTA_CONTABIL)
join BD_DWTG.DBO.WD_UG_EXERCICIO a16
on (a11.ID_ANO_LANC = a16.ID_ANO and
a11.ID_UG_EXEC = a16.ID_UG)
join BD_DWTG.DBO.WD_MES a17
on (a14.ID_ANO = a17.ID_ANO and
a14.ID_MES = a17.ID_MES)
where (a11.ID_ORCA_FISCAL_ORGAO_EXEC_H in (0)
and (a14.ID_ANO = {}) and a14.ID_MES = {})
and a16.ID_UG in (170512)
and a15.ID_CONTA_CONTABIL_DESTINO in (218912600, 228911600, 227310401)
and a11.IN_ISF_LANC in ('P')
and a13.ID_ANO = a17.ID_ANO
and a13.ID_MES = a17.ID_MES)""

```

### ###L6+ Passivos por Insuficiência de Recursos

```

query10 = ""select sum((a11.VA_MOVIMENTO_LIQUIDO * a13.PE_TAXA)) SDCC
from BD_DWTG.DBO.WF_LANCAMENTO_EP20 a11
join BD_DWTG.DBO.WD_MOEDA a12
on (a11.ID_MOEDA_UG_EXEC_H = a12.ID_MOEDA)
join BD_DWTG.DBO.WD_TAXA_CAMBIO_MENSAL a13
on (a12.ID_MOEDA = a13.ID_MOEDA_ORIGEM)
join BD_DWTG.DBO.WA_MES_ACUM a14
on (a11.ID_ANO_LANC = a14.ID_ANO_ACUM_ANO_SALDO and
a11.ID_MES_LANC = a14.ID_MES_ACUM_ANO_SALDO)
join BD_DWTG.DBO.WD_CONTA_CONTABIL_EXERCICIO a15
on (a11.ID_ANO_LANC = a15.ID_ANO and
a11.ID_CONTA_CONTABIL_LANC = a15.ID_CONTA_CONTABIL)
join BD_DWTG.DBO.WD_MES a16
on (a14.ID_ANO = a16.ID_ANO and
a14.ID_MES = a16.ID_MES)
where (a11.ID_ORCA_FISCAL_ORGAO_EXEC_H in (0)
and (a14.ID_ANO = {}) and a14.ID_MES = {})
and a15.ID_CONTA_CONTABIL_DESTINO in (213110400, 211110101, 211419800, 223110100, 2112
10100, 213140400, 213150400, 214119900, 211449800, 211459800)
and a11.IN_ISF_LANC in ('P')
and a13.ID_ANO = a16.ID_ANO
and a13.ID_MES = a16.ID_MES)""

```

### ###L5- Deduções: Precatórios

```

query11 = ""select sum((a11.VA_MOVIMENTO_LIQUIDO * a13.PE_TAXA)) SDCC
from BD_DWTG.DBO.WF_LANCAMENTO_EP03 a11
join BD_DWTG.DBO.WD_MOEDA a12
on (a11.ID_MOEDA_UG_EXEC_H = a12.ID_MOEDA)
join BD_DWTG.DBO.WD_TAXA_CAMBIO_MENSAL a13
on (a12.ID_MOEDA = a13.ID_MOEDA_ORIGEM)
join BD_DWTG.DBO.WA_MES_ACUM a14
on (a11.ID_ANO_LANC = a14.ID_ANO_ACUM_ANO_SALDO and
a11.ID_MES_LANC = a14.ID_MES_ACUM_ANO_SALDO)
join BD_DWTG.DBO.WD_CONTA_CONTABIL_EXERCICIO a15
on (a11.ID_ANO_LANC = a15.ID_ANO and
a11.ID_CONTA_CONTABIL_LANC = a15.ID_CONTA_CONTABIL)
join BD_DWTG.DBO.WD_MES a16

```

```

        on (a14.ID_ANO = a16.ID_ANO and
           a14.ID_MES = a16.ID_MES)
where (a11.ID_ORCA_FISCAL_ORGAO_EXEC_H in (0)
      and (a14.ID_ANO = {} and a14.ID_MES = {}))
      and a11.ID_ACAO_PT in ('0005')
      and a15.ID_CONTA_CONTABIL_DESTINO in (622130400)
      and a13.ID_ANO = a16.ID_ANO
      and a13.ID_MES = a16.ID_MES)""""

```

###coringas retirados à direita

###and (cast(a17.ID\_CONTA\_CONTABIL as varchar) like '1111102%'

###or cast(a17.ID\_CONTA\_CONTABIL as varchar) like '1111103%'

### or cast(a17.ID\_CONTA\_CONTABIL as varchar) like '1111104%'))""""

###L9+ Deduções: Depósitos do TN (em BCB)

```

query12 = """"select      sum((a11.VA_MOVIMENTO_LIQUIDO * a13.PE_TAXA))  SDCC
from      BD_DWTG.DBO.WF_LANCAMENTO_EP02  a11
join      BD_DWTG.DBO.WD_MOEDA            a12
          on (a11.ID_MOEDA_UG_EXEC_H = a12.ID_MOEDA)
join      BD_DWTG.DBO.WD_TAXA_CAMBIO_MENSAL  a13
          on (a12.ID_MOEDA = a13.ID_MOEDA_ORIGEM)
join      BD_DWTG.DBO.WA_MES_ACUM          a14
          on (a11.ID_ANO_LANC = a14.ID_ANO_ACUM_ANO_SALDO and
             a11.ID_MES_LANC = a14.ID_MES_ACUM_ANO_SALDO)
join      BD_DWTG.DBO.WD_CONTA_CONTABIL_EXERCICIO a15
          on (a11.ID_ANO_LANC = a15.ID_ANO and
             a11.ID_CONTA_CONTABIL_LANC = a15.ID_CONTA_CONTABIL)
join      BD_DWTG.DBO.WD_MES              a16
          on (a14.ID_ANO = a16.ID_ANO and
             a14.ID_MES = a16.ID_MES)
join      BD_DWTG.DBO.WD_CONTA_CONTABIL    a17
          on (a15.ID_CONTA_CONTABIL_DESTINO = a17.ID_CONTA_CONTABIL)
where (a11.ID_ORCA_FISCAL_ORGAO_EXEC_H in (0)
      and (a14.ID_ANO = {} and a14.ID_MES = {}))
      and a13.ID_ANO = a16.ID_ANO
      and a13.ID_MES = a16.ID_MES
      and (cast(a17.ID_CONTA_CONTABIL as varchar) like '1111102%'
          or cast(a17.ID_CONTA_CONTABIL as varchar) like '1111103%'
          or cast(a17.ID_CONTA_CONTABIL as varchar) like '1111104%'))""""

```

###L9+ Deduções: Depósitos à Vista

```

query13 = """"select      sum((a11.VA_MOVIMENTO_LIQUIDO * a13.PE_TAXA))  SDCC
from      BD_DWTG.DBO.WF_LANCAMENTO_EP02  a11
join      BD_DWTG.DBO.WD_MOEDA            a12
          on (a11.ID_MOEDA_UG_EXEC_H = a12.ID_MOEDA)
join      BD_DWTG.DBO.WD_TAXA_CAMBIO_MENSAL  a13
          on (a12.ID_MOEDA = a13.ID_MOEDA_ORIGEM)
join      BD_DWTG.DBO.WA_MES_ACUM          a14
          on (a11.ID_ANO_LANC = a14.ID_ANO_ACUM_ANO_SALDO and
             a11.ID_MES_LANC = a14.ID_MES_ACUM_ANO_SALDO)
join      BD_DWTG.DBO.WD_CONTA_CONTABIL_EXERCICIO a15
          on (a11.ID_ANO_LANC = a15.ID_ANO and
             a11.ID_CONTA_CONTABIL_LANC = a15.ID_CONTA_CONTABIL)
join      BD_DWTG.DBO.WD_UG_EXERCICIO      a16
          on (a11.ID_ANO_LANC = a16.ID_ANO and
             a11.ID_UG_EXEC = a16.ID_UG)
join      BD_DWTG.DBO.WD_MES              a17
          on (a14.ID_ANO = a17.ID_ANO and
             a14.ID_MES = a17.ID_MES)
join      BD_DWTG.DBO.WD_CONTA_CONTABIL    a18

```

```

on (a15.ID_CONTA_CONTABIL_DESTINO = a18.ID_CONTA_CONTABIL)
where (a11.ID_ORCA_FISCAL_ORGAO_EXEC_H in (0)
and (a14.ID_ANO = {} and a14.ID_MES = {}))
and a16.ID_ORGAO_UG not in (25901)
and a16.ID_UG not in (380916)
and (cast(a18.ID_CONTA_CONTABIL as varchar) like '1111119%'
or cast(a18.ID_CONTA_CONTABIL as varchar) like '1112102%'
or cast(a18.ID_CONTA_CONTABIL as varchar) like '1112103%'
or cast(a18.ID_CONTA_CONTABIL as varchar) like '1112150%'
or cast(a18.ID_CONTA_CONTABIL as varchar) like '1112152%')
and a13.ID_ANO = a17.ID_ANO
and a13.ID_MES = a17.ID_MES)""

```

###L10+- Deduções: Disponibilidade FAT

```

query14 = ""select      sum((a11.VA_MOVIMENTO_LIQUIDO * a13.PE_TAXA))  SDCC
from      BD_DWTG.DBO.WF_LANCAMENTO_EP02  a11
join      BD_DWTG.DBO.WD_MOEDA            a12
on        (a11.ID_MOEDA_UG_EXEC_H = a12.ID_MOEDA)
join      BD_DWTG.DBO.WD_TAXA_CAMBIO_MENSAL      a13
on        (a12.ID_MOEDA = a13.ID_MOEDA_ORIGEM)
join      BD_DWTG.DBO.WA_MES_ACUM a14
on        (a11.ID_ANO_LANC = a14.ID_ANO_ACUM_ANO_SALDO and
a11.ID_MES_LANC = a14.ID_MES_ACUM_ANO_SALDO)
join      BD_DWTG.DBO.WD_CONTA_CONTABIL_EXERCICIO a15
on        (a11.ID_ANO_LANC = a15.ID_ANO and
a11.ID_CONTA_CONTABIL_LANC = a15.ID_CONTA_CONTABIL)
join      BD_DWTG.DBO.WD_UG_EXERCICIO      a16
on        (a11.ID_ANO_LANC = a16.ID_ANO and
a11.ID_UG_EXEC = a16.ID_UG)
join      BD_DWTG.DBO.WD_MES            a17
on        (a14.ID_ANO = a17.ID_ANO and
a14.ID_MES = a17.ID_MES)
join      BD_DWTG.DBO.WD_CONTA_CONTABIL  a18
on        (a15.ID_CONTA_CONTABIL_DESTINO = a18.ID_CONTA_CONTABIL)
where     (a11.ID_ORCA_FISCAL_ORGAO_EXEC_H in (0)
and (a14.ID_ANO = {} and a14.ID_MES = {}))
and a16.ID_UG in (380916)
and (cast(a18.ID_CONTA_CONTABIL as varchar) like '1111119%'
or cast(a18.ID_CONTA_CONTABIL as varchar) like '1124103%'
or cast(a18.ID_CONTA_CONTABIL as varchar) like '1135407%'
or cast(a18.ID_CONTA_CONTABIL as varchar) like '1135113%'
or cast(a18.ID_CONTA_CONTABIL as varchar) like '1135115%'
or cast(a18.ID_CONTA_CONTABIL as varchar) like '1124101%'
or cast(a18.ID_CONTA_CONTABIL as varchar) like '1135111%'
or cast(a18.ID_CONTA_CONTABIL as varchar) like '1135107%'
or cast(a18.ID_CONTA_CONTABIL as varchar) like '11121%'
or cast(a18.ID_CONTA_CONTABIL as varchar) like '1135114%'
or cast(a18.ID_CONTA_CONTABIL as varchar) like '1135112%'
or cast(a18.ID_CONTA_CONTABIL as varchar) like '1135116%'
or cast(a18.ID_CONTA_CONTABIL as varchar) like '1211503%'
or cast(a18.ID_CONTA_CONTABIL as varchar) like '1211403%'
or cast(a18.ID_CONTA_CONTABIL as varchar) like '1211103%'
or cast(a18.ID_CONTA_CONTABIL as varchar) like '1135507%'
or cast(a18.ID_CONTA_CONTABIL as varchar) like '1212105%'
or a15.ID_CONTA_CONTABIL_DESTINO in (111115009, 111115011))
and a13.ID_ANO = a17.ID_ANO
and a13.ID_MES = a17.ID_MES)""

```

###L10+ Deduções: (+)Aplicações em Fundos Diversos 1

```
query15 = ""select      sum((a11.VA_MOVIMENTO_LIQUIDO * a13.PE_TAXA))  SDCC
from    BD_DWTG.DBO.WF_LANCAMENTO_EP02  a11
      join    BD_DWTG.DBO.WD_MOEDA      a12
            on    (a11.ID_MOEDA_UG_EXEC_H = a12.ID_MOEDA)
      join    BD_DWTG.DBO.WD_TAXA_CAMBIO_MENSAL      a13
            on    (a12.ID_MOEDA = a13.ID_MOEDA_ORIGEM)
      join    BD_DWTG.DBO.WA_MES_ACUM  a14
            on    (a11.ID_ANO_LANC = a14.ID_ANO_ACUM_ANO_SALDO and
a11.ID_MES_LANC = a14.ID_MES_ACUM_ANO_SALDO)
      join    BD_DWTG.DBO.WD_CONTA_CONTABIL_EXERCICIO  a15
            on    (a11.ID_ANO_LANC = a15.ID_ANO and
a11.ID_CONTA_CONTABIL_LANC = a15.ID_CONTA_CONTABIL)
      join    BD_DWTG.DBO.WD_UG_EXERCICIO      a16
            on    (a11.ID_ANO_LANC = a16.ID_ANO and
a11.ID_UG_EXEC = a16.ID_UG)
      join    BD_DWTG.DBO.WD_MES      a17
            on    (a14.ID_ANO = a17.ID_ANO and
a14.ID_MES = a17.ID_MES)
where    (a11.ID_ORCA_FISCAL_ORGAO_EXEC_H in (0)
and (a14.ID_ANO = {} and a14.ID_MES = {}))
and a15.ID_CONTA_CONTABIL_DESTINO in (111215100)
and a16.ID_ORGAO_UG not in (37904, 25915)
and a13.ID_ANO = a17.ID_ANO
and a13.ID_MES = a17.ID_MES)""
```

###L10+ Deduções: (+)Aplicações em Fundos Diversos 2

```
query16 = ""select      sum((a11.VA_MOVIMENTO_LIQUIDO * a13.PE_TAXA))  SDCC
from    BD_DWTG.DBO.WF_LANCAMENTO_EP02  a11
      join    BD_DWTG.DBO.WD_MOEDA      a12
            on    (a11.ID_MOEDA_UG_EXEC_H = a12.ID_MOEDA)
      join    BD_DWTG.DBO.WD_TAXA_CAMBIO_MENSAL      a13
            on    (a12.ID_MOEDA = a13.ID_MOEDA_ORIGEM)
      join    BD_DWTG.DBO.WA_MES_ACUM  a14
            on    (a11.ID_ANO_LANC = a14.ID_ANO_ACUM_ANO_SALDO and
a11.ID_MES_LANC = a14.ID_MES_ACUM_ANO_SALDO)
      join    BD_DWTG.DBO.WD_CONTA_CONTABIL_EXERCICIO  a15
            on    (a11.ID_ANO_LANC = a15.ID_ANO and
a11.ID_CONTA_CONTABIL_LANC = a15.ID_CONTA_CONTABIL)
      join    BD_DWTG.DBO.WD_UG_EXERCICIO      a16
            on    (a11.ID_ANO_LANC = a16.ID_ANO and
a11.ID_UG_EXEC = a16.ID_UG)
      join    BD_DWTG.DBO.WD_MES      a17
            on    (a14.ID_ANO = a17.ID_ANO and
a14.ID_MES = a17.ID_MES)
      join    BD_DWTG.DBO.WD_ORGAO      a18
            on    (a16.ID_ORGAO_UG = a18.ID_ORGAO)
      join    BD_DWTG.DBO.WD_CONTA_CONTABIL      a19
            on    (a15.ID_CONTA_CONTABIL_DESTINO = a19.ID_CONTA_CONTABIL)
where    (a11.ID_ORCA_FISCAL_ORGAO_EXEC_H in (0)
and (a14.ID_ANO = {} and a14.ID_MES = {}))
and a18.ID_TP_ADMIN_ORGAO in (7)
and a16.ID_ORGAO_UG not in (37904, 25915)
and cast(a19.ID_CONTA_CONTABIL as varchar) like '23%'
and a13.ID_ANO = a17.ID_ANO
and a13.ID_MES = a17.ID_MES)""
```

###L10- Deduções: (-) Aplicações em Fundos Diversos 3

```

query17 = ""select      sum((a11.VA_MOVIMENTO_LIQUIDO * a13.PE_TAXA))  SDCC
from    BD_DWTG.DBO.WF_LANCAMENTO_EP02  a11
      join    BD_DWTG.DBO.WD_MOEDA      a12
            on    (a11.ID_MOEDA_UG_EXEC_H = a12.ID_MOEDA)
      join    BD_DWTG.DBO.WD_TAXA_CAMBIO_MENSAL      a13
            on    (a12.ID_MOEDA = a13.ID_MOEDA_ORIGEM)
      join    BD_DWTG.DBO.WA_MES_ACUM  a14
            on    (a11.ID_ANO_LANC = a14.ID_ANO_ACUM_ANO_SALDO and
a11.ID_MES_LANC = a14.ID_MES_ACUM_ANO_SALDO)
      join    BD_DWTG.DBO.WD_CONTA_CONTABIL_EXERCICIO  a15
            on    (a11.ID_ANO_LANC = a15.ID_ANO and
a11.ID_CONTA_CONTABIL_LANC = a15.ID_CONTA_CONTABIL)
      join    BD_DWTG.DBO.WD_UG_EXERCICIO      a16
            on    (a11.ID_ANO_LANC = a16.ID_ANO and
a11.ID_UG_EXEC = a16.ID_UG)
      join    BD_DWTG.DBO.WD_MES      a17
            on    (a14.ID_ANO = a17.ID_ANO and
a14.ID_MES = a17.ID_MES)
      join    BD_DWTG.DBO.WD_CONTA_CONTABIL  a18
            on    (a15.ID_CONTA_CONTABIL_DESTINO = a18.ID_CONTA_CONTABIL)
      join    BD_DWTG.DBO.WD_ORGAO      a19
            on    (a16.ID_ORGAO_UG = a19.ID_ORGAO)
where    (a11.ID_ORCA_FISCAL_ORGAO_EXEC_H in (0)
and (a14.ID_ANO = {} and a14.ID_MES = {}))
and a19.ID_TP_ADMIN_ORGAO in (7)
and a16.ID_ORGAO_UG not in (37904, 25915)
and (cast(a18.ID_CONTA_CONTABIL as varchar) like '1111102%'
or cast(a18.ID_CONTA_CONTABIL as varchar) like '1111103%'
or cast(a18.ID_CONTA_CONTABIL as varchar) like '1111104%'
or a15.ID_CONTA_CONTABIL_DESTINO in (111210200, 111210300, 111215000, 111215200)
or (cast(a18.ID_CONTA_CONTABIL as varchar) like '1111119%'
and a16.ID_UG not in (380916))
or (a18.ID_CLASSE_CCON = 1
and (a18.ID_GRUPO_CCON = 2
and a18.ID_SUBGRUPO_CCON in (2, 3, 4, 5))))
and a13.ID_ANO = a17.ID_ANO
and a13.ID_MES = a17.ID_MES)""

```

###L10+- Deduções: Dívidas Renegociadas

```

query18 = ""select      sum((a11.VA_MOVIMENTO_LIQUIDO * a13.PE_TAXA))  SDCC
from    BD_DWTG.DBO.WF_LANCAMENTO_EP20  a11
      join    BD_DWTG.DBO.WD_MOEDA      a12
            on    (a11.ID_MOEDA_UG_EXEC_H = a12.ID_MOEDA)
      join    BD_DWTG.DBO.WD_TAXA_CAMBIO_MENSAL      a13
            on    (a12.ID_MOEDA = a13.ID_MOEDA_ORIGEM)
      join    BD_DWTG.DBO.WA_MES_ACUM  a14
            on    (a11.ID_ANO_LANC = a14.ID_ANO_ACUM_ANO_SALDO and
a11.ID_MES_LANC = a14.ID_MES_ACUM_ANO_SALDO)
      join    BD_DWTG.DBO.WD_CONTA_CONTABIL_EXERCICIO  a15
            on    (a11.ID_ANO_LANC = a15.ID_ANO and
a11.ID_CONTA_CONTABIL_LANC = a15.ID_CONTA_CONTABIL)
      join    BD_DWTG.DBO.WD_UG_EXERCICIO      a16
            on    (a11.ID_ANO_LANC = a16.ID_ANO and
a11.ID_UG_EXEC = a16.ID_UG)
      join    BD_DWTG.DBO.WD_MES      a17
            on    (a14.ID_ANO = a17.ID_ANO and
a14.ID_MES = a17.ID_MES)

```

```

join    BD_DWTG.DBO.WD_ENTIDADE a18
on      (a11.ID_ENTIDADE_CCOR = a18.ID_ENTIDADE and
a11.ID_TP_ENTIDADE_CCOR = a18.ID_TP_ENTIDADE)
where   (a11.ID_ORCA_FISCAL_ORGAO_EXEC_H in (0)
and (a14.ID_ANO = {}) and a14.ID_MES = {})
and a16.ID_UG in (170512)
and a15.ID_CONTA_CONTABIL_DESTINO in (121110301, 112410100, 121110318, 112410600, 1211
40301, 121150301, 112440100, 112450100, 121140318, 121150318, 112440600, 112450600, 112
410401, 121219818, 112450401, 112440401, 121249818, 113814200, 113844200, 113854200)
and ((a11.ID_TP_ENTIDADE_CCOR = 'IC'
and a11.ID_ENTIDADE_CCOR in ('PF1705320', 'PF1705524', 'PF1705528', 'PF1705546', 'PF17
05547', 'PF1705548', 'PF1705406', 'PF1705525', 'PF1705529', 'PF1705544', 'PF1705545'))
or a18.NO_ENTIDADE like '%9.496/97%'
or a18.NO_ENTIDADE like '%2.185/2001%')
and a13.ID_ANO = a17.ID_ANO
and a13.ID_MES = a17.ID_MES)""""

```

###L10+- Deduções: Créditos Lei 8.727/93

```

query19 = """"select      sum((a11.VA_MOVIMENTO_LIQUIDO * a13.PE_TAXA))  SDCC
from    BD_DWTG.DBO.WF_LANCAMENTO_EP20  a11
join    BD_DWTG.DBO.WD_MOEDA      a12
on      (a11.ID_MOEDA_UG_EXEC_H = a12.ID_MOEDA)
join    BD_DWTG.DBO.WD_TAXA_CAMBIO_MENSAL      a13
on      (a12.ID_MOEDA = a13.ID_MOEDA_ORIGEM)
join    BD_DWTG.DBO.WA_MES_ACUM a14
on      (a11.ID_ANO_LANC = a14.ID_ANO_ACUM_ANO_SALDO and
a11.ID_MES_LANC = a14.ID_MES_ACUM_ANO_SALDO)
join    BD_DWTG.DBO.WD_CONTA_CONTABIL_EXERCICIO a15
on      (a11.ID_ANO_LANC = a15.ID_ANO and
a11.ID_CONTA_CONTABIL_LANC = a15.ID_CONTA_CONTABIL)
join    BD_DWTG.DBO.WD_UG_EXERCICIO      a16
on      (a11.ID_ANO_LANC = a16.ID_ANO and
a11.ID_UG_EXEC = a16.ID_UG)
join    BD_DWTG.DBO.WD_MES      a17
on      (a14.ID_ANO = a17.ID_ANO and
a14.ID_MES = a17.ID_MES)
join    BD_DWTG.DBO.WD_ENTIDADE a18
on      (a11.ID_ENTIDADE_CCOR = a18.ID_ENTIDADE and
a11.ID_TP_ENTIDADE_CCOR = a18.ID_TP_ENTIDADE)
where   (a11.ID_ORCA_FISCAL_ORGAO_EXEC_H in (0)
and (a14.ID_ANO = {}) and a14.ID_MES = {})
and a16.ID_UG in (170512)
and a15.ID_CONTA_CONTABIL_DESTINO in (121110301, 112410100, 121110318, 112410600, 1211
40301, 121150301, 112440100, 112450100, 121140318, 121150318, 112440600, 112450600, 112
410401, 121219818, 112450401, 112440401, 121249818, 113814200, 113844200, 113854200)
and ((a11.ID_TP_ENTIDADE_CCOR = 'IC'
and a11.ID_ENTIDADE_CCOR in ('PF1705109', 'PF1705536', 'TN0000016', 'TN0000017'))
or (a11.ID_TP_ENTIDADE_CCOR = 'IN' and a11.ID_ENTIDADE_CCOR = 'PF1705109')
or a18.NO_ENTIDADE like '%8.727/93%')
and a13.ID_ANO = a17.ID_ANO
and a13.ID_MES = a17.ID_MES)""""

```

###L10+- Deduções: Dívida Externa Renegociada

```

query20 = """"select      sum((a11.VA_MOVIMENTO_LIQUIDO * a13.PE_TAXA))  SDCC
from    BD_DWTG.DBO.WF_LANCAMENTO_EP20  a11
join    BD_DWTG.DBO.WD_MOEDA      a12
on      (a11.ID_MOEDA_UG_EXEC_H = a12.ID_MOEDA)

```



```

join      BD_DWTG.DBO.WD_TAXA_CAMBIO_MENSAL      a13
  on      (a12.ID_MOEDA = a13.ID_MOEDA_ORIGEM)
join      BD_DWTG.DBO.WA_MES_ACUM a14
  on      (a11.ID_ANO_LANC = a14.ID_ANO_ACUM_ANO_SALDO and
a11.ID_MES_LANC = a14.ID_MES_ACUM_ANO_SALDO)
join      BD_DWTG.DBO.WD_CONTA_CONTABIL_EXERCICIO a15
  on      (a11.ID_ANO_LANC = a15.ID_ANO and
a11.ID_CONTA_CONTABIL_LANC = a15.ID_CONTA_CONTABIL)
join      BD_DWTG.DBO.WD_UG_EXERCICIO      a16
  on      (a11.ID_ANO_LANC = a16.ID_ANO and
a11.ID_UG_EXEC = a16.ID_UG)
join      BD_DWTG.DBO.WD_MES      a17
  on      (a14.ID_ANO = a17.ID_ANO and
a14.ID_MES = a17.ID_MES)
join      BD_DWTG.DBO.WD_ENTIDADE a18
  on      (a11.ID_ENTIDADE_CCOR = a18.ID_ENTIDADE and
a11.ID_TP_ENTIDADE_CCOR = a18.ID_TP_ENTIDADE)
where     (a11.ID_ORCA_FISCAL_ORGAO_EXEC_H in (0)
and (a14.ID_ANO = {} and a14.ID_MES = {}))
and a16.ID_UG in (170512)
and a15.ID_CONTA_CONTABIL_DESTINO in (121110301, 112410100, 121110318, 112410600, 1211
40301, 121150301, 112440100, 112450100, 121140318, 121150318, 112440600, 112450600, 112
410401, 121219818, 112450401, 112440401, 121249818, 113844200, 113814200, 113854200)
and (a11.ID_ENTIDADE_CCOR in ('PF1705104', 'PF1705114', 'PF1705117', 'PF1705521', 'PF1
705534', 'PF1705116', 'PF1705531', 'PF1705532', 'PF1705113', 'PF1701536', 'PF1705520',
'PF1705533', 'PF1705464', 'PF1705534', 'PF1705119', 'PF1705384'))
or a18.NO_ENTIDADE like '%DMLP%'
or a18.NO_ENTIDADE like '%FRANCA%'
or a18.NO_ENTIDADE like '%EXTER%'
or a18.NO_ENTIDADE like '%FRANÇA%'
or a18.NO_ENTIDADE like '%MF 030%'
or a18.NO_ENTIDADE like '%BIB%')
and a13.ID_ANO = a17.ID_ANO
and a13.ID_MES = a17.ID_MES)""""

```

###L10+ Deduções: Demais Dívidas Renegociadas (subtrair as outras dívidas)

```

query21 = """"select      sum((a11.VA_MOVIMENTO_LIQUIDO * a13.PE_TAXA)) SDCC
from      BD_DWTG.DBO.WF_LANCAMENTO_EP02 a11
  join    BD_DWTG.DBO.WD_MOEDA      a12
    on    (a11.ID_MOEDA_UG_EXEC_H = a12.ID_MOEDA)
  join    BD_DWTG.DBO.WD_TAXA_CAMBIO_MENSAL      a13
    on    (a12.ID_MOEDA = a13.ID_MOEDA_ORIGEM)
  join    BD_DWTG.DBO.WA_MES_ACUM a14
    on    (a11.ID_ANO_LANC = a14.ID_ANO_ACUM_ANO_SALDO and
a11.ID_MES_LANC = a14.ID_MES_ACUM_ANO_SALDO)
  join    BD_DWTG.DBO.WD_CONTA_CONTABIL_EXERCICIO a15
    on    (a11.ID_ANO_LANC = a15.ID_ANO and
a11.ID_CONTA_CONTABIL_LANC = a15.ID_CONTA_CONTABIL)
  join    BD_DWTG.DBO.WD_UG_EXERCICIO      a16
    on    (a11.ID_ANO_LANC = a16.ID_ANO and
a11.ID_UG_EXEC = a16.ID_UG)
  join    BD_DWTG.DBO.WD_MES      a17
    on    (a14.ID_ANO = a17.ID_ANO and
a14.ID_MES = a17.ID_MES)
where     (a11.ID_ORCA_FISCAL_ORGAO_EXEC_H in (0)
and (a14.ID_ANO = {} and a14.ID_MES = {}))
and a16.ID_UG in (170512)
and a15.ID_CONTA_CONTABIL_DESTINO in (121110301, 112410100, 121110318, 112410600, 1211
40301, 121150301, 112440100, 112450100, 121140318, 121150318, 112440600, 112450600, 112

```

410401, 121219818, 112450401, 112440401, 121249818, 113844200, 113814200, 113854200)  
and a13.ID\_ANO = a17.ID\_ANO  
and a13.ID\_MES = a17.ID\_MES)""

###L10+ Deduções: Ajustes para Perdas

```
query22 = ""select      sum((a11.VA_MOVIMENTO_LIQUIDO * a13.PE_TAXA))  SDCC
from      BD_DWTG.DBO.WF_LANCAMENTO_EP02  a11
         join      BD_DWTG.DBO.WD_MOEDA      a12
           on      (a11.ID_MOEDA_UG_EXEC_H = a12.ID_MOEDA)
         join      BD_DWTG.DBO.WD_TAXA_CAMBIO_MENSAL      a13
           on      (a12.ID_MOEDA = a13.ID_MOEDA_ORIGEM)
         join      BD_DWTG.DBO.WA_MES_ACUM a14
           on      (a11.ID_ANO_LANC = a14.ID_ANO_ACUM_ANO_SALDO and
a11.ID_MES_LANC = a14.ID_MES_ACUM_ANO_SALDO)
         join      BD_DWTG.DBO.WD_CONTA_CONTABIL_EXERCICIO a15
           on      (a11.ID_ANO_LANC = a15.ID_ANO and
a11.ID_CONTA_CONTABIL_LANC = a15.ID_CONTA_CONTABIL)
         join      BD_DWTG.DBO.WD_UG_EXERCICIO      a16
           on      (a11.ID_ANO_LANC = a16.ID_ANO and
a11.ID_UG_EXEC = a16.ID_UG)
         join      BD_DWTG.DBO.WD_MES      a17
           on      (a14.ID_ANO = a17.ID_ANO and
a14.ID_MES = a17.ID_MES)
where     (a11.ID_ORCA_FISCAL_ORGAO_EXEC_H in (0)
and (a14.ID_ANO = {} and a14.ID_MES = {}))
and a16.ID_UG in (170512)
and a15.ID_CONTA_CONTABIL_DESTINO in (121119902, 121149904, 121159904, 112940401, 1129
50401, 121119904)
and a13.ID_ANO = a17.ID_ANO
and a13.ID_MES = a17.ID_MES)""
```

###L10+ Deduções: Outros Créditos Bancários

```
query23 = ""select      sum((a11.VA_MOVIMENTO_LIQUIDO * a13.PE_TAXA))  SDCC
from      BD_DWTG.DBO.WF_LANCAMENTO_EP02  a11
         join      BD_DWTG.DBO.WD_MOEDA      a12
           on      (a11.ID_MOEDA_UG_EXEC_H = a12.ID_MOEDA)
         join      BD_DWTG.DBO.WD_TAXA_CAMBIO_MENSAL      a13
           on      (a12.ID_MOEDA = a13.ID_MOEDA_ORIGEM)
         join      BD_DWTG.DBO.WA_MES_ACUM a14
           on      (a11.ID_ANO_LANC = a14.ID_ANO_ACUM_ANO_SALDO and
a11.ID_MES_LANC = a14.ID_MES_ACUM_ANO_SALDO)
         join      BD_DWTG.DBO.WD_CONTA_CONTABIL_EXERCICIO a15
           on      (a11.ID_ANO_LANC = a15.ID_ANO and
a11.ID_CONTA_CONTABIL_LANC = a15.ID_CONTA_CONTABIL)
         join      BD_DWTG.DBO.WD_UG_EXERCICIO      a16
           on      (a11.ID_ANO_LANC = a16.ID_ANO and
a11.ID_UG_EXEC = a16.ID_UG)
         join      BD_DWTG.DBO.WD_MES      a17
           on      (a14.ID_ANO = a17.ID_ANO and
a14.ID_MES = a17.ID_MES)
where     (a11.ID_ORCA_FISCAL_ORGAO_EXEC_H in (0)
and (a14.ID_ANO = {} and a14.ID_MES = {}))
and a16.ID_UG in (170705, 170526, 170700)
and a15.ID_CONTA_CONTABIL_DESTINO in (112410301, 112410303, 112440301, 112450301, 1124
40303, 112450303, 112410100, 121110301, 121110314, 121110308, 121140301, 121150301, 121
140308, 121150308)
and a13.ID_ANO = a17.ID_ANO
```

```
and a13.ID_MES = a17.ID_MES)""
```

### ###L10+ Deduções: Outros Créditos Bancários - Ajustes para Perdas

```
query24 = ""select      sum((a11.VA_MOVIMENTO_LIQUIDO * a13.PE_TAXA))  SDCC
from    BD_DWTG.DBO.WF_LANCAMENTO_EP02  a11
        join    BD_DWTG.DBO.WD_MOEDA      a12
            on   (a11.ID_MOEDA_UG_EXEC_H = a12.ID_MOEDA)
        join    BD_DWTG.DBO.WD_TAXA_CAMBIO_MENSAL      a13
            on   (a12.ID_MOEDA = a13.ID_MOEDA_ORIGEM)
        join    BD_DWTG.DBO.WA_MES_ACUM  a14
            on   (a11.ID_ANO_LANC = a14.ID_ANO_ACUM_ANO_SALDO and
a11.ID_MES_LANC = a14.ID_MES_ACUM_ANO_SALDO)
        join    BD_DWTG.DBO.WD_CONTA_CONTABIL_EXERCICIO a15
            on   (a11.ID_ANO_LANC = a15.ID_ANO and
a11.ID_CONTA_CONTABIL_LANC = a15.ID_CONTA_CONTABIL)
        join    BD_DWTG.DBO.WD_UG_EXERCICIO      a16
            on   (a11.ID_ANO_LANC = a16.ID_ANO and
a11.ID_UG_EXEC = a16.ID_UG)
        join    BD_DWTG.DBO.WD_MES      a17
            on   (a14.ID_ANO = a17.ID_ANO and
a14.ID_MES = a17.ID_MES)
where    (a11.ID_ORCA_FISCAL_ORGAO_EXEC_H in (0)
and (a14.ID_ANO = {} and a14.ID_MES = {}))
and a16.ID_UG in (170705, 170526, 170700)
and a15.ID_CONTA_CONTABIL_DESTINO in (112910401, 121119904, 121119907)
and a13.ID_ANO = a17.ID_ANO
and a13.ID_MES = a17.ID_MES)""
```

### ###L3- Deduções: Resultado Positivo TN/BCB

```
query25 = ""select      sum((a11.VA_MOVIMENTO_LIQUIDO * a13.PE_TAXA))  SDCC
from    BD_DWTG.DBO.WF_LANCAMENTO_EP02  a11
        join    BD_DWTG.DBO.WD_MOEDA      a12
            on   (a11.ID_MOEDA_UG_EXEC_H = a12.ID_MOEDA)
        join    BD_DWTG.DBO.WD_TAXA_CAMBIO_MENSAL      a13
            on   (a12.ID_MOEDA = a13.ID_MOEDA_ORIGEM)
        join    BD_DWTG.DBO.WA_MES_ACUM  a14
            on   (a11.ID_ANO_LANC = a14.ID_ANO_ACUM_ANO_SALDO and
a11.ID_MES_LANC = a14.ID_MES_ACUM_ANO_SALDO)
        join    BD_DWTG.DBO.WD_CONTA_CONTABIL_EXERCICIO a15
            on   (a11.ID_ANO_LANC = a15.ID_ANO and
a11.ID_CONTA_CONTABIL_LANC = a15.ID_CONTA_CONTABIL)
        join    BD_DWTG.DBO.WD_MES      a16
            on   (a14.ID_ANO = a16.ID_ANO and
a14.ID_MES = a16.ID_MES)
where    (a11.ID_ORCA_FISCAL_ORGAO_EXEC_H in (0)
and (a14.ID_ANO = {} and a14.ID_MES = {}))
and a13.ID_ANO = a16.ID_ANO
and a13.ID_MES = a16.ID_MES
and a15.ID_CONTA_CONTABIL_DESTINO in (113813001, 113813002))""
```

### ###L11- (-) Restos a Pagar Processados

```
query26 = ""select      sum((a11.VA_MOVIMENTO_LIQUIDO * a13.PE_TAXA))  SDCC
from    BD_DWTG.DBO.WF_LANCAMENTO_EP03  a11
        join    BD_DWTG.DBO.WD_MOEDA      a12
            on   (a11.ID_MOEDA_UG_EXEC_H = a12.ID_MOEDA)
```

```

join      BD_DWTG.DBO.WD_TAXA_CAMBIO_MENSAL      a13
on        (a12.ID_MOEDA = a13.ID_MOEDA_ORIGEM)
join      BD_DWTG.DBO.WA_MES_ACUM a14
on        (a11.ID_ANO_LANC = a14.ID_ANO_ACUM_ANO_SALDO and
a11.ID_MES_LANC = a14.ID_MES_ACUM_ANO_SALDO)
join      BD_DWTG.DBO.WD_CONTA_CONTABIL_EXERCICIO a15
on        (a11.ID_ANO_LANC = a15.ID_ANO and
a11.ID_CONTA_CONTABIL_LANC = a15.ID_CONTA_CONTABIL)
join      BD_DWTG.DBO.WD_UG_EXERCICIO      a16
on        (a11.ID_ANO_LANC = a16.ID_ANO and
a11.ID_UG_EXEC = a16.ID_UG)
join      BD_DWTG.DBO.WD_MES      a17
on        (a14.ID_ANO = a17.ID_ANO and
a14.ID_MES = a17.ID_MES)
where     (a11.ID_ORCA_FISCAL_ORGAO_EXEC_H in (0)
and (a14.ID_ANO = {} and a14.ID_MES = {}))
and a11.ID_ACAO_PT not in ('0005')
and (a15.ID_CONTA_CONTABIL_DESTINO in (622920103, 622920107, 632100000, 631300000)
or (a15.ID_CONTA_CONTABIL_DESTINO in (622920102, 622920106, 631200000)
and a16.ID_UG not in (170600)))
and a13.ID_ANO = a17.ID_ANO
and a13.ID_MES = a17.ID_MES)""

```

###L2-Dívida Mobiliária TN Interna (INTRA - Dedução)

```

query27 = ""select      sum((a11.VA_MOVIMENTO_LIQUIDO * a13.PE_TAXA))  SDCC
from      BD_DWTG.DBO.WF_LANCAMENTO_EP02  a11
join      BD_DWTG.DBO.WD_MOEDA      a12
on        (a11.ID_MOEDA_UG_EXEC_H = a12.ID_MOEDA)
join      BD_DWTG.DBO.WD_TAXA_CAMBIO_MENSAL      a13
on        (a12.ID_MOEDA = a13.ID_MOEDA_ORIGEM)
join      BD_DWTG.DBO.WA_MES_ACUM a14
on        (a11.ID_ANO_LANC = a14.ID_ANO_ACUM_ANO_SALDO and
a11.ID_MES_LANC = a14.ID_MES_ACUM_ANO_SALDO)
join      BD_DWTG.DBO.WD_CONTA_CONTABIL_EXERCICIO a15
on        (a11.ID_ANO_LANC = a15.ID_ANO and
a11.ID_CONTA_CONTABIL_LANC = a15.ID_CONTA_CONTABIL)
join      BD_DWTG.DBO.WD_MES      a16
on        (a14.ID_ANO = a16.ID_ANO and
a14.ID_MES = a16.ID_MES)
where     (a11.ID_ORCA_FISCAL_ORGAO_EXEC_H in (0)
and (a14.ID_ANO = {} and a14.ID_MES = {}))
and a13.ID_ANO = a16.ID_ANO
and a13.ID_MES = a16.ID_MES
and a15.ID_CONTA_CONTABIL_DESTINO in (222120101))""

```

In [446]:

```
### Inserir o(s) período(s) nas 27 consultas nos quadrimestres do ano atual
for i in range(1,28):
    for i2 in [1,2,3]:
        globals()[f'query{i}_exec_quad{i2}'] = globals()[f'query{i}'].format(ano_atual,
quad_to_mes[globals()[f'quad{i2}']])

### Comando para rodar efetivamente as consultas
for i in range(1,28):
    for i2 in [1,2,3]:
        globals()[f'df_query{i}_quad{i2}'] = pd.read_sql(globals()[f'query{i}_exec_quad
{i2}'], conn)
```

In [447]:

```
### Inserir o período (mês = 12) nas 27 consultas nos quadrimestres do ano anterior
for i in range(1,28):
    globals()[f'query{i}_exec'] = globals()[f'query{i}'].format(ano_anterior,12)

### Comando para rodar efetivamente as consultas
for i in range(1,28):
    globals()[f'df_query{i}_ano_anterior'] = pd.read_sql(globals()[f'query{i}_exec'], c
onn)
```

In [565]:

```
### Substituição de "None" por 0 (zero) sempre que a consulta não retornar valor
for i in range(1,28):
    for i2 in [1,2,3]:
        globals()[f'df_query{i}_quad{i2}'].fillna(0,inplace=True)

for i in range(1,28):
    globals()[f'df_query{i}_ano_anterior'].fillna(0,inplace=True)
```

In [566]:

```
### Leitura e tratamento do arquivo referente à Receita Corrente Líquida
df_rcl = pd.read_excel('rcl.xls')
rcl_quad_atual = df_rcl[(df_rcl['ANO']==ano_atual) & (df_rcl['QUAD']==quadrimestre)]['R
CL']
rcl_ano_anterior = df_rcl[(df_rcl['ANO']==ano_anterior) & (df_rcl['QUAD']==3)]['RCL']
```

In [567]:

```
### Criação de dicionário para o demonstrativo detalhado da DCL
dic_detalha = {}
for i2 in ['ano_anterior', 'quad1', 'quad2', 'quad3']:
    for i in range(1,28):
        globals()[f'query{i}v'] = float(globals()[f'df_query{i}_{i2}']['SDCC'][0]/1000)
        if i2=='ano_anterior':
            globals()[f'rcl_{i2}'] = df_rcl[(df_rcl['ANO']==globals()[f'{i2}']) & (df_rcl['QUAD']==3)].RCL.values[0]
        else:
            globals()[f'rcl_{i2}'] = df_rcl[(df_rcl['ANO']==ano_atual) & (df_rcl['QUAD']==globals()[f'{i2}'])].RCL.values[0]

Ld3 = query1v - query27v
Ld4 = 0 - query2v
Ld5 = query3v
Ld6 = query4v
Ld7 = query5v
Ld8 = query6v - query25v

Ld10 = 0
Ld11 = query7v
Ld12 = query8v - query11v
Ld13 = query9v
Ld14 = query10v

Ld17 = query12v
Ld18 = query13v
Ld19 = 0

Ld22 = query14v
Ld23 = query15v + query16v - query14v - query17v
Ld24 = 0

Ld26 = query18v
Ld27 = query19v
Ld28 = query20v
Ld29 = query21v - query18v - query19v - query20v
Ld30 = query22v

Ld32 = 0
Ld33 = query23v
Ld34 = query24v
Ld35 = 0 - query26v
Ld36 = 0
Ld37 = 0
Ld38 = 0
Ld39 = 0
Ld40 = 0

Ld2 = Ld3 + Ld4 + Ld5 + Ld6 + Ld7
Ld9 = Ld10 + Ld11
Ld1 = Ld2 + Ld8 + Ld9 + Ld12 + Ld13 + Ld14
Ld25 = Ld26 + Ld27 + Ld28 + Ld29 + Ld30
Ld21 = Ld22 + Ld23 + Ld24
Ld31 = Ld32 + Ld33 + Ld34
Ld20 = Ld21 + Ld25 + Ld31
Ld16 = Ld17 + Ld18 + Ld19
```

```
Ld15 = Ld16 + Ld20 + Ld35
```

```
dic_detalha[i2] = [Ld1,Ld2,Ld3,Ld4,Ld5,Ld6,Ld7,Ld8,Ld9,Ld10,Ld11,Ld12,Ld13,Ld14,Ld15,Ld16,Ld17,Ld18,Ld19,Ld20,Ld21,Ld22,Ld23,Ld24,Ld25,Ld26,Ld27,Ld28,Ld29,Ld30,Ld31,Ld32,Ld33,Ld34,Ld35,Ld36,Ld37,Ld38,Ld39,Ld40]  
df_dcl_detalha = pd.DataFrame(dic_detalha)
```

In [568]:

```
### Definição de nome para as linhas do df_dcl_detalha, criado acima, e exportação para excel
```

```
linhas = ['DÍVIDA CONSOLIDADA - DC (I)',  
         ' Dívida Mobiliária',  
         ' Dívida Mobiliária do TN Interna (em mercado)',  
         ' (-) Aplicações em Títulos Públicos',  
         ' Dívida Mobiliária do TN Interna (em carteira BCB)',  
         ' Dívida Securitizada ',  
         ' Dívida Mobiliária Externa',  
         ' Operações de Equalização Cambial - Relacionamento TN/ BCB (Lei nº 11.803/08) ',  
         'Dívida Contratual',  
         ' Dívida Contratual de PPP',  
         ' Demais Dívidas Contratuais',  
         ' Precatórios posteriores a 5.5.2000 (inclusive)',  
         ' Dívida Assumida pela União (Lei nº 8.727/93)',  
         ' Passivos reconhecidos com insuficiência de créditos / recursos',  
         'DEDUÇÕES (II)',  
         ' Ativo Disponível',  
         ' Depósitos do TN no BCB',  
         ' Depósitos à Vista',  
         ' Arrecadação a Recolher',  
         ' Haveres Financeiros',  
         ' Aplicações Financeiras',  
         ' Disponibilidades do FAT',  
         ' Aplicações de Fundos Diversos Junto ao Setor Privado *',  
         ' Recursos da Reserva Monetária',  
         ' Renegociação de Dívidas de Entes da Federação',  
         ' Dívida Renegociada Estados e Municípios (Lei nº 9.496/97 e MP nº 2.185/01) ',  
         'Créditos da Lei nº 8.727/93',  
         ' Dívida Externa Renegociada (Aviso MF nº 30 e outros)',  
         ' Demais Dívidas Renegociadas',  
         ' Ajustes para Perdas',  
         ' Demais Ativos Financeiros',  
         ' Haveres Externos (Garantias)',  
         ' Outros Créditos Bancários',  
         ' Ajustes para Perdas',  
         ' (-) Restos a Pagar Processados1',  
         'DÍVIDA CONSOLIDADA LÍQUIDA (DCL) (III) = (I - II)',  
         'RECEITA CORRENTE LÍQUIDA - RCL',  
         '% da DC sobre a RCL (I/RCL)',  
         '% da DCL sobre a RCL (III/RCL)',  
         'LIMITE DEFINIDO POR RESOLUÇÃO DO SENADO FEDERAL - <%>2',]  
df_dcl_detalha.index = linhas  
df_dcl_detalha.to_excel('df_dcl_detalha.xlsx')  
df_dcl_detalha.to_excel('df_dcl_detalha.xlsx')
```

In [569]:

```
### Criação de dicionário para o demonstrativo sintético da DCL
###tava for i in range (1,18) - excluir comentário
dic = {}
for i2 in ['ano_anterior','quad1','quad2','quad3']:
    for i in range(1,28):
        globals()[f'query{i}v'] = float(globals()[f'df_query{i}_{i2}']['SDCC'][0]/1000)
        if i2=='ano_anterior':
            globals()[f'rc1_{i2}'] = df_rc1[(df_rc1['ANO']==globals()[f'{i2}']) & (df_rc1['QUAD']==3)].RCL.values[0]
        else:
            globals()[f'rc1_{i2}'] = df_rc1[(df_rc1['ANO']==ano_atual) & (df_rc1['QUAD']==globals()[f'{i2}'])].RCL.values[0]

L2 = query1v - query2v + query3v + query4v + query5v - query27v
L3 = query6v - query25v
L4 = query7v
L5 = query8v - query11v
L6 = query9v + query10v
L7 = 0

L9 = query12v + query13v
L10 = query15v + query16v + query21v + query22v + query23v + query24v - query17v
L11 = -query26v

L13 = globals()[f'rc1_{i2}']

L1 = L2 + L3 + L4 + L5 + L6
L8 = L9 + L10 + L11
L12 = L1 - L8
L14 = L1 / L13 * 100
L15 = L12 / L13 * 100
L16 = L2 / L13 * 100

dic[i2] = [L1,L2,L3,L4,L5,L6,L7,L8,L9,L10,L11,L12,L13,L14,L15,L16]
df_dcl = pd.DataFrame(dic)
df_dcl
```



Out[569]:

	ano_anterior	quad1	quad2	quad3
0	5.764561e+09	5.781968e+09	5.943382e+09	6.200607e+09
1	5.684015e+09	5.735956e+09	5.893481e+09	6.146316e+09
2	2.718639e+07	1.049927e+06	5.618827e+06	6.705687e+06
3	4.455271e+07	3.840910e+07	3.845097e+07	4.049280e+07
4	4.115666e+05	2.614105e+06	1.963128e+06	5.570300e+05
5	8.395601e+06	3.938972e+06	3.867575e+06	6.535525e+06
6	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00
7	2.579392e+09	2.470974e+09	2.456118e+09	2.616523e+09
8	1.276823e+09	1.173015e+09	1.217631e+09	1.438785e+09
9	1.363020e+09	1.363679e+09	1.324122e+09	1.251837e+09
10	-6.045089e+07	-6.571925e+07	-8.563492e+07	-7.409905e+07
11	3.185168e+09	3.310994e+09	3.487263e+09	3.584084e+09
12	8.053484e+08	8.186166e+08	8.291573e+08	9.056586e+08
13	7.157847e+02	7.063097e+02	7.167978e+02	6.846517e+02
14	3.955019e+02	4.044622e+02	4.205792e+02	3.957434e+02
15	7.057833e+02	7.006889e+02	7.107796e+02	6.786571e+02

In [570]:

```
### Criação de colunas referente às variações entre períodos
df_dcl['var1'] = ((df_dcl['quad1'] / df_dcl['ano_anterior'])-1)*100
df_dcl['var2'] = ((df_dcl['quad2'] / df_dcl['quad1'])-1)*100
df_dcl['var3'] = ((df_dcl['quad3'] / df_dcl['quad2'])-1)*100

### Reordenamento das colunas
df_dcl = df_dcl[['ano_anterior', 'quad1', 'var1', 'quad2', 'var2', 'quad3', 'var3']]

### Atribuição de 0 (zero) sempre que o resultado retornar erro
df_dcl.fillna(0,inplace = True)
```

In [571]:

```
### Delimitação da quantidade de colunas úteis de acordo com o quadrimestre atual
df_dcl=df_dcl[df_dcl.columns[:1+quadrimestre*2]]
```

In [572]:

```
### Visualização do df_dcl
df_dcl
```

Out[572]:

	ano_anterior	quad1	var1	quad2	var2	qua
0	5.764561e+09	5.781968e+09	0.301972	5.943382e+09	2.791667	6.200607e+
1	5.684015e+09	5.735956e+09	0.913817	5.893481e+09	2.746271	6.146316e+
2	2.718639e+07	1.049927e+06	-96.138040	5.618827e+06	435.163456	6.705687e+
3	4.455271e+07	3.840910e+07	-13.789531	3.845097e+07	0.109007	4.049280e+
4	4.115666e+05	2.614105e+06	535.159775	1.963128e+06	-24.902489	5.570300e+
5	8.395601e+06	3.938972e+06	-53.082898	3.867575e+06	-1.812595	6.535525e+
6	0.000000e+00	0.000000e+00	0.000000	0.000000e+00	0.000000	0.000000e+
7	2.579392e+09	2.470974e+09	-4.203263	2.456118e+09	-0.601194	2.616523e+
8	1.276823e+09	1.173015e+09	-8.130227	1.217631e+09	3.803580	1.438785e+
9	1.363020e+09	1.363679e+09	0.048298	1.324122e+09	-2.900700	1.251837e+
10	-6.045089e+07	-6.571925e+07	8.715103	-8.563492e+07	30.304167	-7.409905e-
11	3.185168e+09	3.310994e+09	3.950372	3.487263e+09	5.323737	3.584084e+
12	8.053484e+08	8.186166e+08	1.647513	8.291573e+08	1.287620	9.056586e+
13	7.157847e+02	7.063097e+02	-1.323732	7.167978e+02	1.484927	6.846517e+
14	3.955019e+02	4.044622e+02	2.265535	4.205792e+02	3.984808	3.957434e+
15	7.057833e+02	7.006889e+02	-0.721804	7.107796e+02	1.440108	6.786571e+

In [573]:

```
### Criação de dicionário referente ao template no formato em que "L" é a linha e "C" é
a coluna de cada campo da tabela
### Exemplo, L1C1 é a linha 1 e coluna 1, L5C3 é a linha 5 da coluna 3
### Formatação de valores de acordo com o tipo de cada coluna (colunas de percentuais d
a tabela acima com 2 casas decimais)
template_dict = {}
df_dcl = df_dcl.reset_index()
df_dcl.drop('index',axis=1,inplace=True)
for l in df_dcl.index:
    for c in range(0,df_dcl.columns.size):
        if c > 0 and c % 2==0:
            template_dict['L'+str(l+1)+'_C'+str(c+1)]=str(locale.currency(df_dcl.iloc[l
][c] ,grouping=True,symbol=False ))
        else:
            template_dict['L'+str(l+1)+'_C'+str(c+1)]=str(locale.currency(round(df_dcl.
iloc[l][c]) ,grouping=True,symbol=False ))[:-3]
```

In [574]:

```
### Definição de nomes para as linhas do demonstrativo sintético da DCL e exportação para excel
```

```
linhas = ['DÍVIDA CONSOLIDADA - DC (I)',  
         'Dívida Mobiliária do Tesouro Nacional',  
         'Oper de Eq Camb - Rel TN/BCB',  
         'Dívida Contratual',  
         'Precatórios posteriores a 5/5/2000',  
         'Dív Assumida pela União (Lei 8.727/93)',  
         'Pass reconh insuf créd/recursos',  
         'DEDUÇÕES (II)',  
         'Ativo Disponível',  
         'Haveres Financeiros',  
         '(-) Restos a Pagar Processados*',  
         'DÍV CONS LÍQ-DCL (III) = (I - II)',  
         'RECEITA CORR LÍQUIDA - RCL',  
         '% da DC sobre a RCL',  
         '% da DCL sobre a RCL',  
         '% da DM* sobre a RCL (DM*/ RCL)']
```


```
df_dcl.index = linhas
```

```
df_dcl.to_excel('df_dcl.xlsx')
```

```
df_dcl
```

Out[574]:

	ano_anterior	quad1	var1	quad2	var2
<b>DÍVIDA CONSOLIDADA - DC (I)</b>	5.764561e+09	5.781968e+09	0.301972	5.943382e+09	2.791667
Dívida Mobiliária do Tesouro Nacional	5.684015e+09	5.735956e+09	0.913817	5.893481e+09	2.746271
Oper de Eq Camb - Rel TN/BCB	2.718639e+07	1.049927e+06	-96.138040	5.618827e+06	435.163451
Dívida Contratual	4.455271e+07	3.840910e+07	-13.789531	3.845097e+07	0.109007
Precatórios posteriores a 5/5/2000	4.115666e+05	2.614105e+06	535.159775	1.963128e+06	-24.902489
Dív Assumida pela União (Lei 8.727/93)	8.395601e+06	3.938972e+06	-53.082898	3.867575e+06	-1.812595
Pass reconh insuf créd/recursos	0.000000e+00	0.000000e+00	0.000000	0.000000e+00	0.000000
<b>DEDUÇÕES (II)</b>	2.579392e+09	2.470974e+09	-4.203263	2.456118e+09	-0.601194
Ativo Disponível	1.276823e+09	1.173015e+09	-8.130227	1.217631e+09	3.803580
Haveres Financeiros	1.363020e+09	1.363679e+09	0.048298	1.324122e+09	-2.900700
(-) Restos a Pagar Processados*	-6.045089e+07	-6.571925e+07	8.715103	-8.563492e+07	30.304167
<b>DÍV CONS LÍQ-DCL (III) = (I - II)</b>	3.185168e+09	3.310994e+09	3.950372	3.487263e+09	5.323737
<b>RECEITA CORR LÍQUIDA - RCL</b>	8.053484e+08	8.186166e+08	1.647513	8.291573e+08	1.287620
% da DC sobre a RCL	7.157847e+02	7.063097e+02	-1.323732	7.167978e+02	1.484927
% da DCL sobre a RCL	3.955019e+02	4.044622e+02	2.265535	4.205792e+02	3.984808
% da DM* sobre a RCL (DM*/ RCL)	7.057833e+02	7.006889e+02	-0.721804	7.107796e+02	1.440108



In [575]:

```
### Comando auxiliar para possibilitar operações aritméticas com o quadrimestre anterior no código seguinte
if quadrimestre == 1:
    aux1 = -3
else:
    aux1 = -4
```

In [576]:

```
### Criação de df_assistido, cuja função é alimentar os parágrafos da minuta de instrução assistida
### Cada "Valores" é uma coluna
valores1 = []
valores2 = []
valores3 = []
valores4 = []
valores5 = []
valores6 = []
for x in df_dc1.reset_index().index:

    valores1.append(df_dc1.loc[df_dc1.index[x],df_dc1.columns[-2]])

    valores2.append(df_dc1.loc[df_dc1.index[x],df_dc1.columns[aux1]])

    valores3.append(df_dc1.loc[df_dc1.index[x],df_dc1.columns[-2]]-df_dc1.loc[df_dc1.index[x],df_dc1.columns[aux1]])

    valores4.append(df_dc1.loc[df_dc1.index[x],df_dc1.columns[-1]])

    if df_dc1.loc[df_dc1.index[x],df_dc1.columns[-1]]==0:
        valores5.append('permaneceu inalterado')
    else:
        if df_dc1.loc[df_dc1.index[x],df_dc1.columns[-1]]>0:
            valores5.append('aumentou')
        else:
            valores5.append('diminuiu')

    if df_dc1.loc[df_dc1.index[x],df_dc1.columns[-1]]==0:
        valores6.append('manutenção de valores')
    else:
        if df_dc1.loc[df_dc1.index[x],df_dc1.columns[-1]]>0:
            valores6.append('crescimento de')
        else:
            valores6.append('redução de')

df_assistido = pd.DataFrame({
                                'quad_atual': valores1,
                                'quad_anterior':valores2,
                                'Variação em Reais': valores3,
                                'Variação percentual': valores4,
                                'verbo':valores5,
                                'substantivo':valores6          })

df_assistido.index = linhas
df_assistido
```

Out[576]:

	quad_atual	quad_anterior	Variação em Reais	Variação percentual	verbo
<b>DÍVIDA CONSOLIDADA - DC (I)</b>	6.200607e+09	5.943382e+09	2.572258e+08	4.327936	aumentou
<b>Dívida Mobiliária do Tesouro Nacional</b>	6.146316e+09	5.893481e+09	2.528352e+08	4.290083	aumentou
<b>Oper de Eq Camb - Rel TN/BCB</b>	6.705687e+06	5.618827e+06	1.086859e+06	19.343171	aumentou
<b>Dívida Contratual</b>	4.049280e+07	3.845097e+07	2.041835e+06	5.310230	aumentou
<b>Precatórios posteriores a 5/5/2000</b>	5.570300e+05	1.963128e+06	-1.406098e+06	-71.625388	diminuiu
<b>Dív Assumida pela União (Lei 8.727/93)</b>	6.535525e+06	3.867575e+06	2.667950e+06	68.982514	aumentou
<b>Pass reconh insuf créd/recursos</b>	0.000000e+00	0.000000e+00	0.000000e+00	0.000000	permanece inalterado
<b>DEDUÇÕES (II)</b>	2.616523e+09	2.456118e+09	1.604050e+08	6.530832	aumentou
<b>Ativo Disponível</b>	1.438785e+09	1.217631e+09	2.211543e+08	18.162667	aumentou
<b>Haveres Financeiros</b>	1.251837e+09	1.324122e+09	-7.228518e+07	-5.459102	diminuiu
<b>(-) Restos a Pagar Processados*</b>	-7.409905e+07	-8.563492e+07	1.153586e+07	-13.470979	diminuiu
<b>DÍV CONS LÍQ-DCL (III) = (I - II)</b>	3.584084e+09	3.487263e+09	9.682079e+07	2.776412	aumentou
<b>RECEITA CORR LÍQUIDA - RCL</b>	9.056586e+08	8.291573e+08	7.650130e+07	9.226391	aumentou
<b>% da DC sobre a RCL</b>	6.846517e+02	7.167978e+02	-3.214610e+01	-4.484681	diminuiu
<b>% da DCL sobre a RCL</b>	3.957434e+02	4.205792e+02	-2.483582e+01	-5.905148	diminuiu

	quad_atual	quad_anterior	Variação em Reais	Variação percentual	verbo
% da DM* sobre a RCL (DM*/ RCL)	6.786571e+02	7.107796e+02	-3.212253e+01	-4.519337	diminuiu



In [577]:

```
### Retirada de sinal negativo na 3ª coluna do df_assistido, para uso na minuta de instrução assistida
df_assistido['Variação em Reais'] = df_assistido['Variação em Reais'].apply(lambda x: abs(x))
```



In [578]:

```
### Visualização do df_assistido  
df_assistido
```

Out[578]:

	quad_atual	quad_anterior	Varição em Reais	Varição percentual	verbo
<b>DÍVIDA CONSOLIDADA - DC (I)</b>	6.200607e+09	5.943382e+09	2.572258e+08	4.327936	aumentou
<b>Dívida Mobiliária do Tesouro Nacional</b>	6.146316e+09	5.893481e+09	2.528352e+08	4.290083	aumentou
<b>Oper de Eq Camb - Rel TN/BCB</b>	6.705687e+06	5.618827e+06	1.086859e+06	19.343171	aumentou
<b>Dívida Contratual</b>	4.049280e+07	3.845097e+07	2.041835e+06	5.310230	aumentou
<b>Precatórios posteriores a 5/5/2000</b>	5.570300e+05	1.963128e+06	1.406098e+06	-71.625388	diminuiu
<b>Dív Assumida pela União (Lei 8.727/93)</b>	6.535525e+06	3.867575e+06	2.667950e+06	68.982514	aumentou
<b>Pass reconh insuf créd/recursos</b>	0.000000e+00	0.000000e+00	0.000000e+00	0.000000	permaneceu inalterado
<b>DEDUÇÕES (II)</b>	2.616523e+09	2.456118e+09	1.604050e+08	6.530832	aumentou
<b>Ativo Disponível</b>	1.438785e+09	1.217631e+09	2.211543e+08	18.162667	aumentou
<b>Haveres Financeiros</b>	1.251837e+09	1.324122e+09	7.228518e+07	-5.459102	diminuiu
<b>(-) Restos a Pagar Processados*</b>	-7.409905e+07	-8.563492e+07	1.153586e+07	-13.470979	diminuiu
<b>DÍV CONS LÍQ-DCL (III) = (I - II)</b>	3.584084e+09	3.487263e+09	9.682079e+07	2.776412	aumentou
<b>RECEITA CORR LÍQUIDA - RCL</b>	9.056586e+08	8.291573e+08	7.650130e+07	9.226391	aumentou
<b>% da DC sobre a RCL</b>	6.846517e+02	7.167978e+02	3.214610e+01	-4.484681	diminuiu
<b>% da DCL sobre a RCL</b>	3.957434e+02	4.205792e+02	2.483582e+01	-5.905148	diminuiu

	quad_atual	quad_anterior	Variação em Reais	Variação percentual	verbo
% da DM* sobre a RCL (DM*/ RCL)	6.786571e+02	7.107796e+02	3.212253e+01	-4.519337	diminuiu

In [579]:

```
df_assistido = df_assistido.reset_index(drop = True)
df_assistido
```

Out[579]:

	quad_atual	quad_anterior	Variação em Reais	Variação percentual	verbo	substantivo
0	6.200607e+09	5.943382e+09	2.572258e+08	4.327936	aumentou	crescimento de
1	6.146316e+09	5.893481e+09	2.528352e+08	4.290083	aumentou	crescimento de
2	6.705687e+06	5.618827e+06	1.086859e+06	19.343171	aumentou	crescimento de
3	4.049280e+07	3.845097e+07	2.041835e+06	5.310230	aumentou	crescimento de
4	5.570300e+05	1.963128e+06	1.406098e+06	-71.625388	diminuiu	redução de
5	6.535525e+06	3.867575e+06	2.667950e+06	68.982514	aumentou	crescimento de
6	0.000000e+00	0.000000e+00	0.000000e+00	0.000000	permaneceu inalterado	manutenção de valores
7	2.616523e+09	2.456118e+09	1.604050e+08	6.530832	aumentou	crescimento de
8	1.438785e+09	1.217631e+09	2.211543e+08	18.162667	aumentou	crescimento de
9	1.251837e+09	1.324122e+09	7.228518e+07	-5.459102	diminuiu	redução de
10	-7.409905e+07	-8.563492e+07	1.153586e+07	-13.470979	diminuiu	redução de
11	3.584084e+09	3.487263e+09	9.682079e+07	2.776412	aumentou	crescimento de
12	9.056586e+08	8.291573e+08	7.650130e+07	9.226391	aumentou	crescimento de
13	6.846517e+02	7.167978e+02	3.214610e+01	-4.484681	diminuiu	redução de
14	3.957434e+02	4.205792e+02	2.483582e+01	-5.905148	diminuiu	redução de
15	6.786571e+02	7.107796e+02	3.212253e+01	-4.519337	diminuiu	redução de

In [580]:

```
### Criação de dicionário para alimentação da minuta de instrução assistida
### Z1_1 é a linha 1 e coluna 1, Z5_3 é a linha 5 da coluna 3
### Formatação dos valores da tabela acima.
### As 3 primeiras colunas são números, a quarta é percentual, as duas últimas são text
o.

template_dict_z = {}

strMilhao = ''
strReais = 'R$ '
for l in df_assistido.index:
    if l < len(df_assistido)-3:
        for c in range(0,df_assistido.columns.size):
            strMilhao=''
            if c < 3:
                if round(abs(df_assistido.iloc[l][c])/1000000 >= 2:
                    strMilhao = ' bilhões'
                    template_dict_z['Z'+str(l+1)+'_'+str(c+1)]=strReais + str(locale.cu
rrency(round(df_assistido.iloc[l][c]/1000000,2) ,grouping=True,symbol=False )) + strMil
hao
                else:
                    if round(abs(df_assistido.iloc[l][c])/1000000 >= 1:
                        strMilhao = ' bilhão'
                        template_dict_z['Z'+str(l+1)+'_'+str(c+1)]=strReais + str(local
e.currency(round(df_assistido.iloc[l][c]/1000000,2) ,grouping=True,symbol=False )) + st
rMilhao
                    else:
                        if round(abs(df_assistido.iloc[l][c])/1000 >= 2:
                            strMilhao = ' milhões'
                            template_dict_z['Z'+str(l+1)+'_'+str(c+1)]=strReais + str(l
ocale.currency(round(df_assistido.iloc[l][c]/1000,2) ,grouping=True,symbol=False )) + s
trMilhao
                        else:
                            if round(abs(df_assistido.iloc[l][c])/1000 >= 1:
                                strMilhao = ' milhão'
                                template_dict_z['Z'+str(l+1)+'_'+str(c+1)]=strReais + s
tr(locale.currency(round(df_assistido.iloc[l][c]/1000,2) ,grouping=True,symbol=False ))
                                + strMilhao
                            else:
                                if abs(df_assistido.iloc[l][c])==0:
                                    template_dict_z['Z'+str(l+1)+'_'+str(c+1)]= ''
                                else:
                                    template_dict_z['Z'+str(l+1)+'_'+str(c+1)]=strReais
+ str(locale.currency(round(df_assistido.iloc[l][c]/1000,2) ,grouping=True,symbol=False
)) + strMilhao

            if c == 3:
                if abs(df_assistido.iloc[l][c])==0:
                    template_dict_z['Z'+str(l+1)+'_'+str(c+1)]=''
                else:
                    template_dict_z['Z'+str(l+1)+'_'+str(c+1)]=str(round(df_assistido.i
loc[l][c],2)).replace('.',',')
            if c > 3:
                template_dict_z['Z'+str(l+1)+'_'+str(c+1)]=df_assistido.iloc[l][c]
        else:
            for c in range(0,df_assistido.columns.size):
                if c < 3:
                    template_dict_z['Z'+str(l+1)+'_'+str(c+1)]=str(locale.currency(round(df
```

```

_assistido.iloc[1][c],2) ,grouping=True,symbol=False ))
    else:
        if c == 3:
            if abs(df_assistido.iloc[1][c])==0:
                template_dict_z['Z'+str(l+1)+'_'+str(c+1)]=''
            else:
                template_dict_z['Z'+str(l+1)+'_'+str(c+1)]=str(round(df_assistido.iloc[1][c],2)).replace('.',',')
        else:
            template_dict_z['Z'+str(l+1)+'_'+str(c+1)]=df_assistido.iloc[1][c]

```

In [581]:

```
template_dict_z['Z15_1']
```

Out[581]:

```
'395,74'
```

In [582]:

```

### Texto para conclusão da análise, a partir do resultado da DCL (Z14_1)
if df_assistido.iloc[14][0] < (350 *0.9):
    Z_dcl_conclusao = str('O percentual da DCL em relação à RCL foi de '+str(template_dict_z['Z15_1'])+', portanto, inferior a 90% do limite de referência.')
else:
    if df_assistido.iloc[14][0] <= 350:
        Z_dcl_conclusao = str('O percentual da DCL em relação à RCL foi de '+str(template_dict_z['Z15_1'])+', portanto, ficou entre 90 e 100% do limite de referência. Nessa situação, em atenção ao disposto no art. 59, § 1º da LRF, propõe-se a emissão de alerta ao Poder Executivo, registrando que, caso o limite seja efetivamente ultrapassado, aplicam-se as medidas do art. 31 da LRF no tocante à recondução aos limites e às vedações impostas.')
    else:
        Z_dcl_conclusao = str('O percentual da DCL em relação à RCL foi de '+str(template_dict_z['Z15_1'])+', portanto, superior ao limite de referência. Em virtude disso, propõe-se determinar ao Poder Executivo que adote as medidas do art. 31 da LRF no tocante à recondução aos limites e às vedações impostas.')
Z_dcl_conclusao

```

Out[582]:

```
'O percentual da DCL em relação à RCL foi de 395,74, portanto, superior ao limite de referência. Em virtude disso, propõe-se determinar ao Poder Executivo que adote as medidas do art. 31 da LRF no tocante à recondução aos limites e às vedações impostas.'
```

In [583]:

```

### Definição de outros campos do template.
template_dict['ano_anterior']=(ano_atual -1)
template_dict['ano_atual']=(ano_atual)
template_dict['quadrimestre']=(quadrimestre)
template_dict['Z_dcl_conclusao']=(Z_dcl_conclusao)

```

In [584]:

```
### Atualização do template para inserir os campos do tipo "Z".  
template_dict.update(template_dict_z)  
  
### Visualização da lista referente ao template_dict atualizado.  
template_dict
```

Out[584]:

```
{ 'L1_C1': '5.764.560.886',  
  'L1_C2': '5.781.968.233',  
  'L1_C3': '0,30',  
  'L1_C4': '5.943.381.534',  
  'L1_C5': '2,79',  
  'L1_C6': '6.200.607.290',  
  'L1_C7': '4,33',  
  'L2_C1': '5.684.014.621',  
  'L2_C2': '5.735.956.128',  
  'L2_C3': '0,91',  
  'L2_C4': '5.893.481.036',  
  'L2_C5': '2,75',  
  'L2_C6': '6.146.316.245',  
  'L2_C7': '4,29',  
  'L3_C1': '27.186.388',  
  'L3_C2': '1.049.927',  
  'L3_C3': '-96,14',  
  'L3_C4': '5.618.827',  
  'L3_C5': '435,16',  
  'L3_C6': '6.705.687',  
  'L3_C7': '19,34',  
  'L4_C1': '44.552.710',  
  'L4_C2': '38.409.100',  
  'L4_C3': '-13,79',  
  'L4_C4': '38.450.968',  
  'L4_C5': '0,11',  
  'L4_C6': '40.492.803',  
  'L4_C7': '5,31',  
  'L5_C1': '411.567',  
  'L5_C2': '2.614.105',  
  'L5_C3': '535,16',  
  'L5_C4': '1.963.128',  
  'L5_C5': '-24,90',  
  'L5_C6': '557.030',  
  'L5_C7': '-71,63',  
  'L6_C1': '8.395.601',  
  'L6_C2': '3.938.972',  
  'L6_C3': '-53,08',  
  'L6_C4': '3.867.575',  
  'L6_C5': '-1,81',  
  'L6_C6': '6.535.525',  
  'L6_C7': '68,98',  
  'L7_C1': '0',  
  'L7_C2': '0',  
  'L7_C3': '0,00',  
  'L7_C4': '0',  
  'L7_C5': '0,00',  
  'L7_C6': '0',  
  'L7_C7': '0,00',  
  'L8_C1': '2.579.392.494',  
  'L8_C2': '2.470.973.830',  
  'L8_C3': '-4,20',  
  'L8_C4': '2.456.118.483',  
  'L8_C5': '-0,60',  
  'L8_C6': '2.616.523.454',  
  'L8_C7': '6,53',  
  'L9_C1': '1.276.823.195',  
  'L9_C2': '1.173.014.572',  
  'L9_C3': '-8,13',
```

'L9\_C4': '1.217.631.121',  
'L9\_C5': '3,80',  
'L9\_C6': '1.438.785.413',  
'L9\_C7': '18,16',  
'L10\_C1': '1.363.020.187',  
'L10\_C2': '1.363.678.504',  
'L10\_C3': '0,05',  
'L10\_C4': '1.324.122.277',  
'L10\_C5': '-2,90',  
'L10\_C6': '1.251.837.095',  
'L10\_C7': '-5,46',  
'L11\_C1': '-60.450.888',  
'L11\_C2': '-65.719.246',  
'L11\_C3': '8,72',  
'L11\_C4': '-85.634.915',  
'L11\_C5': '30,30',  
'L11\_C6': '-74.099.054',  
'L11\_C7': '-13,47',  
'L12\_C1': '3.185.168.392',  
'L12\_C2': '3.310.994.402',  
'L12\_C3': '3,95',  
'L12\_C4': '3.487.263.051',  
'L12\_C5': '5,32',  
'L12\_C6': '3.584.083.837',  
'L12\_C7': '2,78',  
'L13\_C1': '805.348.403',  
'L13\_C2': '818.616.620',  
'L13\_C3': '1,65',  
'L13\_C4': '829.157.292',  
'L13\_C5': '1,29',  
'L13\_C6': '905.658.590',  
'L13\_C7': '9,23',  
'L14\_C1': '716',  
'L14\_C2': '706',  
'L14\_C3': '-1,32',  
'L14\_C4': '717',  
'L14\_C5': '1,48',  
'L14\_C6': '685',  
'L14\_C7': '-4,48',  
'L15\_C1': '396',  
'L15\_C2': '404',  
'L15\_C3': '2,27',  
'L15\_C4': '421',  
'L15\_C5': '3,98',  
'L15\_C6': '396',  
'L15\_C7': '-5,91',  
'L16\_C1': '706',  
'L16\_C2': '701',  
'L16\_C3': '-0,72',  
'L16\_C4': '711',  
'L16\_C5': '1,44',  
'L16\_C6': '679',  
'L16\_C7': '-4,52',  
'ano\_anterior': 2018,  
'ano\_atual': 2019,  
'quadrimestre': 3,  
'Z\_dcl\_conclusao': 'O percentual da DCL em relação à RCL foi de 395,74, portanto, superior ao limite de referência. Em virtude disso, propõe-se determinar ao Poder Executivo que adote as medidas do art. 31 da LRF no tocante à recondução aos limites e às vedações impostas.',  
'Z1\_1': 'R\$ 6.200,61 bilhões',



'Z1\_2': 'R\$ 5.943,38 bilhões',  
'Z1\_3': 'R\$ 257,23 bilhões',  
'Z1\_4': '4,33',  
'Z1\_5': 'aumentou',  
'Z1\_6': 'crescimento de',  
'Z2\_1': 'R\$ 6.146,32 bilhões',  
'Z2\_2': 'R\$ 5.893,48 bilhões',  
'Z2\_3': 'R\$ 252,84 bilhões',  
'Z2\_4': '4,29',  
'Z2\_5': 'aumentou',  
'Z2\_6': 'crescimento de',  
'Z3\_1': 'R\$ 6,71 bilhões',  
'Z3\_2': 'R\$ 5,62 bilhões',  
'Z3\_3': 'R\$ 1,09 bilhão',  
'Z3\_4': '19,34',  
'Z3\_5': 'aumentou',  
'Z3\_6': 'crescimento de',  
'Z4\_1': 'R\$ 40,49 bilhões',  
'Z4\_2': 'R\$ 38,45 bilhões',  
'Z4\_3': 'R\$ 2,04 bilhões',  
'Z4\_4': '5,31',  
'Z4\_5': 'aumentou',  
'Z4\_6': 'crescimento de',  
'Z5\_1': 'R\$ 557,03 milhões',  
'Z5\_2': 'R\$ 1,96 bilhão',  
'Z5\_3': 'R\$ 1,41 bilhão',  
'Z5\_4': '-71,63',  
'Z5\_5': 'diminuiu',  
'Z5\_6': 'redução de',  
'Z6\_1': 'R\$ 6,54 bilhões',  
'Z6\_2': 'R\$ 3,87 bilhões',  
'Z6\_3': 'R\$ 2,67 bilhões',  
'Z6\_4': '68,98',  
'Z6\_5': 'aumentou',  
'Z6\_6': 'crescimento de',  
'Z7\_1': '',  
'Z7\_2': '',  
'Z7\_3': '',  
'Z7\_4': '',  
'Z7\_5': 'permaneceu inalterado',  
'Z7\_6': 'manutenção de valores',  
'Z8\_1': 'R\$ 2.616,52 bilhões',  
'Z8\_2': 'R\$ 2.456,12 bilhões',  
'Z8\_3': 'R\$ 160,40 bilhões',  
'Z8\_4': '6,53',  
'Z8\_5': 'aumentou',  
'Z8\_6': 'crescimento de',  
'Z9\_1': 'R\$ 1.438,79 bilhões',  
'Z9\_2': 'R\$ 1.217,63 bilhões',  
'Z9\_3': 'R\$ 221,15 bilhões',  
'Z9\_4': '18,16',  
'Z9\_5': 'aumentou',  
'Z9\_6': 'crescimento de',  
'Z10\_1': 'R\$ 1.251,84 bilhões',  
'Z10\_2': 'R\$ 1.324,12 bilhões',  
'Z10\_3': 'R\$ 72,29 bilhões',  
'Z10\_4': '-5,46',  
'Z10\_5': 'diminuiu',  
'Z10\_6': 'redução de',  
'Z11\_1': 'R\$ -74,10 bilhões',  
'Z11\_2': 'R\$ -85,63 bilhões',

```
'Z11_3': 'R$ 11,54 bilhões',
'Z11_4': '-13,47',
'Z11_5': 'diminuiu',
'Z11_6': 'redução de',
'Z12_1': 'R$ 3.584,08 bilhões',
'Z12_2': 'R$ 3.487,26 bilhões',
'Z12_3': 'R$ 96,82 bilhões',
'Z12_4': '2,78',
'Z12_5': 'aumentou',
'Z12_6': 'crescimento de',
'Z13_1': 'R$ 905,66 bilhões',
'Z13_2': 'R$ 829,16 bilhões',
'Z13_3': 'R$ 76,50 bilhões',
'Z13_4': '9,23',
'Z13_5': 'aumentou',
'Z13_6': 'crescimento de',
'Z14_1': '684,65',
'Z14_2': '716,80',
'Z14_3': '32,15',
'Z14_4': '-4,48',
'Z14_5': 'diminuiu',
'Z14_6': 'redução de',
'Z15_1': '395,74',
'Z15_2': '420,58',
'Z15_3': '24,84',
'Z15_4': '-5,91',
'Z15_5': 'diminuiu',
'Z15_6': 'redução de',
'Z16_1': '678,66',
'Z16_2': '710,78',
'Z16_3': '32,12',
'Z16_4': '-4,52',
'Z16_5': 'diminuiu',
'Z16_6': 'redução de'}
```

In [585]:

```
### Leitura do template relativo ao quadrimestre atual
doc_rcl = docxtpl.DocxTemplate("templates_rgf/template_rgf_dcl_"+str(quadrimestre)+".docx")
### Renderização do documento
doc_rcl.render(template_dict)
### Salvamento do o documento
doc_rcl.save("rgf_dcl.docx")
```

In [ ]:

```
ANEXO II - GARANTIAS E CONTRAGARANTIAS DE VALORES
```

In [1]:

```
### Importação de bibliotecas e definições iniciais
import docxtp1
import pandas as pd
import pyodbc
import locale

conn = pyodbc.connect('DSN=LABCONTAS')
quad_to_mes = {1:4,2:8,3:12}
locale.setlocale( locale.LC_MONETARY , 'pt_BR.UTF-8' )
'{} {}'.format('one', 'two')
```

Out[1]:

```
'one two'
```

In [2]:

```
### DEFINIÇÃO DO PERÍODO OBJETO DA ANÁLISE --- ALTERE AQUI!!!
ano_atual = 2019
quadrimestre = 3

### Definições fixas
ano_anterior = ano_atual - 1
quad1 = 1
quad2 = 2
quad3 = 3
```

In [3]:

```
### Queries de acordo com as consultas do sistema TesouroGerencial convertidas em SQL
```

```
###query_garantia_estados
```

```
query1 = """select sum((a11.VA_MOVIMENTO_LIQUIDO * a13.PE_TAXA)) SDCC
from BD_DWTG.DBO.WF_LANCAMENTO_EP20 a11
  join BD_DWTG.DBO.WD_MOEDA a12
    on (a11.ID_MOEDA_UG_EXEC_H = a12.ID_MOEDA)
  join BD_DWTG.DBO.WD_TAXA_CAMBIO_MENSAL a13
    on (a12.ID_MOEDA = a13.ID_MOEDA_ORIGEM)
  join BD_DWTG.DBO.WA_MES_ACUM a14
    on (a11.ID_ANO_LANC = a14.ID_ANO_ACUM_ANO_SALDO and
a11.ID_MES_LANC = a14.ID_MES_ACUM_ANO_SALDO)
  join BD_DWTG.DBO.WD_CONTA_CONTABIL_EXERCICIO a15
    on (a11.ID_ANO_LANC = a15.ID_ANO and
a11.ID_CONTA_CONTABIL_LANC = a15.ID_CONTA_CONTABIL)
  join BD_DWTG.DBO.WD_MES a16
    on (a14.ID_ANO = a16.ID_ANO and
a14.ID_MES = a16.ID_MES)
  join BD_DWTG.DBO.WD_CONTA_CORRENTE a17
    on (a11.ID_CONTA_CORRENTE_SK = a17.ID_CONTA_CORRENTE_SK)
where(a11.ID_ORCA_FISCAL_ORGAO_EXEC_H in (0)
and (a14.ID_ANO = {} and a14.ID_MES = {})
and a15.ID_CONTA_CONTABIL_DESTINO in (812110204, 812110104)
and a17.NR_CONTA_CORRENTE in ('CG0000064', 'CG0000069')
and a13.ID_ANO = a16.ID_ANO
and a13.ID_MES = a16.ID_MES)"""
```

```
###query_garantia_municipios
```

```
query2 = """select sum((a11.VA_MOVIMENTO_LIQUIDO * a13.PE_TAXA)) SDCC
from BD_DWTG.DBO.WF_LANCAMENTO_EP20 a11
  join BD_DWTG.DBO.WD_MOEDA a12
    on (a11.ID_MOEDA_UG_EXEC_H = a12.ID_MOEDA)
  join BD_DWTG.DBO.WD_TAXA_CAMBIO_MENSAL a13
    on (a12.ID_MOEDA = a13.ID_MOEDA_ORIGEM)
  join BD_DWTG.DBO.WA_MES_ACUM a14
    on (a11.ID_ANO_LANC = a14.ID_ANO_ACUM_ANO_SALDO and
a11.ID_MES_LANC = a14.ID_MES_ACUM_ANO_SALDO)
  join BD_DWTG.DBO.WD_CONTA_CONTABIL_EXERCICIO a15
    on (a11.ID_ANO_LANC = a15.ID_ANO and
a11.ID_CONTA_CONTABIL_LANC = a15.ID_CONTA_CONTABIL)
  join BD_DWTG.DBO.WD_MES a16
    on (a14.ID_ANO = a16.ID_ANO and
a14.ID_MES = a16.ID_MES)
  join BD_DWTG.DBO.WD_CONTA_CORRENTE a17
    on (a11.ID_CONTA_CORRENTE_SK = a17.ID_CONTA_CORRENTE_SK)
where (a11.ID_ORCA_FISCAL_ORGAO_EXEC_H in (0)
and (a14.ID_ANO = {} and a14.ID_MES = {})
and a15.ID_CONTA_CONTABIL_DESTINO in (812110204, 812110104)
and a17.NR_CONTA_CORRENTE in ('CG0000065', 'CG0000070')
and a13.ID_ANO = a16.ID_ANO
and a13.ID_MES = a16.ID_MES)"""
```

```
###query_garantia_entidades_controladas
```

```
query3 = """select sum((a11.VA_MOVIMENTO_LIQUIDO * a13.PE_TAXA)) SDCC
from BD_DWTG.DBO.WF_LANCAMENTO_EP20 a11
  join BD_DWTG.DBO.WD_MOEDA a12
```

```

    on      (a11.ID_MOEDA_UG_EXEC_H = a12.ID_MOEDA)
join      BD_DWTG.DBO.WD_TAXA_CAMBIO_MENSAL      a13
    on      (a12.ID_MOEDA = a13.ID_MOEDA_ORIGEM)
join      BD_DWTG.DBO.WA_MES_ACUM a14
    on      (a11.ID_ANO_LANC = a14.ID_ANO_ACUM_ANO_SALDO and
a11.ID_MES_LANC = a14.ID_MES_ACUM_ANO_SALDO)
join      BD_DWTG.DBO.WD_CONTA_CONTABIL_EXERCICIO a15
    on      (a11.ID_ANO_LANC = a15.ID_ANO and
a11.ID_CONTA_CONTABIL_LANC = a15.ID_CONTA_CONTABIL)
join      BD_DWTG.DBO.WD_MES      a16
    on      (a14.ID_ANO = a16.ID_ANO and
a14.ID_MES = a16.ID_MES)
join      BD_DWTG.DBO.WD_CONTA_CORRENTE      a17
    on      (a11.ID_CONTA_CORRENTE_SK = a17.ID_CONTA_CORRENTE_SK)
where     (a11.ID_ORCA_FISCAL_ORGAO_EXEC_H in (0)
and (a14.ID_ANO = {} and a14.ID_MES = {}))
and a15.ID_CONTA_CONTABIL_DESTINO in (812110204, 812110104)
and a17.NR_CONTA_CORRENTE in ('CG0000066', 'CG0000067', 'CG0000071', 'CG0000072')
and a13.ID_ANO = a16.ID_ANO
and a13.ID_MES = a16.ID_MES)""

```

### ###query\_garantia\_fundos\_e\_programas

```

query4 = ""select      sum((a11.VA_MOVIMENTO_LIQUIDO * a13.PE_TAXA))      SDCC
from      BD_DWTG.DBO.WF_LANCAMENTO_EP20      a11
join      BD_DWTG.DBO.WD_MOEDA      a12
    on      (a11.ID_MOEDA_UG_EXEC_H = a12.ID_MOEDA)
join      BD_DWTG.DBO.WD_TAXA_CAMBIO_MENSAL      a13
    on      (a12.ID_MOEDA = a13.ID_MOEDA_ORIGEM)
join      BD_DWTG.DBO.WA_MES_ACUM a14
    on      (a11.ID_ANO_LANC = a14.ID_ANO_ACUM_ANO_SALDO and
a11.ID_MES_LANC = a14.ID_MES_ACUM_ANO_SALDO)
join      BD_DWTG.DBO.WD_CONTA_CONTABIL_EXERCICIO a15
    on      (a11.ID_ANO_LANC = a15.ID_ANO and
a11.ID_CONTA_CONTABIL_LANC = a15.ID_CONTA_CONTABIL)
join      BD_DWTG.DBO.WD_MES      a16
    on      (a14.ID_ANO = a16.ID_ANO and
a14.ID_MES = a16.ID_MES)
join      BD_DWTG.DBO.WD_CONTA_CORRENTE      a17
    on      (a11.ID_CONTA_CORRENTE_SK = a17.ID_CONTA_CORRENTE_SK)
where     (a11.ID_ORCA_FISCAL_ORGAO_EXEC_H in (0)
and (a14.ID_ANO = {} and a14.ID_MES = {}))
and a15.ID_CONTA_CONTABIL_DESTINO in (812110204, 812110104)
and a17.NR_CONTA_CORRENTE in ('CG0000068', 'CGFSCEIRB', 'CGLEI8036', 'CGPPRONAF', 'CGP
RCACAU', 'CGFSCEIRB', 'CGLEI8036', 'CGPPRONAF', 'CGPRCACAU')
and a13.ID_ANO = a16.ID_ANO
and a13.ID_MES = a16.ID_MES)""

```

### ###query\_contragarantia\_estados

```

query5 = ""select      sum((a11.VA_MOVIMENTO_LIQUIDO * a13.PE_TAXA))      SDCC
from      BD_DWTG.DBO.WF_LANCAMENTO_EP20      a11
join      BD_DWTG.DBO.WD_MOEDA      a12
    on      (a11.ID_MOEDA_UG_EXEC_H = a12.ID_MOEDA)
join      BD_DWTG.DBO.WD_TAXA_CAMBIO_MENSAL      a13
    on      (a12.ID_MOEDA = a13.ID_MOEDA_ORIGEM)
join      BD_DWTG.DBO.WA_MES_ACUM a14
    on      (a11.ID_ANO_LANC = a14.ID_ANO_ACUM_ANO_SALDO and
a11.ID_MES_LANC = a14.ID_MES_ACUM_ANO_SALDO)

```

```

join    BD_DWTG.DBO.WD_CONTA_CONTABIL_EXERCICIO a15
on      (a11.ID_ANO_LANC = a15.ID_ANO and
a11.ID_CONTA_CONTABIL_LANC = a15.ID_CONTA_CONTABIL)
join    BD_DWTG.DBO.WD_MES      a16
on      (a14.ID_ANO = a16.ID_ANO and
a14.ID_MES = a16.ID_MES)
join    BD_DWTG.DBO.WD_CONTA_CORRENTE  a17
on      (a11.ID_CONTA_CORRENTE_SK = a17.ID_CONTA_CORRENTE_SK)
where   (a11.ID_ORCA_FISCAL_ORGAO_EXEC_H in (0)
and (a14.ID_ANO = {} and a14.ID_MES = {}))
and a15.ID_CONTA_CONTABIL_DESTINO in (811110304, 811110404)
and a17.NR_CONTA_CORRENTE in ('CG0000064', 'CG0000069')
and a13.ID_ANO = a16.ID_ANO
and a13.ID_MES = a16.ID_MES)""""

```

### ###query\_contragarantia\_municipios

```

query6 = """"select          sum((a11.VA_MOVIMENTO_LIQUIDO * a13.PE_TAXA))  SDCC
from    BD_DWTG.DBO.WF_LANCAMENTO_EP20  a11
join    BD_DWTG.DBO.WD_MOEDA      a12
on      (a11.ID_MOEDA_UG_EXEC_H = a12.ID_MOEDA)
join    BD_DWTG.DBO.WD_TAXA_CAMBIO_MENSAL      a13
on      (a12.ID_MOEDA = a13.ID_MOEDA_ORIGEM)
join    BD_DWTG.DBO.WA_MES_ACUM a14
on      (a11.ID_ANO_LANC = a14.ID_ANO_ACUM_ANO_SALDO and
a11.ID_MES_LANC = a14.ID_MES_ACUM_ANO_SALDO)
join    BD_DWTG.DBO.WD_CONTA_CONTABIL_EXERCICIO a15
on      (a11.ID_ANO_LANC = a15.ID_ANO and
a11.ID_CONTA_CONTABIL_LANC = a15.ID_CONTA_CONTABIL)
join    BD_DWTG.DBO.WD_MES      a16
on      (a14.ID_ANO = a16.ID_ANO and
a14.ID_MES = a16.ID_MES)
join    BD_DWTG.DBO.WD_CONTA_CORRENTE  a17
on      (a11.ID_CONTA_CORRENTE_SK = a17.ID_CONTA_CORRENTE_SK)
where   (a11.ID_ORCA_FISCAL_ORGAO_EXEC_H in (0)
and (a14.ID_ANO = {} and a14.ID_MES = {}))
and a15.ID_CONTA_CONTABIL_DESTINO in (811110304, 811110404)
and a17.NR_CONTA_CORRENTE in ('CG0000065', 'CG0000070')
and a13.ID_ANO = a16.ID_ANO
and a13.ID_MES = a16.ID_MES)""""

```

### ###query\_contragarantia\_entidades\_controladas

```

query7 = """"select          sum((a11.VA_MOVIMENTO_LIQUIDO * a13.PE_TAXA))  SDCC
from    BD_DWTG.DBO.WF_LANCAMENTO_EP20  a11
join    BD_DWTG.DBO.WD_MOEDA      a12
on      (a11.ID_MOEDA_UG_EXEC_H = a12.ID_MOEDA)
join    BD_DWTG.DBO.WD_TAXA_CAMBIO_MENSAL      a13
on      (a12.ID_MOEDA = a13.ID_MOEDA_ORIGEM)
join    BD_DWTG.DBO.WA_MES_ACUM a14
on      (a11.ID_ANO_LANC = a14.ID_ANO_ACUM_ANO_SALDO and
a11.ID_MES_LANC = a14.ID_MES_ACUM_ANO_SALDO)
join    BD_DWTG.DBO.WD_CONTA_CONTABIL_EXERCICIO a15
on      (a11.ID_ANO_LANC = a15.ID_ANO and
a11.ID_CONTA_CONTABIL_LANC = a15.ID_CONTA_CONTABIL)
join    BD_DWTG.DBO.WD_MES      a16
on      (a14.ID_ANO = a16.ID_ANO and
a14.ID_MES = a16.ID_MES)
join    BD_DWTG.DBO.WD_CONTA_CORRENTE  a17
on      (a11.ID_CONTA_CORRENTE_SK = a17.ID_CONTA_CORRENTE_SK)
where   (a11.ID_ORCA_FISCAL_ORGAO_EXEC_H in (0)

```

```

and (a14.ID_ANO = {}) and a14.ID_MES = {})
and a15.ID_CONTA_CONTABIL_DESTINO in (811110304, 811110404)
and a17.NR_CONTA_CORRENTE in ('CG0000068;CGFSCEIRB;CGLEI8036;CGPPRONAF;CGPRCACA;CGFSC
EIRB;CGLEI8036;CGPPRONAF;CGPRCACA', 'CGFSCEIRB', 'CGLEI8036', 'CGPPRONAF', 'CGPRCACA
U', 'CGFSCEIRB', 'CGLEI8036', 'CGPPRONAF', 'CGPRCACA')
and a13.ID_ANO = a16.ID_ANO
and a13.ID_MES = a16.ID_MES)""

```

###query\_contragarantia\_fundos\_e\_programas

```

query8 = ""select          sum((a11.VA_MOVIMENTO_LIQUIDO * a13.PE_TAXA))  SDCC
from      BD_DWTG.DBO.WF_LANCAMENTO_EP20  a11
      join  BD_DWTG.DBO.WD_MOEDA          a12
      on    (a11.ID_MOEDA_UG_EXEC_H = a12.ID_MOEDA)
      join  BD_DWTG.DBO.WD_TAXA_CAMBIO_MENSAL          a13
      on    (a12.ID_MOEDA = a13.ID_MOEDA_ORIGEM)
      join  BD_DWTG.DBO.WA_MES_ACUM  a14
      on    (a11.ID_ANO_LANC = a14.ID_ANO_ACUM_ANO_SALDO and
a11.ID_MES_LANC = a14.ID_MES_ACUM_ANO_SALDO)
      join  BD_DWTG.DBO.WD_CONTA_CONTABIL_EXERCICIO  a15
      on    (a11.ID_ANO_LANC = a15.ID_ANO and
a11.ID_CONTA_CONTABIL_LANC = a15.ID_CONTA_CONTABIL)
      join  BD_DWTG.DBO.WD_MES          a16
      on    (a14.ID_ANO = a16.ID_ANO and
a14.ID_MES = a16.ID_MES)
      join  BD_DWTG.DBO.WD_CONTA_CORRENTE  a17
      on    (a11.ID_CONTA_CORRENTE_SK = a17.ID_CONTA_CORRENTE_SK)
where  (a11.ID_ORCA_FISCAL_ORGAO_EXEC_H in (0)
and (a14.ID_ANO = {}) and a14.ID_MES = {})
and a15.ID_CONTA_CONTABIL_DESTINO in (811110304, 811110404)
and a17.NR_CONTA_CORRENTE in ('CG0000066', 'CG0000067', 'CG0000071', 'CG0000072')
and a13.ID_ANO = a16.ID_ANO
and a13.ID_MES = a16.ID_MES)""

```

In [4]:

```

### Inserir o(s) período(s) nas 8 consultas nos quadrimestres do ano atual
for i in range(1,9):
    for i2 in [1,2,3]:
        globals()[f'query{i}_exec_quad{i2}'] = globals()[f'query{i}'].format(ano_atual,
quad_to_mes[globals()[f'quad{i2}']])

### Comando para rodar efetivamente as consultas
for i in range(1,9):
    for i2 in [1,2,3]:
        globals()[f'df_query{i}_quad{i2}'] = pd.read_sql(globals()[f'query{i}_exec_quad
{i2}'], conn)

```

In [5]:

```

### Inserir o período (mês = 12) nas 27 consultas nos quadrimestres do ano anterior
for i in range(1,9):
    globals()[f'query{i}_exec'] = globals()[f'query{i}'].format(ano_anterior,12)

### Comando para rodar efetivamente as consultas
for i in range(1,9):
    globals()[f'df_query{i}_ano_anterior'] = pd.read_sql(globals()[f'query{i}_exec'], c
onn)

```

In [76]:

```
### Substituição de "None" por 0 (zero) sempre que a consulta não retornar valor
for i in range(1,9):
    for i2 in [1,2,3]:
        globals()[f'df_query{i}_quad{i2}'].fillna(0,inplace=True)

for i in range(1,9):
    globals()[f'df_query{i}_ano_anterior'].fillna(0,inplace=True)
```

In [77]:

```
### Leitura e tratamento do arquivo referente à Receita Corrente Líquida
df_rcl = pd.read_excel('rcl.xls')
rcl_quad_atual = df_rcl[(df_rcl['ANO']==ano_atual) & (df_rcl['QUAD']==quadrimestre)][['RCL']]
rcl_ano_anterior = df_rcl[(df_rcl['ANO']==ano_atual-1) & (df_rcl['QUAD']==3)][['RCL']]
```



In [78]:

```
### Criação de dicionário para o demonstrativo das Garantias e Contragarantias de Valores
dic = {}
for i2 in ['ano_anterior', 'quad1', 'quad2', 'quad3']:
    for i in range(1,9):
        globals()[f'query{i}v'] = float(globals()[f'df_query{i}_{i2}']['SDCC'][0]/1000)
        if i2=='ano_anterior':
            globals()[f'rcl_{i2}'] = df_rcl[(df_rcl['ANO']==globals()[f'{i2}']) & (df_rcl['QUAD']==3)].RCL.values[0]
        else:
            globals()[f'rcl_{i2}'] = df_rcl[(df_rcl['ANO']==ano_atual) & (df_rcl['QUAD']==globals()[f'{i2}'])].RCL.values[0]

    L1 = query1v
    L2 = query2v
    L3 = query3v
    L4 = query4v
    L5 = query1v + query2v + query3v + query4v
    L6 = globals()[f'rcl_{i2}']
    L7 = L5 / L6 * 100
    L8 = L6 * .6
    L9 = L8 * .9
    L10 = query5v
    L11 = query6v
    L12 = query8v
    L13 = query7v
    L14 = query5v + query6v + query7v + query8v

    dic[i2] = [L1,L2,L3,L4,L5,L6,L7,L8,L9,L10,L11,L12,L13,L14]
df_garant = pd.DataFrame(dic)
df_garant
```

Out[78]:

	ano_anterior	quad1	quad2	quad3
0	1.971862e+08	1.971136e+08	2.031827e+08	2.004635e+08
1	1.673953e+07	1.727904e+07	1.821576e+07	1.945558e+07
2	4.109998e+07	4.060083e+07	4.013792e+07	3.591237e+07
3	2.286233e+07	2.290232e+07	2.289346e+07	2.367457e+07
4	2.778880e+08	2.778958e+08	2.844299e+08	2.795060e+08
5	8.053484e+08	8.186166e+08	8.291573e+08	9.056586e+08
6	3.450532e+01	3.394700e+01	3.430349e+01	3.086218e+01
7	4.832090e+08	4.911700e+08	4.974944e+08	5.433952e+08
8	4.348881e+08	4.420530e+08	4.477449e+08	4.890556e+08
9	1.971862e+08	1.990937e+08	2.031827e+08	2.004635e+08
10	1.673953e+07	1.733300e+07	1.821576e+07	1.945558e+07
11	1.638167e+07	1.680116e+07	1.612278e+07	1.555035e+07
12	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00
13	2.303074e+08	2.332279e+08	2.375213e+08	2.354694e+08

In [79]:

```
### Criação de dicionário referente ao template no formato em que "L" é a linha e "C" é a coluna de cada campo da tabela
### Exemplo, L1C1 é a linha 1 e coluna 1, L5C3 é a linha 5 da coluna 3
### Formatação de valores de acordo com o tipo de cada linha (Linha 7 com 2 casas decimais)
template_dict = {}
df_garant = df_garant.reset_index()
df_garant.drop('index',axis=1,inplace=True)
l1_c1 = df_garant.iloc[0][0]
for l in df_garant.index:
    for c in range(0,df_garant.columns.size):
        if l == 6:
            template_dict['L'+str(l+1)+'_C'+str(c+1)]=str(locale.currency(df_garant.iloc[l][c] ,grouping=True,symbol=False ))
        else:
            template_dict['L'+str(l+1)+'_C'+str(c+1)]=str(locale.currency(round(df_garant.iloc[l][c]) ,grouping=True,symbol=False ))[:-3]
```

In [80]:

```
### Definição de nome para as linhas do df_garant e exportação para excel
linhas = ['AOS ESTADOS (I)',
          'AOS MUNICÍPIOS (II)',
          'ÀS ENTIDADES CONTROLADAS (III)',
          'POR MEIO DE FUNDOS E PROGRAMAS (IV)',
          'TOTAL GARANTIAS CONCEDIDAS (V) = (I + II + III + IV)',
          'RECEITA CORRENTE LÍQUIDA - RCL (VI)',
          '% do TOTAL DAS GARANTIAS sobre a RCL',
          'LIMITE DEFINIDO POR RESOLUÇÃO DO SENADO FEDERAL - 60%',
          'LIMITE DE ALERTA (inc III do §1º do art. 59 da LRF) - (0,9 x 60%) ',
          'DOS ESTADOS (VII)',
          'DOS MUNICÍPIOS (VIII)',
          'DAS ENTIDADES CONTROLADAS (IX)',
          'EM GARANTIAS POR MEIO DE FUNDOS E PROGRAMAS (X)',
          'TOTAL CONTRAGARANTIAS RECEBIDAS (XI) = (VII + VIII + IX + X),]
df_garant.index = linhas
df_garant.to_excel('df_garant.xlsx')
```

In [81]:

```
### Criação de df_assistido, cuja função é alimentar os parágrafos da minuta de instrução assistida
### Cada "Valores" é uma coluna
valores1 = []
valores2 = []
valores3 = []
valores4 = []
valores5 = []
valores6 = []
for x in df_garant.reset_index().index:

    valores1.append(df_garant.loc[df_garant.index[x],df_garant.columns[-1]])

    valores2.append(df_garant.loc[df_garant.index[x],df_garant.columns[-2]])

    valores3.append(df_garant.loc[df_garant.index[x],df_garant.columns[-1]]-df_garant.loc[df_garant.index[x],df_garant.columns[-2]])

    valores4.append((df_garant.loc[df_garant.index[x],df_garant.columns[-1]]/df_garant.loc[df_garant.index[x],df_garant.columns[-2]]-1)*100)

    if df_garant.loc[df_garant.index[x],df_garant.columns[-1]]==df_garant.loc[df_garant.index[x],df_garant.columns[-2]]:
        valores5.append('permaneceu inalterado')
    else:
        if df_garant.loc[df_garant.index[x],df_garant.columns[-1]]==df_garant.loc[df_garant.index[x],df_garant.columns[-2]]>0:
            valores5.append('aumentou')
        else:
            valores5.append('diminuiu')

    if df_garant.loc[df_garant.index[x],df_garant.columns[-1]]==df_garant.loc[df_garant.index[x],df_garant.columns[-2]]:
        valores6.append('manutenção de valores')
    else:
        if df_garant.loc[df_garant.index[x],df_garant.columns[-1]]==df_garant.loc[df_garant.index[x],df_garant.columns[-2]]>0:
            valores6.append('crescimento de')
        else:
            valores6.append('redução de')

df_assistido = pd.DataFrame({
    'quad_atual': valores1,
    'quad_anterior':valores2,
    'Variação em Reais': valores3,
    'Variação percentual': valores4,
    'verbo':valores5,
    'substantivo':valores6    })

### Atribuição de 0 (zero) sempre que o resultado retornar erro
df_assistido.fillna(0,inplace = True)
df_assistido
```

/usr/lib/python3.6/site-packages/ipykernel\_launcher.py:17: RuntimeWarning:  
invalid value encountered in double\_scalars

Out[81]:

	quad_atual	quad_anterior	Variação em Reais	Variação percentual	verbo	substantivo
0	2.004635e+08	2.031827e+08	-2.719246e+06	-1.338325	diminuiu	redução de
1	1.945558e+07	1.821576e+07	1.239815e+06	6.806274	diminuiu	redução de
2	3.591237e+07	4.013792e+07	-4.225545e+06	-10.527565	diminuiu	redução de
3	2.367457e+07	2.289346e+07	7.811090e+05	3.411930	diminuiu	redução de
4	2.795060e+08	2.844299e+08	-4.923867e+06	-1.731136	diminuiu	redução de
5	9.056586e+08	8.291573e+08	7.650130e+07	9.226391	diminuiu	redução de
6	3.086218e+01	3.430349e+01	-3.441306e+00	-10.031941	diminuiu	redução de
7	5.433952e+08	4.974944e+08	4.590078e+07	9.226391	diminuiu	redução de
8	4.890556e+08	4.477449e+08	4.131070e+07	9.226391	diminuiu	redução de
9	2.004635e+08	2.031827e+08	-2.719246e+06	-1.338325	diminuiu	redução de
10	1.945558e+07	1.821576e+07	1.239815e+06	6.806274	diminuiu	redução de
11	1.555035e+07	1.612278e+07	-5.724352e+05	-3.550473	diminuiu	redução de
12	0.000000e+00	0.000000e+00	0.000000e+00	0.000000	permaneceu inalterado	manutenção de valores
13	2.354694e+08	2.375213e+08	-2.051866e+06	-0.863866	diminuiu	redução de

In [82]:

```
### Total das Contragarantias dividido pela RCL
Z_aux1 = str(locale.currency(df_assistido.loc[5,'quad_atual']/df_assistido.loc[5,'quad_atual']*100,grouping=True)).replace('R$ ','')

### Total das Contragarantias dividido pelo Total das Garantias
Z_aux2 = str(locale.currency(df_assistido.loc[13,'quad_atual']/df_assistido.loc[4,'quad_atual']*100,grouping=True)).replace('R$ ','')

print(Z_aux1)
print(Z_aux2)
```

26,00  
84,24

In [83]:

```
### Retirada de sinal negativo na 3ª coluna do df_assistido, para uso na minuta de instrução assistida
df_assistido['Variação em Reais'] = df_assistido['Variação em Reais'].apply(lambda x: abs(x))
```

In [84]:

```
### Visualização do df_assistido  
df_assistido
```

Out[84]:

	quad_atual	quad_anterior	Varição em Reais	Varição percentual	verbo	substantivo
0	2.004635e+08	2.031827e+08	2.719246e+06	-1.338325	diminuiu	redução de
1	1.945558e+07	1.821576e+07	1.239815e+06	6.806274	diminuiu	redução de
2	3.591237e+07	4.013792e+07	4.225545e+06	-10.527565	diminuiu	redução de
3	2.367457e+07	2.289346e+07	7.811090e+05	3.411930	diminuiu	redução de
4	2.795060e+08	2.844299e+08	4.923867e+06	-1.731136	diminuiu	redução de
5	9.056586e+08	8.291573e+08	7.650130e+07	9.226391	diminuiu	redução de
6	3.086218e+01	3.430349e+01	3.441306e+00	-10.031941	diminuiu	redução de
7	5.433952e+08	4.974944e+08	4.590078e+07	9.226391	diminuiu	redução de
8	4.890556e+08	4.477449e+08	4.131070e+07	9.226391	diminuiu	redução de
9	2.004635e+08	2.031827e+08	2.719246e+06	-1.338325	diminuiu	redução de
10	1.945558e+07	1.821576e+07	1.239815e+06	6.806274	diminuiu	redução de
11	1.555035e+07	1.612278e+07	5.724352e+05	-3.550473	diminuiu	redução de
12	0.000000e+00	0.000000e+00	0.000000e+00	0.000000	permaneceu inalterado	manutenção de valores
13	2.354694e+08	2.375213e+08	2.051866e+06	-0.863866	diminuiu	redução de



In [85]:

```
### Criação de dicionário para alimentação da minuta de instrução assistida
### Z1_1 é a linha 1 e coluna 1, Z5_3 é a linha 5 da coluna 3
### Formatação dos valores da tabela acima.
### As 3 primeiras colunas são números, a quarta é percentual, as duas últimas são text
o.
template_dict_z = {}
strMilhao = ''
strReais = 'R$ '
for l in df_assistido.index:
    if l != 6:
        for c in range(0,df_assistido.columns.size):
            strMilhao=''
            if c < 3:
                if round(abs(df_assistido.iloc[l][c])/1000000 >= 2:
                    strMilhao = ' bilhões'
                    template_dict_z['Z'+str(l+1)+'_'+str(c+1)]=strReais + str(locale.cu
rrency(round(df_assistido.iloc[l][c]/1000000,2) ,grouping=True,symbol=False )) + strMil
hao
                else:
                    if round(abs(df_assistido.iloc[l][c])/1000000 >= 1:
                        strMilhao = ' bilhão'
                        template_dict_z['Z'+str(l+1)+'_'+str(c+1)]=strReais + str(local
e.currency(round(df_assistido.iloc[l][c]/1000000,2) ,grouping=True,symbol=False )) + st
rMilhao
                    else:
                        if round(abs(df_assistido.iloc[l][c])/1000 >= 2:
                            strMilhao = ' milhões'
                            template_dict_z['Z'+str(l+1)+'_'+str(c+1)]=strReais + str(l
ocale.currency(round(df_assistido.iloc[l][c]/1000,2) ,grouping=True,symbol=False )) + s
trMilhao
                        else:
                            if round(abs(df_assistido.iloc[l][c])/1000 >= 1:
                                strMilhao = ' milhão'
                                template_dict_z['Z'+str(l+1)+'_'+str(c+1)]=strReais + s
tr(locale.currency(round(df_assistido.iloc[l][c]/1000,2) ,grouping=True,symbol=False ))
                                + strMilhao
                            else:
                                if abs(df_assistido.iloc[l][c])==0:
                                    template_dict_z['Z'+str(l+1)+'_'+str(c+1)]= ''
                                else:
                                    template_dict_z['Z'+str(l+1)+'_'+str(c+1)]=strReais
                                    + str(locale.currency(round(df_assistido.iloc[l][c]/1000,2) ,grouping=True,symbol=False
                                    )) + strMilhao

            if c == 3:
                if abs(df_assistido.iloc[l][c])==0:
                    template_dict_z['Z'+str(l+1)+'_'+str(c+1)]=''
                else:
                    template_dict_z['Z'+str(l+1)+'_'+str(c+1)]=str(round(df_assistido.i
loc[l][c],2)).replace('.',',')
            if c > 3:
                template_dict_z['Z'+str(l+1)+'_'+str(c+1)]=df_assistido.iloc[l][c]
        else:
            for c in range(0,df_assistido.columns.size):
                if c < 3:
                    template_dict_z['Z'+str(l+1)+'_'+str(c+1)]=str(locale.currency(round(df
_assistido.iloc[l][c],2) ,grouping=True,symbol=False ))
                else:
                    if c == 3:
```

```
        if abs(df_assistido.iloc[1][c])==0:
            template_dict_z['Z'+str(l+1)+'_'+str(c+1)]=''
        else:
            template_dict_z['Z'+str(l+1)+'_'+str(c+1)]=str(round(df_assistido.iloc[1][c],2)).replace('.',',')
        else:
            template_dict_z['Z'+str(l+1)+'_'+str(c+1)]=df_assistido.iloc[1][c]
```

In [86]:

```
### Visualização do % do total das garantias em relação à RCL
df_assistido.iloc[6][0]
```

Out[86]:

30.862183106934467

In [87]:

```
### Texto para conclusão da análise, a partir do resultado do resultado acima (Z7_1, ou
df_assistido.iloc[6][0])
if df_assistido.iloc[6][0] < (60 *0.9):
    Z_garant_conclusao = str('O percentual das Garantias em relação à RCL foi de '+str(
template_dict_z['Z7_1'])+', portanto, inferior a 90% do limite de referência, que é 60%
da RCL.')
else:
    if df_assistido.iloc[6][0] <= 60:
        Z_garant_conclusao = str('O percentual da DCL em relação à RCL foi de '+str(tem
plate_dict_z['Z7_1'])+', portanto, ficou entre 90 e 100% do limite de referência. Nessa
situação, em atenção ao disposto no art. 59, § 1º da LRF, propõe-se a emissão de alerta
ao Poder Executivo, registrando que, nos termos do art. 40, § 5º da LRF, é nula a garan
tia concedida acima dos limites fixados pelo Senado Federal.')
    else:
        Z_garant_conclusao = str('O percentual da DCL em relação à RCL foi de '+str(tem
plate_dict_z['Z7_1'])+', portanto, superior ao limite de referência. Em virtude disso,
propõe-se determinar ao Poder Executivo a adoção de medidas com vistas a adequação ao
disposto no art. 40, § 5º da LRF, que diz que é nula a garantia concedida acima dos li
mites fixados pelo Senado Federal.')

Z_garant_conclusao
```

Out[87]:

'O percentual das Garantias em relação à RCL foi de 30,86, portanto, inferior a 90% do limite de referência, que é 60% da RCL.'

In [88]:

```
### Definição de outros campos do template.
template_dict['ano_anterior']=(ano_atual -1)
template_dict['ano_atual']=(ano_atual)
template_dict['quadrimestre']=(quadrimestre)
template_dict['Z_aux1']=(Z_aux1)
template_dict['Z_aux2']=(Z_aux2)
template_dict['Z_garant_conclusao']=(Z_garant_conclusao)
```



In [89]:

```
### Atualização do template para inserir os campos do tipo "Z".  
template_dict.update(template_dict_z)  
  
### Visualização da lista referente ao template_dict atualizado.  
template_dict
```

Out[89]:

```
{ 'L1_C1': '197.186.182',  
  'L1_C2': '197.113.572',  
  'L1_C3': '203.182.735',  
  'L1_C4': '200.463.490',  
  'L2_C1': '16.739.529',  
  'L2_C2': '17.279.045',  
  'L2_C3': '18.215.762',  
  'L2_C4': '19.455.577',  
  'L3_C1': '41.099.979',  
  'L3_C2': '40.600.834',  
  'L3_C3': '40.137.918',  
  'L3_C4': '35.912.373',  
  'L4_C1': '22.862.328',  
  'L4_C2': '22.902.323',  
  'L4_C3': '22.893.464',  
  'L4_C4': '23.674.573',  
  'L5_C1': '277.888.019',  
  'L5_C2': '277.895.773',  
  'L5_C3': '284.429.879',  
  'L5_C4': '279.506.012',  
  'L6_C1': '805.348.403',  
  'L6_C2': '818.616.620',  
  'L6_C3': '829.157.292',  
  'L6_C4': '905.658.590',  
  'L7_C1': '34,51',  
  'L7_C2': '33,95',  
  'L7_C3': '34,30',  
  'L7_C4': '30,86',  
  'L8_C1': '483.209.042',  
  'L8_C2': '491.169.972',  
  'L8_C3': '497.494.375',  
  'L8_C4': '543.395.154',  
  'L9_C1': '434.888.138',  
  'L9_C2': '442.052.975',  
  'L9_C3': '447.744.938',  
  'L9_C4': '489.055.638',  
  'L10_C1': '197.186.182',  
  'L10_C2': '199.093.740',  
  'L10_C3': '203.182.735',  
  'L10_C4': '200.463.490',  
  'L11_C1': '16.739.529',  
  'L11_C2': '17.332.999',  
  'L11_C3': '18.215.762',  
  'L11_C4': '19.455.577',  
  'L12_C1': '16.381.671',  
  'L12_C2': '16.801.162',  
  'L12_C3': '16.122.785',  
  'L12_C4': '15.550.349',  
  'L13_C1': '0',  
  'L13_C2': '0',  
  'L13_C3': '0',  
  'L13_C4': '0',  
  'L14_C1': '230.307.382',  
  'L14_C2': '233.227.901',  
  'L14_C3': '237.521.282',  
  'L14_C4': '235.469.416',  
  'ano_anterior': 2018,  
  'ano_atual': 2019,  
  'quadrimestre': 3,
```

'Z\_aux1': '26,00',  
'Z\_aux2': '84,24',  
'Z\_garant\_conclusao': 'O percentual das Garantias em relação à RCL foi de 30,86, portanto, inferior a 90% do limite de referência, que é 60% da RC L.',  
'Z1\_1': 'R\$ 200,46 bilhões',  
'Z1\_2': 'R\$ 203,18 bilhões',  
'Z1\_3': 'R\$ 2,72 bilhões',  
'Z1\_4': '-1,34',  
'Z1\_5': 'diminuiu',  
'Z1\_6': 'redução de',  
'Z2\_1': 'R\$ 19,46 bilhões',  
'Z2\_2': 'R\$ 18,22 bilhões',  
'Z2\_3': 'R\$ 1,24 bilhão',  
'Z2\_4': '6,81',  
'Z2\_5': 'diminuiu',  
'Z2\_6': 'redução de',  
'Z3\_1': 'R\$ 35,91 bilhões',  
'Z3\_2': 'R\$ 40,14 bilhões',  
'Z3\_3': 'R\$ 4,23 bilhões',  
'Z3\_4': '-10,53',  
'Z3\_5': 'diminuiu',  
'Z3\_6': 'redução de',  
'Z4\_1': 'R\$ 23,67 bilhões',  
'Z4\_2': 'R\$ 22,89 bilhões',  
'Z4\_3': 'R\$ 781,11 milhões',  
'Z4\_4': '3,41',  
'Z4\_5': 'diminuiu',  
'Z4\_6': 'redução de',  
'Z5\_1': 'R\$ 279,51 bilhões',  
'Z5\_2': 'R\$ 284,43 bilhões',  
'Z5\_3': 'R\$ 4,92 bilhões',  
'Z5\_4': '-1,73',  
'Z5\_5': 'diminuiu',  
'Z5\_6': 'redução de',  
'Z6\_1': 'R\$ 905,66 bilhões',  
'Z6\_2': 'R\$ 829,16 bilhões',  
'Z6\_3': 'R\$ 76,50 bilhões',  
'Z6\_4': '9,23',  
'Z6\_5': 'diminuiu',  
'Z6\_6': 'redução de',  
'Z7\_1': '30,86',  
'Z7\_2': '34,30',  
'Z7\_3': '3,44',  
'Z7\_4': '-10,03',  
'Z7\_5': 'diminuiu',  
'Z7\_6': 'redução de',  
'Z8\_1': 'R\$ 543,40 bilhões',  
'Z8\_2': 'R\$ 497,49 bilhões',  
'Z8\_3': 'R\$ 45,90 bilhões',  
'Z8\_4': '9,23',  
'Z8\_5': 'diminuiu',  
'Z8\_6': 'redução de',  
'Z9\_1': 'R\$ 489,06 bilhões',  
'Z9\_2': 'R\$ 447,74 bilhões',  
'Z9\_3': 'R\$ 41,31 bilhões',  
'Z9\_4': '9,23',  
'Z9\_5': 'diminuiu',  
'Z9\_6': 'redução de',  
'Z10\_1': 'R\$ 200,46 bilhões',  
'Z10\_2': 'R\$ 203,18 bilhões',

```
'Z10_3': 'R$ 2,72 bilhões',
'Z10_4': '-1,34',
'Z10_5': 'diminuiu',
'Z10_6': 'redução de',
'Z11_1': 'R$ 19,46 bilhões',
'Z11_2': 'R$ 18,22 bilhões',
'Z11_3': 'R$ 1,24 bilhão',
'Z11_4': '6,81',
'Z11_5': 'diminuiu',
'Z11_6': 'redução de',
'Z12_1': 'R$ 15,55 bilhões',
'Z12_2': 'R$ 16,12 bilhões',
'Z12_3': 'R$ 572,44 milhões',
'Z12_4': '-3,55',
'Z12_5': 'diminuiu',
'Z12_6': 'redução de',
'Z13_1': '',
'Z13_2': '',
'Z13_3': '',
'Z13_4': '',
'Z13_5': 'permaneceu inalterado',
'Z13_6': 'manutenção de valores',
'Z14_1': 'R$ 235,47 bilhões',
'Z14_2': 'R$ 237,52 bilhões',
'Z14_3': 'R$ 2,05 bilhões',
'Z14_4': '-0,86',
'Z14_5': 'diminuiu',
'Z14_6': 'redução de'}
```

In [90]:

```
### Leitura do template relativo ao quadrimestre atual
doc_garant = docxtpl.DocxTemplate(f"templates_rgf/template_garantias_{quadrimestre}.doc
x")
#renderização do documento
doc_garant.render(template_dict)
#salvamento o documento
doc_garant.save("doc_garant.docx")
```

In [91]:

```
# ### glossário para alimentação eficiente do template seguinte
# L1_C1v = float(df_L1_ano_anterior['SDCC'][0]/1000)
# L2_C1v = float(df_L2_ano_anterior['SDCC'][0]/1000)
# L3_C1v = float(df_L3_ano_anterior['SDCC'][0]/1000)
# L4_C1v = float(df_L4_ano_anterior['SDCC'][0]/1000)
# L5_C1v = L1_C1v + L2_C1v + L3_C1v + L4_C1v
# L6_C1v = float(rcl_ano_anterior)
# L7_C1v = L5_C1v / L6_C1v
# L8_C1v = L6_C1v * .6
# L9_C1v = L8_C1v * .9
# L10_C1v = float(df_L10_ano_anterior['SDCC'][0]/1000)
# L11_C1v = float(df_L11_ano_anterior['SDCC'][0]/1000)
# L12_C1v = float(df_L12_ano_anterior['SDCC'][0]/1000)
# L13_C1v = float(df_L12_ano_anterior['SDCC'][0]/1000)
# L14_C1v = L10_C1v + L11_C1v + L12_C1v + L13_C1v

# L1_C2v = float(df_L1['SDCC'][0]/1000)
# L2_C2v = float(df_L2['SDCC'][0]/1000)
# L3_C2v = float(df_L3['SDCC'][0]/1000)
# L4_C2v = float(df_L4['SDCC'][0]/1000)
# L5_C2v = L1_C2v + L2_C2v + L3_C2v + L4_C2v
# L6_C2v = float(rcl_quad_atual)
# L7_C2v = L5_C2v / L6_C2v
# L8_C2v = L6_C2v * .6
# L9_C2v = L8_C2v * .9
# L10_C2v = float(df_L10['SDCC'][0]/1000)
# L11_C2v = float(df_L11['SDCC'][0]/1000)
# L12_C2v = float(df_L12['SDCC'][0]/1000)
# L13_C2v = float(df_L13['SDCC'][0]/1000)
# L14_C2v = L10_C2v + L11_C2v + L12_C2v + L13_C2v
```

In [92]:

```
# ### glossário para alimentação eficiente do template seguinte
# L1_C1 = str(locale.currency( L1_C1v,grouping=True )).replace('R$ ','')
# L2_C1 = str(locale.currency( L2_C1v,grouping=True )).replace('R$ ','')
# L3_C1 = str(locale.currency( L3_C1v,grouping=True )).replace('R$ ','')
# L4_C1 = str(locale.currency( L4_C1v,grouping=True )).replace('R$ ','')
# L5_C1 = str(locale.currency( L5_C1v,grouping=True )).replace('R$ ','')
# L6_C1 = str(locale.currency( L6_C1v,grouping=True )).replace('R$ ','')
# L7_C1 = str(locale.currency( L7_C1v,grouping=True )).replace('R$ ','')
# L8_C1 = str(locale.currency( L8_C1v,grouping=True )).replace('R$ ','')
# L9_C1 = str(locale.currency( L9_C1v,grouping=True )).replace('R$ ','')
# L10_C1 = str(locale.currency( L10_C1v,grouping=True )).replace('R$ ','')
# L11_C1 = str(locale.currency( L11_C1v,grouping=True )).replace('R$ ','')
# L12_C1 = str(locale.currency( L12_C1v,grouping=True )).replace('R$ ','')
# L13_C1 = str(locale.currency( L13_C1v,grouping=True )).replace('R$ ','')
# L14_C1 = str(locale.currency( L14_C1v,grouping=True )).replace('R$ ','')

# L1_C2 = str(locale.currency( L1_C2v,grouping=True )).replace('R$ ','')
# L2_C2 = str(locale.currency( L2_C2v,grouping=True )).replace('R$ ','')
# L3_C2 = str(locale.currency( L3_C2v,grouping=True )).replace('R$ ','')
# L4_C2 = str(locale.currency( L4_C2v,grouping=True )).replace('R$ ','')
# L5_C2 = str(locale.currency( L5_C2v,grouping=True )).replace('R$ ','')
# L6_C2 = str(locale.currency( L6_C2v,grouping=True )).replace('R$ ','')
# L7_C2 = str(locale.currency( L7_C2v,grouping=True )).replace('R$ ','')
# L8_C2 = str(locale.currency( L8_C2v,grouping=True )).replace('R$ ','')
# L9_C2 = str(locale.currency( L9_C2v,grouping=True )).replace('R$ ','')
# L10_C2 = str(locale.currency( L10_C2v,grouping=True )).replace('R$ ','')
# L11_C2 = str(locale.currency( L11_C2v,grouping=True )).replace('R$ ','')
# L12_C2 = str(locale.currency( L12_C2v,grouping=True )).replace('R$ ','')
# L13_C2 = str(locale.currency( L13_C2v,grouping=True )).replace('R$ ','')
# L14_C2 = str(locale.currency( L14_C2v,grouping=True )).replace('R$ ','')
```

conn.close()

In [ ]:

```
ANEXO III - OPERAÇÕES DE CRÉDITO
```

In [16]:

```
import docxtempl
import pandas as pd
import pyodbc
import locale

conn = pyodbc.connect('DSN=LABCONTAS')
quad_to_mes = {1:4,2:8,3:12}
locale.setlocale( locale.LC_MONETARY , 'pt_BR.UTF-8' )
'{} {}'.format('one', 'two')
```

Out[16]:

```
'one two'
```

In [17]:

```
ano_atual = 2019
quadrimestre = 3
ano_anterior = ano_atual - 1
quad1 = 1
quad2 = 2
quad3 = 3
dict_auxilar={3:['ano_anterior','quad1','quad2','quad3'],
              2:['ano_anterior','quad1','quad2'],
              1:['ano_anterior','quad1']}
```

In [19]:

###L3+ *Mobiliária Interna - Refinanciamento*

```
query1 = """select          sum((a11.VA_MOVIMENTO_LIQUIDO * a13.PE_TAXA))  SDCC
from      BD_DWTG.DBO.WF_LANCAMENTO_EP20  a11
join      BD_DWTG.DBO.WD_MOEDA            a12
          on      (a11.ID_MOEDA_UG_EXEC_H = a12.ID_MOEDA)
join      BD_DWTG.DBO.WD_TAXA_CAMBIO_MENSAL  a13
          on      (a12.ID_MOEDA = a13.ID_MOEDA_ORIGEM)
join      BD_DWTG.DBO.WA_MES_ACUM          a14
          on      (a11.ID_ANO_LANC = a14.ID_ANO_ACUM_ANO_SALDO and
a11.ID_MES_LANC = a14.ID_MES_ACUM_ANO_SALDO)
join      BD_DWTG.DBO.WD_CONTA_CONTABIL_EXERCICIO a15
          on      (a11.ID_ANO_LANC = a15.ID_ANO and
a11.ID_CONTA_CONTABIL_LANC = a15.ID_CONTA_CONTABIL)
join      BD_DWTG.DBO.WD_MES              a16
          on      (a14.ID_ANO = a16.ID_ANO and
a14.ID_MES = a16.ID_MES)
join      BD_DWTG.DBO.WD_NATUREZA_RECEITA2  a17
          on      (a11.ID_ALINEA2_NARE = a17.ID_ALINEA2_NARE and
a11.ID_CATEGORIA_ECONOMICA2_NARE = a17.ID_CATEGORIA_ECONOMICA2_NARE and
a11.ID_ESPECIE_RECEITA2_NARE = a17.ID_ESPECIE_RECEITA2_NARE and
a11.ID_ORIGEM_RECEITA2_NARE = a17.ID_ORIGEM_RECEITA2_NARE and
a11.ID_RUBRICA2_NARE = a17.ID_RUBRICA2_NARE and
a11.ID_SUBALINEA2_NARE = a17.ID_SUBALINEA2_NARE and
a11.ID_TIPO_RECEITA2_NARE = a17.ID_TIPO_RECEITA2_NARE)
where     ((a14.ID_ANO = {}) and a14.ID_MES = {})
and a11.ID_ORCA_FISCAL_ORGAO_EXEC_H in (0)
and a15.ID_CONTA_CONTABIL_DESTINO in (621310000, 621320000, 621330000, 621340000, 6213
90000, 621200000)
and (a17.CO_NATUREZA_RECEITA2 like '2111002%'
or a17.CO_NATUREZA_RECEITA2 like '8111002%')
and a13.ID_ANO = a16.ID_ANO
and a13.ID_MES = a16.ID_MES)"""
```

###L3+ *Mobiliária Interna - Assunção e Reconhecimento de Dívidas (art. 29. LRF)*

```
query2 = """select          sum((a11.VA_MOVIMENTO_LIQUIDO * a13.PE_TAXA))  SDCC
from      BD_DWTG.DBO.WF_LANCAMENTO_EP02  a11
join      BD_DWTG.DBO.WD_MOEDA            a12
          on      (a11.ID_MOEDA_UG_EXEC_H = a12.ID_MOEDA)
join      BD_DWTG.DBO.WD_TAXA_CAMBIO_MENSAL  a13
          on      (a12.ID_MOEDA = a13.ID_MOEDA_ORIGEM)
join      BD_DWTG.DBO.WA_MES_ACUM          a14
          on      (a11.ID_ANO_LANC = a14.ID_ANO_ACUM_ANO_SALDO and
a11.ID_MES_LANC = a14.ID_MES_ACUM_ANO_SALDO)
join      BD_DWTG.DBO.WD_CONTA_CONTABIL_EXERCICIO a15
          on      (a11.ID_ANO_LANC = a15.ID_ANO and
a11.ID_CONTA_CONTABIL_LANC = a15.ID_CONTA_CONTABIL)
join      BD_DWTG.DBO.WD_MES              a16
          on      (a14.ID_ANO = a16.ID_ANO and
a14.ID_MES = a16.ID_MES)
where     ((a14.ID_ANO = {}) and a14.ID_MES = {})
and a11.ID_ORCA_FISCAL_ORGAO_EXEC_H in (0)
and a13.ID_ANO = a16.ID_ANO
and a13.ID_MES = a16.ID_MES
and a15.ID_CONTA_CONTABIL_DESTINO in (896110309, 896110310)"""
```



###L3+ Mobiliária Interna - Outras Orçamentárias

```
query3 = ""select          sum((a11.VA_MOVIMENTO_LIQUIDO * a13.PE_TAXA))  SDCC
from    BD_DWTG.DBO.WF_LANCAMENTO_EP20  a11
        join    BD_DWTG.DBO.WD_MOEDA      a12
            on   (a11.ID_MOEDA_UG_EXEC_H = a12.ID_MOEDA)
        join    BD_DWTG.DBO.WD_TAXA_CAMBIO_MENSAL      a13
            on   (a12.ID_MOEDA = a13.ID_MOEDA_ORIGEM)
        join    BD_DWTG.DBO.WA_MES_ACUM  a14
            on   (a11.ID_ANO_LANC = a14.ID_ANO_ACUM_ANO_SALDO and
a11.ID_MES_LANC = a14.ID_MES_ACUM_ANO_SALDO)
        join    BD_DWTG.DBO.WD_CONTA_CONTABIL_EXERCICIO a15
            on   (a11.ID_ANO_LANC = a15.ID_ANO and
a11.ID_CONTA_CONTABIL_LANC = a15.ID_CONTA_CONTABIL)
        join    BD_DWTG.DBO.WD_MES      a16
            on   (a14.ID_ANO = a16.ID_ANO and
a14.ID_MES = a16.ID_MES)
        join    BD_DWTG.DBO.WD_NATUREZA_RECEITA2      a17
            on   (a11.ID_ALINEA2_NARE = a17.ID_ALINEA2_NARE and
a11.ID_CATEGORIA_ECONOMICA2_NARE = a17.ID_CATEGORIA_ECONOMICA2_NARE and
a11.ID_ESPECIE_RECEITA2_NARE = a17.ID_ESPECIE_RECEITA2_NARE and
a11.ID_ORIGEM_RECEITA2_NARE = a17.ID_ORIGEM_RECEITA2_NARE and
a11.ID_RUBRICA2_NARE = a17.ID_RUBRICA2_NARE and
a11.ID_SUBALINEA2_NARE = a17.ID_SUBALINEA2_NARE and
a11.ID_TIPO_RECEITA2_NARE = a17.ID_TIPO_RECEITA2_NARE)
where    ((a14.ID_ANO = { } and a14.ID_MES = { })
and a11.ID_ORCA_FISCAL_ORGAO_EXEC_H in (0)
and a15.ID_CONTA_CONTABIL_DESTINO in (621310000, 621320000, 621330000, 621340000, 6213
90000, 621200000)
and (a17.CO_NATUREZA_RECEITA2 like '2111003%'
or a17.CO_NATUREZA_RECEITA2 like '2111001%')
and a13.ID_ANO = a16.ID_ANO
and a13.ID_MES = a16.ID_MES)""
```

###L3+ L12+ Mobiliária Interna - Extra-Orçamentárias - Aporte BACEN

```
query4 = ""select          sum((a11.VA_MOVIMENTO_LIQUIDO * a13.PE_TAXA))  SDCC
from    BD_DWTG.DBO.WF_LANCAMENTO_EP02  a11
        join    BD_DWTG.DBO.WD_MOEDA      a12
            on   (a11.ID_MOEDA_UG_EXEC_H = a12.ID_MOEDA)
        join    BD_DWTG.DBO.WD_TAXA_CAMBIO_MENSAL      a13
            on   (a12.ID_MOEDA = a13.ID_MOEDA_ORIGEM)
        join    BD_DWTG.DBO.WA_MES_ACUM  a14
            on   (a11.ID_ANO_LANC = a14.ID_ANO_ACUM_ANO_SALDO and
a11.ID_MES_LANC = a14.ID_MES_ACUM_ANO_SALDO)
        join    BD_DWTG.DBO.WD_CONTA_CONTABIL_EXERCICIO a15
            on   (a11.ID_ANO_LANC = a15.ID_ANO and
a11.ID_CONTA_CONTABIL_LANC = a15.ID_CONTA_CONTABIL)
        join    BD_DWTG.DBO.WD_MES      a16
            on   (a14.ID_ANO = a16.ID_ANO and
a14.ID_MES = a16.ID_MES)
where    ((a14.ID_ANO = { } and a14.ID_MES = { })
and a11.ID_ORCA_FISCAL_ORGAO_EXEC_H in (0)
and a13.ID_ANO = a16.ID_ANO
and a13.ID_MES = a16.ID_MES
and a15.ID_CONTA_CONTABIL_DESTINO in (896110303, 896110304))""
```

###L3+ Mobiliária Interna - Extra-Orçamentárias - Aporte em Empresas

```
query5 = ""select          sum((a11.VA_MOVIMENTO_LIQUIDO * a13.PE_TAXA))  SDCC
```

```

from BD_DWTG.DBO.WF_LANCAMENTO_EP02 a11
join BD_DWTG.DBO.WD_MOEDA a12
  on (a11.ID_MOEDA_UG_EXEC_H = a12.ID_MOEDA)
join BD_DWTG.DBO.WD_TAXA_CAMBIO_MENSAL a13
  on (a12.ID_MOEDA = a13.ID_MOEDA_ORIGEM)
join BD_DWTG.DBO.WA_MES_ACUM a14
  on (a11.ID_ANO_LANC = a14.ID_ANO_ACUM_ANO_SALDO and
a11.ID_MES_LANC = a14.ID_MES_ACUM_ANO_SALDO)
join BD_DWTG.DBO.WD_CONTA_CONTABIL_EXERCICIO a15
  on (a11.ID_ANO_LANC = a15.ID_ANO and
a11.ID_CONTA_CONTABIL_LANC = a15.ID_CONTA_CONTABIL)
join BD_DWTG.DBO.WD_MES a16
  on (a14.ID_ANO = a16.ID_ANO and
a14.ID_MES = a16.ID_MES)
where ((a14.ID_ANO = {}) and a14.ID_MES = {})
and a11.ID_ORCA_FISCAL_ORGAO_EXEC_H in (0)
and a13.ID_ANO = a16.ID_ANO
and a13.ID_MES = a16.ID_MES
and a15.ID_CONTA_CONTABIL_DESTINO in (896110311, 896110312))""

```

###L3+ Mobiliária Interna - Extra-Orçamentárias - Demais

```

query6 = ""select      sum((a11.VA_MOVIMENTO_LIQUIDO * a13.PE_TAXA))  SDCC
from BD_DWTG.DBO.WF_LANCAMENTO_EP02 a11
join BD_DWTG.DBO.WD_MOEDA a12
  on (a11.ID_MOEDA_UG_EXEC_H = a12.ID_MOEDA)
join BD_DWTG.DBO.WD_TAXA_CAMBIO_MENSAL a13
  on (a12.ID_MOEDA = a13.ID_MOEDA_ORIGEM)
join BD_DWTG.DBO.WA_MES_ACUM a14
  on (a11.ID_ANO_LANC = a14.ID_ANO_ACUM_ANO_SALDO and
a11.ID_MES_LANC = a14.ID_MES_ACUM_ANO_SALDO)
join BD_DWTG.DBO.WD_CONTA_CONTABIL_EXERCICIO a15
  on (a11.ID_ANO_LANC = a15.ID_ANO and
a11.ID_CONTA_CONTABIL_LANC = a15.ID_CONTA_CONTABIL)
join BD_DWTG.DBO.WD_MES a16
  on (a14.ID_ANO = a16.ID_ANO and
a14.ID_MES = a16.ID_MES)
where ((a14.ID_ANO = {}) and a14.ID_MES = {})
and a11.ID_ORCA_FISCAL_ORGAO_EXEC_H in (0)
and a13.ID_ANO = a16.ID_ANO
and a13.ID_MES = a16.ID_MES
and a15.ID_CONTA_CONTABIL_DESTINO in (896110301, 896110302, 896110305, 896110306))""

```

###L4+ Mobiliária Externa - Refinanciamento

```

query7 = ""select      sum((a11.VA_MOVIMENTO_LIQUIDO * a13.PE_TAXA))  SDCC
from BD_DWTG.DBO.WF_LANCAMENTO_EP20 a11
join BD_DWTG.DBO.WD_MOEDA a12
  on (a11.ID_MOEDA_UG_EXEC_H = a12.ID_MOEDA)
join BD_DWTG.DBO.WD_TAXA_CAMBIO_MENSAL a13
  on (a12.ID_MOEDA = a13.ID_MOEDA_ORIGEM)
join BD_DWTG.DBO.WA_MES_ACUM a14
  on (a11.ID_ANO_LANC = a14.ID_ANO_ACUM_ANO_SALDO and
a11.ID_MES_LANC = a14.ID_MES_ACUM_ANO_SALDO)
join BD_DWTG.DBO.WD_CONTA_CONTABIL_EXERCICIO a15
  on (a11.ID_ANO_LANC = a15.ID_ANO and
a11.ID_CONTA_CONTABIL_LANC = a15.ID_CONTA_CONTABIL)
join BD_DWTG.DBO.WD_MES a16
  on (a14.ID_ANO = a16.ID_ANO and

```

```

a14.ID_MES = a16.ID_MES)
join    BD_DWTG.DBO.WD_NATUREZA_RECEITA2      a17
  on    (a11.ID_ALINEA2_NARE = a17.ID_ALINEA2_NARE and
a11.ID_CATEGORIA_ECONOMICA2_NARE = a17.ID_CATEGORIA_ECONOMICA2_NARE and
a11.ID_ESPECIE_RECEITA2_NARE = a17.ID_ESPECIE_RECEITA2_NARE and
a11.ID_ORIGEM_RECEITA2_NARE = a17.ID_ORIGEM_RECEITA2_NARE and
a11.ID_RUBRICA2_NARE = a17.ID_RUBRICA2_NARE and
a11.ID_SUBALINEA2_NARE = a17.ID_SUBALINEA2_NARE and
a11.ID_TIPO_RECEITA2_NARE = a17.ID_TIPO_RECEITA2_NARE)
where   ((a14.ID_ANO = {}) and a14.ID_MES = {})
  and a11.ID_ORCA_FISCAL_ORGAO_EXEC_H in (0)
  and a15.ID_CONTA_CONTABIL_DESTINO in (621310000, 621320000, 621330000, 621340000, 6213
90000, 621200000)
  and a17.CO_NATUREZA_RECEITA2 like '2121002%'
  and a13.ID_ANO = a16.ID_ANO
  and a13.ID_MES = a16.ID_MES)""""

```

### ###L4+ Mobiliária Externa - Demais Externas

```

query8 = ""select          sum((a11.VA_MOVIMENTO_LIQUIDO * a13.PE_TAXA))  SDCC
from    BD_DWTG.DBO.WF_LANCAMENTO_EP20  a11
  join  BD_DWTG.DBO.WD_MOEDA            a12
    on  (a11.ID_MOEDA_UG_EXEC_H = a12.ID_MOEDA)
  join  BD_DWTG.DBO.WD_TAXA_CAMBIO_MENSAL      a13
    on  (a12.ID_MOEDA = a13.ID_MOEDA_ORIGEM)
  join  BD_DWTG.DBO.WA_MES_ACUM  a14
    on  (a11.ID_ANO_LANC = a14.ID_ANO_ACUM_ANO_SALDO and
a11.ID_MES_LANC = a14.ID_MES_ACUM_ANO_SALDO)
  join  BD_DWTG.DBO.WD_CONTA_CONTABIL_EXERCICIO a15
    on  (a11.ID_ANO_LANC = a15.ID_ANO and
a11.ID_CONTA_CONTABIL_LANC = a15.ID_CONTA_CONTABIL)
  join  BD_DWTG.DBO.WD_MES            a16
    on  (a14.ID_ANO = a16.ID_ANO and
a14.ID_MES = a16.ID_MES)
  join  BD_DWTG.DBO.WD_NATUREZA_RECEITA2      a17
    on  (a11.ID_ALINEA2_NARE = a17.ID_ALINEA2_NARE and
a11.ID_CATEGORIA_ECONOMICA2_NARE = a17.ID_CATEGORIA_ECONOMICA2_NARE and
a11.ID_ESPECIE_RECEITA2_NARE = a17.ID_ESPECIE_RECEITA2_NARE and
a11.ID_ORIGEM_RECEITA2_NARE = a17.ID_ORIGEM_RECEITA2_NARE and
a11.ID_RUBRICA2_NARE = a17.ID_RUBRICA2_NARE and
a11.ID_SUBALINEA2_NARE = a17.ID_SUBALINEA2_NARE and
a11.ID_TIPO_RECEITA2_NARE = a17.ID_TIPO_RECEITA2_NARE)
where   ((a14.ID_ANO = {}) and a14.ID_MES = {})
  and a11.ID_ORCA_FISCAL_ORGAO_EXEC_H in (0)
  and a13.ID_ANO = a16.ID_ANO
  and a13.ID_MES = a16.ID_MES
  and ((a15.ID_CONTA_CONTABIL_DESTINO in (621310000, 621320000, 621330000, 621340000, 62
1390000, 621200000)
  and a17.CO_NATUREZA_RECEITA2 like '2121001%')
  or a15.ID_CONTA_CONTABIL_DESTINO in (896110307, 896110308))""""

```

### ### Contrataual Interna - Abertura de Crédito

```

query9 = ""select          sum((a11.VA_MOVIMENTO_LIQUIDO * a13.PE_TAXA))  SDCC
from    BD_DWTG.DBO.WF_LANCAMENTO_EP20  a11
  join  BD_DWTG.DBO.WD_MOEDA            a12
    on  (a11.ID_MOEDA_UG_EXEC_H = a12.ID_MOEDA)
  join  BD_DWTG.DBO.WD_TAXA_CAMBIO_MENSAL      a13
    on  (a12.ID_MOEDA = a13.ID_MOEDA_ORIGEM)

```

```

join    BD_DWTG.DBO.WA_MES_ACUM a14
  on    (a11.ID_ANO_LANC = a14.ID_ANO_ACUM_ANO_SALDO and
a11.ID_MES_LANC = a14.ID_MES_ACUM_ANO_SALDO)
join    BD_DWTG.DBO.WD_CONTA_CONTABIL_EXERCICIO a15
  on    (a11.ID_ANO_LANC = a15.ID_ANO and
a11.ID_CONTA_CONTABIL_LANC = a15.ID_CONTA_CONTABIL)
join    BD_DWTG.DBO.WD_MES a16
  on    (a14.ID_ANO = a16.ID_ANO and
a14.ID_MES = a16.ID_MES)
join    BD_DWTG.DBO.WD_NATUREZA_RECEITA2 a17
  on    (a11.ID_ALINEA2_NARE = a17.ID_ALINEA2_NARE and
a11.ID_CATEGORIA_ECONOMICA2_NARE = a17.ID_CATEGORIA_ECONOMICA2_NARE and
a11.ID_ESPECIE_RECEITA2_NARE = a17.ID_ESPECIE_RECEITA2_NARE and
a11.ID_ORIGEM_RECEITA2_NARE = a17.ID_ORIGEM_RECEITA2_NARE and
a11.ID_RUBRICA2_NARE = a17.ID_RUBRICA2_NARE and
a11.ID_SUBALINEA2_NARE = a17.ID_SUBALINEA2_NARE and
a11.ID_TIPO_RECEITA2_NARE = a17.ID_TIPO_RECEITA2_NARE)
where  ((a14.ID_ANO = {}) and a14.ID_MES = {})
  and a11.ID_ORCA_FISCAL_ORGAO_EXEC_H in (0)
  and a15.ID_CONTA_CONTABIL_DESTINO in (621310000, 621320000, 621330000, 621340000, 6213
90000, 621200000)
  and a17.CO_NATUREZA_RECEITA2 like '2112001%'
  and a13.ID_ANO = a16.ID_ANO
  and a13.ID_MES = a16.ID_MES)""""

```

### Contratual Interna - Assunção e Reconhecimento de Dívidas (art. 29. LRF) \*\*

```

query10 = """"select      sum((a11.VA_MOVIMENTO_LIQUIDO * a13.PE_TAXA)) SDCC
from    BD_DWTG.DBO.WF_LANCAMENTO_EP02 a11
  join  BD_DWTG.DBO.WD_MOEDA a12
    on  (a11.ID_MOEDA_UG_EXEC_H = a12.ID_MOEDA)
  join  BD_DWTG.DBO.WD_TAXA_CAMBIO_MENSAL a13
    on  (a12.ID_MOEDA = a13.ID_MOEDA_ORIGEM)
  join  BD_DWTG.DBO.WA_MES_ACUM a14
    on  (a11.ID_ANO_LANC = a14.ID_ANO_ACUM_ANO_SALDO and
a11.ID_MES_LANC = a14.ID_MES_ACUM_ANO_SALDO)
  join  BD_DWTG.DBO.WD_ITEM_DECODIFICADO_CCON a15
    on  (a11.ID_ANO_LANC = a15.ID_ANO_ITEM_CONTA and
a11.ID_CONTA_CONTABIL_LANC = a15.ID_CONTA_CONTABIL)
  join  BD_DWTG.DBO.WD_MES a16
    on  (a14.ID_ANO = a16.ID_ANO and
a14.ID_MES = a16.ID_MES)
  join  BD_DWTG.DBO.WD_ITEM_INFORMACAO a17
    on  (a15.ID_ITEM_INFORMACAO = a17.ID_ITEM_INFORMACAO)
where  ((a14.ID_ANO = {}) and a14.ID_MES = {})
  and a11.ID_ORCA_FISCAL_ORGAO_EXEC_H in (0)
  and a13.ID_ANO = a16.ID_ANO
  and a13.ID_MES = a16.ID_MES
  and a17.NO_ITEM_INFORMACAO = 'ASSUNÇÃO 1')""""

```

### Contratual Interna - Demais Internas \*

```

query11 = """"select      sum((a11.VA_MOVIMENTO_LIQUIDO * a13.PE_TAXA)) SDCC
from    BD_DWTG.DBO.WF_LANCAMENTO_EP02 a11
  join  BD_DWTG.DBO.WD_MOEDA a12
    on  (a11.ID_MOEDA_UG_EXEC_H = a12.ID_MOEDA)
  join  BD_DWTG.DBO.WD_TAXA_CAMBIO_MENSAL a13
    on  (a12.ID_MOEDA = a13.ID_MOEDA_ORIGEM)
  join  BD_DWTG.DBO.WA_MES_ACUM a14

```

```

on (a11.ID_ANO_LANC = a14.ID_ANO_ACUM_ANO_SALDO and
a11.ID_MES_LANC = a14.ID_MES_ACUM_ANO_SALDO)
join BD_DWTG.DBO.WD_CONTA_CONTABIL_EXERCICIO a15
on (a11.ID_ANO_LANC = a15.ID_ANO and
a11.ID_CONTA_CONTABIL_LANC = a15.ID_CONTA_CONTABIL)
join BD_DWTG.DBO.WD_MES a16
on (a14.ID_ANO = a16.ID_ANO and
a14.ID_MES = a16.ID_MES)
where ((a14.ID_ANO = {}) and a14.ID_MES = {})
and a11.ID_ORCA_FISCAL_ORGAO_EXEC_H in (0)
and a13.ID_ANO = a16.ID_ANO
and a13.ID_MES = a16.ID_MES
and a15.ID_CONTA_CONTABIL_DESTINO in (212110398))""

```

###L6+ Contratual Externa - Abertura de Crédito - Orçamentárias

```

query12 = ""select sum((a11.VA_MOVIMENTO_LIQUIDO * a13.PE_TAXA)) SDCC
from BD_DWTG.DBO.WF_LANCAMENTO_EP20 a11
join BD_DWTG.DBO.WD_MOEDA a12
on (a11.ID_MOEDA_UG_EXEC_H = a12.ID_MOEDA)
join BD_DWTG.DBO.WD_TAXA_CAMBIO_MENSAL a13
on (a12.ID_MOEDA = a13.ID_MOEDA_ORIGEM)
join BD_DWTG.DBO.WA_MES_ACUM a14
on (a11.ID_ANO_LANC = a14.ID_ANO_ACUM_ANO_SALDO and
a11.ID_MES_LANC = a14.ID_MES_ACUM_ANO_SALDO)
join BD_DWTG.DBO.WD_CONTA_CONTABIL_EXERCICIO a15
on (a11.ID_ANO_LANC = a15.ID_ANO and
a11.ID_CONTA_CONTABIL_LANC = a15.ID_CONTA_CONTABIL)
join BD_DWTG.DBO.WD_MES a16
on (a14.ID_ANO = a16.ID_ANO and
a14.ID_MES = a16.ID_MES)
join BD_DWTG.DBO.WD_NATUREZA_RECEITA2 a17
on (a11.ID_ALINEA2_NARE = a17.ID_ALINEA2_NARE and
a11.ID_CATEGORIA_ECONOMICA2_NARE = a17.ID_CATEGORIA_ECONOMICA2_NARE and
a11.ID_ESPECIE_RECEITA2_NARE = a17.ID_ESPECIE_RECEITA2_NARE and
a11.ID_ORIGEM_RECEITA2_NARE = a17.ID_ORIGEM_RECEITA2_NARE and
a11.ID_RUBRICA2_NARE = a17.ID_RUBRICA2_NARE and
a11.ID_SUBALINEA2_NARE = a17.ID_SUBALINEA2_NARE and
a11.ID_TIPO_RECEITA2_NARE = a17.ID_TIPO_RECEITA2_NARE)
where ((a14.ID_ANO = {}) and a14.ID_MES = {})
and a11.ID_ORCA_FISCAL_ORGAO_EXEC_H in (0)
and a15.ID_CONTA_CONTABIL_DESTINO in (621310000, 621320000, 621330000, 621340000, 6213
90000, 621200000)
and a17.CO_NATUREZA_RECEITA2 like '2122001%'
and a13.ID_ANO = a16.ID_ANO
and a13.ID_MES = a16.ID_MES)""

```

### Contratual Externa - Abertura de Crédito - Extra-Orçamentárias \*

```

query13 = ""select sum((a11.VA_MOVIMENTO_LIQUIDO * a13.PE_TAXA)) SDCC
from BD_DWTG.DBO.WF_LANCAMENTO_EP02 a11
join BD_DWTG.DBO.WD_MOEDA a12
on (a11.ID_MOEDA_UG_EXEC_H = a12.ID_MOEDA)
join BD_DWTG.DBO.WD_TAXA_CAMBIO_MENSAL a13
on (a12.ID_MOEDA = a13.ID_MOEDA_ORIGEM)
join BD_DWTG.DBO.WA_MES_ACUM a14
on (a11.ID_ANO_LANC = a14.ID_ANO_ACUM_ANO_SALDO and
a11.ID_MES_LANC = a14.ID_MES_ACUM_ANO_SALDO)
join BD_DWTG.DBO.WD_CONTA_CONTABIL_EXERCICIO a15

```

```

on (a11.ID_ANO_LANC = a15.ID_ANO and
a11.ID_CONTA_CONTABIL_LANC = a15.ID_CONTA_CONTABIL)
join BD_DWTG.DBO.WD_MES a16
on (a14.ID_ANO = a16.ID_ANO and
a14.ID_MES = a16.ID_MES)
join BD_DWTG.DBO.WD_CONTA_CONTABIL a17
on (a15.ID_CONTA_CONTABIL_DESTINO = a17.ID_CONTA_CONTABIL)
where ((a14.ID_ANO = {}) and a14.ID_MES = {})
and a11.ID_ORCA_FISCAL_ORGAO_EXEC_H in (0)
and a13.ID_ANO = a16.ID_ANO
and a13.ID_MES = a16.ID_MES
and a17.NO_CONTA_CONTABIL = 'EXTRA')""""

```

### *Contratual Externa - Assunção e Reconhecimento de Dívidas (art. 29. LRF) \*\**

```

query14 = """"select      sum((a11.VA_MOVIMENTO_LIQUIDO * a13.PE_TAXA))  SDCC
from    BD_DWTG.DBO.WF_LANCAMENTO_EP02  a11
join    BD_DWTG.DBO.WD_MOEDA  a12
on      (a11.ID_MOEDA_UG_EXEC_H = a12.ID_MOEDA)
join    BD_DWTG.DBO.WD_TAXA_CAMBIO_MENSAL  a13
on      (a12.ID_MOEDA = a13.ID_MOEDA_ORIGEM)
join    BD_DWTG.DBO.WA_MES_ACUM  a14
on      (a11.ID_ANO_LANC = a14.ID_ANO_ACUM_ANO_SALDO and
a11.ID_MES_LANC = a14.ID_MES_ACUM_ANO_SALDO)
join    BD_DWTG.DBO.WD_ITEM_DECODIFICADO_CCON  a15
on      (a11.ID_ANO_LANC = a15.ID_ANO_ITEM_CONTA and
a11.ID_CONTA_CONTABIL_LANC = a15.ID_CONTA_CONTABIL)
join    BD_DWTG.DBO.WD_MES  a16
on      (a14.ID_ANO = a16.ID_ANO and
a14.ID_MES = a16.ID_MES)
join    BD_DWTG.DBO.WD_ITEM_INFORMACAO  a17
on      (a15.ID_ITEM_INFORMACAO = a17.ID_ITEM_INFORMACAO)
where  ((a14.ID_ANO = {}) and a14.ID_MES = {})
and a11.ID_ORCA_FISCAL_ORGAO_EXEC_H in (0)
and a13.ID_ANO = a16.ID_ANO
and a13.ID_MES = a16.ID_MES
and a17.NO_ITEM_INFORMACAO = 'ASSUNÇÃO 2')""""

```

### *Contratual Externa - Demais Externas \*\**

```

query15 = """"select      sum((a11.VA_MOVIMENTO_LIQUIDO * a13.PE_TAXA))  SDCC
from    BD_DWTG.DBO.WF_LANCAMENTO_EP20  a11
join    BD_DWTG.DBO.WD_MOEDA  a12
on      (a11.ID_MOEDA_UG_EXEC_H = a12.ID_MOEDA)
join    BD_DWTG.DBO.WD_TAXA_CAMBIO_MENSAL  a13
on      (a12.ID_MOEDA = a13.ID_MOEDA_ORIGEM)
join    BD_DWTG.DBO.WA_MES_ACUM  a14
on      (a11.ID_ANO_LANC = a14.ID_ANO_ACUM_ANO_SALDO and
a11.ID_MES_LANC = a14.ID_MES_ACUM_ANO_SALDO)
join    BD_DWTG.DBO.WD_MES  a15
on      (a14.ID_ANO = a15.ID_ANO and
a14.ID_MES = a15.ID_MES)
join    BD_DWTG.DBO.WD_NATUREZA_RECEITA  a16
on      (a11.ID_ALINEA_NARE = a16.ID_ALINEA_NARE and
a11.ID_CATEGORIA_ECONOMICA_NARE = a16.ID_CATEGORIA_ECONOMICA_NARE and
a11.ID_ESPECIE_RECEITA_NARE = a16.ID_ESPECIE_RECEITA_NARE and
a11.ID_ORIGEM_RECEITA_NARE = a16.ID_ORIGEM_RECEITA_NARE and
a11.ID_RUBRICA_NARE = a16.ID_RUBRICA_NARE and
a11.ID_SUBALINEA_NARE = a16.ID_SUBALINEA_NARE)

```

```

where ((a14.ID_ANO = {}) and a14.ID_MES = {})
and a11.ID_ORCA_FISCAL_ORGAO_EXEC_H in (0)
and a13.ID_ANO = a15.ID_ANO
and a13.ID_MES = a15.ID_MES
and a16.NO_NATUREZA_RECEITA = 'DEMAIS EXTERNAS')""

```

###Outras Operações Orçamentárias Internas ou Externas (não é possível identificar se c ontratuaais ou mobiliárias)

```

query16 = ""select      sum((a11.VA_MOVIMENTO_LIQUIDO * a13.PE_TAXA))  SDCC
from      BD_DWTG.DBO.WF_LANCAMENTO_EP20  a11
join      BD_DWTG.DBO.WD_MOEDA      a12
on        (a11.ID_MOEDA_UG_EXEC_H = a12.ID_MOEDA)
join      BD_DWTG.DBO.WD_TAXA_CAMBIO_MENSAL      a13
on        (a12.ID_MOEDA = a13.ID_MOEDA_ORIGEM)
join      BD_DWTG.DBO.WA_MES_ACUM a14
on        (a11.ID_ANO_LANC = a14.ID_ANO_ACUM_ANO_SALDO and
a11.ID_MES_LANC = a14.ID_MES_ACUM_ANO_SALDO)
join      BD_DWTG.DBO.WD_CONTA_CONTABIL_EXERCICIO a15
on        (a11.ID_ANO_LANC = a15.ID_ANO and
a11.ID_CONTA_CONTABIL_LANC = a15.ID_CONTA_CONTABIL)
join      BD_DWTG.DBO.WD_MES      a16
on        (a14.ID_ANO = a16.ID_ANO and
a14.ID_MES = a16.ID_MES)
join      BD_DWTG.DBO.WD_NATUREZA_RECEITA2      a17
on        (a11.ID_ALINEA2_NARE = a17.ID_ALINEA2_NARE and
a11.ID_CATEGORIA_ECONOMICA2_NARE = a17.ID_CATEGORIA_ECONOMICA2_NARE and
a11.ID_ESPECIE_RECEITA2_NARE = a17.ID_ESPECIE_RECEITA2_NARE and
a11.ID_ORIGEM_RECEITA2_NARE = a17.ID_ORIGEM_RECEITA2_NARE and
a11.ID_RUBRICA2_NARE = a17.ID_RUBRICA2_NARE and
a11.ID_SUBALINEA2_NARE = a17.ID_SUBALINEA2_NARE and
a11.ID_TIPO_RECEITA2_NARE = a17.ID_TIPO_RECEITA2_NARE)
where ((a14.ID_ANO = {}) and a14.ID_MES = {})
and a11.ID_ORCA_FISCAL_ORGAO_EXEC_H in (0)
and a15.ID_CONTA_CONTABIL_DESTINO in (621310000, 621320000, 621330000, 621340000, 6213
90000, 621200000)
and (a17.CO_NATUREZA_RECEITA2 like '2119001%'
or a17.CO_NATUREZA_RECEITA2 like '2129001%')
and a13.ID_ANO = a16.ID_ANO
and a13.ID_MES = a16.ID_MES)""

```

###L10+ Amortização e Refinanciamento da Dívida

```

query17 = ""select      sum((a11.VA_MOVIMENTO_LIQUIDO * a13.PE_TAXA))  SDCC
from      BD_DWTG.DBO.WF_LANCAMENTO_EP04  a11
join      BD_DWTG.DBO.WD_MOEDA      a12
on        (a11.ID_MOEDA_UG_EXEC_H = a12.ID_MOEDA)
join      BD_DWTG.DBO.WD_TAXA_CAMBIO_MENSAL      a13
on        (a12.ID_MOEDA = a13.ID_MOEDA_ORIGEM)
join      BD_DWTG.DBO.WA_MES_ACUM a14
on        (a11.ID_ANO_LANC = a14.ID_ANO_ACUM_ANO_SALDO and
a11.ID_MES_LANC = a14.ID_MES_ACUM_ANO_SALDO)
join      BD_DWTG.DBO.WD_CONTA_CONTABIL_EXERCICIO a15
on        (a11.ID_ANO_LANC = a15.ID_ANO and
a11.ID_CONTA_CONTABIL_LANC = a15.ID_CONTA_CONTABIL)
join      BD_DWTG.DBO.WD_MES      a16
on        (a14.ID_ANO = a16.ID_ANO and
a14.ID_MES = a16.ID_MES)
where ((a14.ID_ANO = {}) and a14.ID_MES = {})

```

```
and a11.ID_ORCA_FISCAL_ORGAO_EXEC_H in (0)
and a15.ID_CONTA_CONTABIL_DESTINO in (622130300, 622130400, 622130700, 622130600, 6221
30500)
and a11.ID_GRUPO_DESPESA_NADE in (6)
and a13.ID_ANO = a16.ID_ANO
and a13.ID_MES = a16.ID_MES)"""
```

In [20]:

```
for i in range(1,18):
    for i2 in [1,2,3]:
        globals()[f'query{i}_exec_quad{i2}'] = globals()[f'query{i}'].format(ano_atual,
quad_to_mes[globals()[f'quad{i2}']])

for i in range(1,18):
    for i2 in [1,2,3]:
        globals()[f'df_query{i}_quad{i2}'] = pd.read_sql(globals()[f'query{i}_exec_quad
{i2}'], conn)
```

In [21]:

```
for i in range(1,18):
    globals()[f'query{i}_exec'] = globals()[f'query{i}'].format(ano_anterior,12)

for i in range(1,18):
    globals()[f'df_query{i}_ano_anterior'] = pd.read_sql(globals()[f'query{i}_exec'], c
onn)
```

In [65]:

```
for i in range(1,18):
    for i2 in [1,2,3]:
        globals()[f'df_query{i}_quad{i2}'].fillna(0,inplace=True)

for i in range(1,18):
    globals()[f'df_query{i}_ano_anterior'].fillna(0,inplace=True)
```

In [66]:

```
df_rcl = pd.read_excel('rcl.xls')
rcl_quad_atual = df_rcl[(df_rcl['ANO']==ano_atual) & (df_rcl['QUAD']==quadrimestre)]['R
CL']
rcl_ano_anterior = df_rcl[(df_rcl['ANO']==ano_anterior) & (df_rcl['QUAD']==3)]['RCL']
```



In [67]:

```
dic = {}
for i2 in ['ano_anterior', 'quad1', 'quad2', 'quad3']:
    for i in range(1,18):
        globals()[f'query{i}v'] = float(globals()[f'df_query{i}_{i2}']['SDCC'][0]/1000)
        if i2=='ano_anterior':
            globals()[f'rcl_{i2}'] = df_rcl[(df_rcl['ANO']==globals()[f'{i2}']) & (df_rcl['QUAD']==3)].RCL.values[0]
        else:
            globals()[f'rcl_{i2}'] = df_rcl[(df_rcl['ANO']==ano_atual) & (df_rcl['QUAD']==globals()[f'{i2}'])].RCL.values[0]

L3 = query1v + query2v + query3v + query4v + query5v + query6v + query16v
L4 = query7v + query8v

L6 = query12v
L7 = globals()[f'rcl_{i2}']
L8 = 0

L10 = query17v
L11 = 0
L12 = query4v
L13 = 0

L15 = L7 * .6
L16 = 0
L17 = 0

L2 = L3 + L4
L5 = L6
L1 = L2 + L5
L9 = L10 + L12
L14 = L1 + L8 - L9

dic[i2] = [L1,L2,L3,L4,L5,L6,L7,L8,L9,L10,L11,L12,L13,L14,L15]
df_opcred = pd.DataFrame(dic)
```

In [68]:

```
df_opcred
```

Out[68]:

	<b>ano_anterior</b>	<b>quad1</b>	<b>quad2</b>	<b>quad3</b>
<b>0</b>	9.353300e+08	3.538465e+08	6.936098e+08	9.792050e+08
<b>1</b>	9.327568e+08	3.536945e+08	6.929762e+08	9.766453e+08
<b>2</b>	9.278096e+08	3.477398e+08	6.870215e+08	9.579882e+08
<b>3</b>	4.947230e+06	5.954700e+06	5.954700e+06	1.865703e+07
<b>4</b>	2.573182e+06	1.520265e+05	6.335598e+05	2.559767e+06
<b>5</b>	2.573182e+06	1.520265e+05	6.335598e+05	2.559767e+06
<b>6</b>	8.053484e+08	8.186166e+08	8.291573e+08	9.056586e+08
<b>7</b>	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00
<b>8</b>	7.879943e+08	3.762358e+08	5.995496e+08	7.535567e+08
<b>9</b>	7.864021e+08	3.754768e+08	5.987907e+08	7.525345e+08
<b>10</b>	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00
<b>11</b>	1.592191e+06	7.589640e+05	7.589640e+05	1.022219e+06
<b>12</b>	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00
<b>13</b>	1.473357e+08	-2.238923e+07	9.406014e+07	2.256483e+08
<b>14</b>	4.832090e+08	4.911700e+08	4.974944e+08	5.433952e+08

In [69]:

```
df_opcred2 = df_opcred.rename(columns={df_opcred.columns[-2]: 'NO QUADRIMESTRE', df_opcred.columns[-1]: 'ATÉ QUADRIMESTRE'})
df_opcred2['NO QUADRIMESTRE'] = df_opcred2['ATÉ QUADRIMESTRE'] - df_opcred2['NO QUADRIMESTRE']
df_opcred2 = df_opcred2[['NO QUADRIMESTRE', 'ATÉ QUADRIMESTRE']]
```

In [70]:

```
df_opcred2
```

Out[70]:

	<b>NO QUADRIMESTRE</b>	<b>ATÉ QUADRIMESTRE</b>
<b>0</b>	2.855952e+08	9.792050e+08
<b>1</b>	2.836690e+08	9.766453e+08
<b>2</b>	2.709667e+08	9.579882e+08
<b>3</b>	1.270233e+07	1.865703e+07
<b>4</b>	1.926207e+06	2.559767e+06
<b>5</b>	1.926207e+06	2.559767e+06
<b>6</b>	7.650130e+07	9.056586e+08
<b>7</b>	0.000000e+00	0.000000e+00
<b>8</b>	1.540071e+08	7.535567e+08
<b>9</b>	1.537438e+08	7.525345e+08
<b>10</b>	0.000000e+00	0.000000e+00
<b>11</b>	2.632553e+05	1.022219e+06
<b>12</b>	0.000000e+00	0.000000e+00
<b>13</b>	1.315882e+08	2.256483e+08
<b>14</b>	4.590078e+07	5.433952e+08

In [71]:

```
df_opcred2.reset_index()
```

```
L7C1 = float(rc1_quad_atual)
L7C2 = float(df_opcred2.iloc[6][1] / rc1_quad_atual * 100)
L8C1 = df_opcred2.iloc[7][1]
L8C2 = float(df_opcred2.iloc[7][1] / rc1_quad_atual* 100)
L9C1 = df_opcred2.iloc[8][1]
L9C2 = float(df_opcred2.iloc[8][1] / rc1_quad_atual * 100)
L10C1 = df_opcred2.iloc[9][1]
L10C2 = float(df_opcred2.iloc[9][1] / rc1_quad_atual * 100)
L11C1 = df_opcred2.iloc[10][1]
L11C2 = float(df_opcred2.iloc[10][1] / rc1_quad_atual * 100)
L12C1 = df_opcred2.iloc[11][1]
L12C2 = float(df_opcred2.iloc[11][1] / rc1_quad_atual * 100)
L13C1 = df_opcred2.iloc[12][1]
L13C2 = float(df_opcred2.iloc[12][1] / rc1_quad_atual * 100)
L14C1 = df_opcred2.iloc[13][1]
L14C2 = float(df_opcred2.iloc[13][1] / rc1_quad_atual * 100)
L15C1 = df_opcred2.iloc[14][1]
```

In [72]:

```
template_dict = {}
df_opcred2.reset_index()
l1_c1 = df_opcred2.iloc[0][0]
for l in df_opcred2.index:
    for c in range(0,df_opcred2.columns.size):
        template_dict['L'+str(l+1)+'_C'+str(c+1)]=str(locale.currency(round(df_opcred2.
iloc[l][c]) ,grouping=True,symbol=False ))[: -3]
```

In [73]:

```
template_dict['L7C1'] =str(locale.currency(round(L7C1) ,grouping=True,symbol=False ))[: -3]
template_dict['L7C2'] =str(locale.currency(L7C2,symbol=False))
template_dict['L8C1'] =str(locale.currency(round(L8C1) ,symbol=False ))[: -3]
template_dict['L8C2'] =str(locale.currency(L8C2,symbol=False))
template_dict['L9C1'] =str(locale.currency(round(L9C1) ,grouping=True,symbol=False ))[: -3]
template_dict['L9C2'] =str(locale.currency(L9C2,symbol=False))
template_dict['L10C1'] =str(locale.currency(round(L10C1) ,grouping=True,symbol=False ))
[: -3]
template_dict['L10C2'] =str(locale.currency(L10C2,symbol=False))
template_dict['L11C1'] =str(locale.currency(round(L11C1) ,grouping=True,symbol=False ))
[: -3]
template_dict['L11C2'] =str(locale.currency(L11C2,symbol=False))
template_dict['L12C1'] =str(locale.currency(round(L12C1) ,grouping=True,symbol=False ))
[: -3]
template_dict['L12C2'] =str(locale.currency(L12C2,symbol=False))
template_dict['L13C1'] =str(locale.currency(round(L13C1) ,grouping=True,symbol=False ))
[: -3]
template_dict['L13C2'] =str(locale.currency(L13C2,symbol=False))
template_dict['L14C1'] =str(locale.currency(round(L14C1) ,grouping=True,symbol=False ))
[: -3]
template_dict['L14C2'] =str(locale.currency(L14C2,symbol=False))
template_dict['L15C1'] =str(locale.currency(round(L15C1) ,grouping=True,symbol=False ))
[: -3]
```

In [74]:

```
linhas = ['OPERAÇÕES DE CRÉDITO (I)',
        ' Mobiliária',
        ' Interna',
        ' Externa',
        ' Contratual ',
        ' Externa ',
        'RECEITA CORRENTE LÍQUIDA',
        'OPERAÇÕES VEDADAS (II)',
        'OUTRAS OPERAÇÕES DEDUZIDAS DO LIMITE (III)',
        ' Amortização/Refinanciamento do Principal de Dívidas',
        ' Cancelamento de títulos aceitos em leilões de trocas',
        ' Aporte Bacen Lei 11.803/20081',
        ' Concessão de Garantias²',
        'TOTAL CONSIDERADO P/APURAÇÃO DO CUMPR DO LIMITE (IV) = (Ia) + (II) - (III)',
        'LIMITE GERAL DEF P/RESOL DO SENADO P/AS OPER DE CRÉD EXT E INT',]

df_opcred2.index = linhas
df_opcred2.to_excel('df_opcred2.xlsx')
```

In [75]:

```
### Criação de df_assistido, cuja função é alimentar os parágrafos da minuta de instrução assistida
### Cada "Valores" é uma coluna
valores1 = []
valores2 = []
valores3 = []
valores4 = []
valores5 = []
valores6 = []
for x in df_opcred2.reset_index().index:

    valores1.append(df_opcred2.loc[df_opcred2.index[x],df_opcred2.columns[-1]])

    valores2.append(df_opcred2.loc[df_opcred2.index[x],df_opcred2.columns[-2]])

    valores3.append(df_opcred2.loc[df_opcred2.index[x],df_opcred2.columns[-1]]-df_opcred2.loc[df_opcred2.index[x],df_opcred2.columns[-2]])

    valores4.append((df_opcred2.loc[df_opcred2.index[x],df_opcred2.columns[-1]]/df_opcred2.loc[df_opcred2.index[x],df_opcred2.columns[-2]]-1)*100)

    if df_opcred2.loc[df_opcred2.index[x],df_opcred2.columns[-1]]==df_opcred2.loc[df_opcred2.index[x],df_opcred2.columns[-2]]:
        valores5.append('permaneceu inalterado')
    else:
        if df_opcred2.loc[df_opcred2.index[x],df_opcred2.columns[-1]]==df_opcred2.loc[df_opcred2.index[x],df_opcred2.columns[-2]]>0:
            valores5.append('aumentou')
        else:
            valores5.append('diminuiu')

    if df_opcred2.loc[df_opcred2.index[x],df_opcred2.columns[-1]]==df_opcred2.loc[df_opcred2.index[x],df_opcred2.columns[-2]]:
        valores6.append('manutenção de valores')
    else:
        if df_opcred2.loc[df_opcred2.index[x],df_opcred2.columns[-1]]==df_opcred2.loc[df_opcred2.index[x],df_opcred2.columns[-2]]>0:
            valores6.append('crescimento de')
        else:
            valores6.append('redução de')

df_assistido = pd.DataFrame({
                                'quad_atual': valores1,
                                'quad_anterior':valores2,
                                'Variação em Reais': valores3,
                                'Variação percentual': valores4,
                                'verbo':valores5,
                                'substantivo':valores6
                            })

### Atribuição de 0 (zero) sempre que o resultado retornar erro
df_assistido.fillna(0,inplace = True)
df_assistido
```

/usr/lib/python3.6/site-packages/ipykernel\_launcher.py:17: RuntimeWarning:  
invalid value encountered in double\_scalars

Out[75]:

	quad_atual	quad_anterior	Variação em Reais	Variação percentual	verbo	substantivo
0	9.792050e+08	2.855952e+08	6.936098e+08	242.864607	diminuiu	redução de
1	9.766453e+08	2.836690e+08	6.929762e+08	244.290394	diminuiu	redução de
2	9.579882e+08	2.709667e+08	6.870215e+08	253.544621	diminuiu	redução de
3	1.865703e+07	1.270233e+07	5.954700e+06	46.878796	diminuiu	redução de
4	2.559767e+06	1.926207e+06	6.335598e+05	32.891573	diminuiu	redução de
5	2.559767e+06	1.926207e+06	6.335598e+05	32.891573	diminuiu	redução de
6	9.056586e+08	7.650130e+07	8.291573e+08	1083.847358	diminuiu	redução de
7	0.000000e+00	0.000000e+00	0.000000e+00	0.000000	permaneceu inalterado	manutenção de valores
8	7.535567e+08	1.540071e+08	5.995496e+08	389.300036	diminuiu	redução de
9	7.525345e+08	1.537438e+08	5.987907e+08	389.472979	diminuiu	redução de
10	0.000000e+00	0.000000e+00	0.000000e+00	0.000000	permaneceu inalterado	manutenção de valores
11	1.022219e+06	2.632553e+05	7.589640e+05	288.299664	diminuiu	redução de
12	0.000000e+00	0.000000e+00	0.000000e+00	0.000000	permaneceu inalterado	manutenção de valores
13	2.256483e+08	1.315882e+08	9.406014e+07	71.480698	diminuiu	redução de
14	5.433952e+08	4.590078e+07	4.974944e+08	1083.847358	diminuiu	redução de

In [76]:

```
### Total das Operações de Crédito dividido pela RCL  
Z_aux1 = str(locale.currency(L14C2,grouping=True)).replace('R$ ','')  
print(Z_aux1)  
print(L14C2)  
template_dict['Z_aux1'] =(Z_aux1)
```

24,92  
24.915382349435227

In [77]:

```
### Criação de dicionário para alimentação da minuta de instrução assistida
### Z1_1 é a linha 1 e coluna 1, Z5_3 é a linha 5 da coluna 3
### Formatação dos valores da tabela acima.
### As 3 primeiras colunas são números, a quarta é percentual, as duas últimas são text
o.
template_dict_z = {}
strMilhao = ''
strReais = 'R$ '
for l in df_assistido.index:
    if l != 6:
        for c in range(0,df_assistido.columns.size):
            strMilhao=''
            if c < 3:
                if round(abs(df_assistido.iloc[l][c])/1000000 >= 2:
                    strMilhao = ' bilhões'
                    template_dict_z['Z'+str(l+1)+'_'+str(c+1)]=strReais + str(locale.cu
rrency(round(df_assistido.iloc[l][c]/1000000,2) ,grouping=True,symbol=False )) + strMil
hao
                else:
                    if round(abs(df_assistido.iloc[l][c])/1000000 >= 1:
                        strMilhao = ' bilhão'
                        template_dict_z['Z'+str(l+1)+'_'+str(c+1)]=strReais + str(local
e.currency(round(df_assistido.iloc[l][c]/1000000,2) ,grouping=True,symbol=False )) + st
rMilhao
                    else:
                        if round(abs(df_assistido.iloc[l][c])/1000 >= 2:
                            strMilhao = ' milhões'
                            template_dict_z['Z'+str(l+1)+'_'+str(c+1)]=strReais + str(l
ocale.currency(round(df_assistido.iloc[l][c]/1000,2) ,grouping=True,symbol=False )) + s
trMilhao
                        else:
                            if round(abs(df_assistido.iloc[l][c])/1000 >= 1:
                                strMilhao = ' milhão'
                                template_dict_z['Z'+str(l+1)+'_'+str(c+1)]=strReais + s
tr(locale.currency(round(df_assistido.iloc[l][c]/1000,2) ,grouping=True,symbol=False ))
                                + strMilhao
                            else:
                                if abs(df_assistido.iloc[l][c])==0:
                                    template_dict_z['Z'+str(l+1)+'_'+str(c+1)]= ''
                                else:
                                    template_dict_z['Z'+str(l+1)+'_'+str(c+1)]=strReais
                                    + str(locale.currency(round(df_assistido.iloc[l][c]/1000,2) ,grouping=True,symbol=False
                                    )) + strMilhao

            if c == 3:
                if abs(df_assistido.iloc[l][c])==0:
                    template_dict_z['Z'+str(l+1)+'_'+str(c+1)]=''
                else:
                    template_dict_z['Z'+str(l+1)+'_'+str(c+1)]=str(round(df_assistido.i
loc[l][c],2)).replace('.',',')
            if c > 3:
                template_dict_z['Z'+str(l+1)+'_'+str(c+1)]=df_assistido.iloc[l][c]
        else:
            for c in range(0,df_assistido.columns.size):
                if c < 3:
                    template_dict_z['Z'+str(l+1)+'_'+str(c+1)]=str(locale.currency(round(df
_assistido.iloc[l][c],2) ,grouping=True,symbol=False ))
                else:
                    if c == 3:
```

```

        if abs(df_assistido.iloc[1][c])==0:
            template_dict_z['Z'+str(l+1)+'_'+str(c+1)]=''
        else:
            template_dict_z['Z'+str(l+1)+'_'+str(c+1)]=str(round(df_assistido.iloc[1][c],2)).replace('.',',')
        else:
            template_dict_z['Z'+str(l+1)+'_'+str(c+1)]=df_assistido.iloc[1][c]

```

In [78]:

```

### Texto para conclusão da análise, a partir do resultado do resultado acima (Z1_aux, ou df_assistido.iloc[6][0])
if L14C2 < (60 *0.91):
    Z_opcred_conclusao = str('O percentual das Operações de Crédito em relação à RCL foi de '+str(Z_aux1)+' , portanto, inferior a 90% do limite de referência, que é 60% da RCL.')
else:
    if L14C2 <= 60:
        Z_opcred_conclusao = str('O percentual da DCL em relação à RCL foi de '+str(Z_aux1)+' , portanto, ficou entre 90 e 100% do limite de referência. Nessa situação, em atenção ao disposto no art. 59, § 1º da LRF, propõe-se a emissão de alerta ao Poder Executivo, registrando que, nos termos do art. 32, § 1º, inciso III da LRF, a realização de operações de crédito deve observar os limites e condições fixados pelo Senado Federal, ou seja, 60% da Receita Corrente Líquida.')
    else:
        Z_opcred_conclusao = str('O percentual da DCL em relação à RCL foi de '+str(Z_aux1)+' , portanto, superior ao limite de referência. Em virtude disso, propõe-se realizar oitiva dos responsáveis a fim de que expliquem os motivos do descumprimento do art. 32, § 1º, inciso III da Lei de Responsabilidade Fiscal.')

Z_opcred_conclusao

```

Out[78]:

'O percentual das Operações de Crédito em relação à RCL foi de 24,92, portanto, inferior a 90% do limite de referência, que é 60% da RCL.'

In [79]:

```

### Definição de outros campos do template.
template_dict['ano_anterior']=(ano_atual -1)
template_dict['ano_atual']=(ano_atual)
template_dict['quadrimestre']=(quadrimestre)
template_dict['Z_aux1']=(Z_aux1)
template_dict['Z_opcred_conclusao']=(Z_opcred_conclusao)

```



In [80]:

```
### Atualização do template para inserir os campos do tipo "Z".  
template_dict.update(template_dict_z)  
  
### Visualização da lista referente ao template_dict atualizado.  
template_dict
```

Out[80]:

```
{ 'L1_C1': '285.595.248',  
  'L1_C2': '979.205.026',  
  'L2_C1': '283.669.041',  
  'L2_C2': '976.645.259',  
  'L3_C1': '270.966.710',  
  'L3_C2': '957.988.228',  
  'L4_C1': '12.702.331',  
  'L4_C2': '18.657.031',  
  'L5_C1': '1.926.207',  
  'L5_C2': '2.559.767',  
  'L6_C1': '1.926.207',  
  'L6_C2': '2.559.767',  
  'L7_C1': '76.501.298',  
  'L7_C2': '905.658.590',  
  'L8_C1': '0',  
  'L8_C2': '0',  
  'L9_C1': '154.007.086',  
  'L9_C2': '753.556.725',  
  'L10_C1': '153.743.830',  
  'L10_C2': '752.534.506',  
  'L11_C1': '0',  
  'L11_C2': '0',  
  'L12_C1': '263.255',  
  'L12_C2': '1.022.219',  
  'L13_C1': '0',  
  'L13_C2': '0',  
  'L14_C1': '131.588.163',  
  'L14_C2': '225.648.300',  
  'L15_C1': '45.900.779',  
  'L15_C2': '543.395.154',  
  'L7C1': '905.658.590',  
  'L7C2': '100,00',  
  'L8C1': '0',  
  'L8C2': '0,00',  
  'L9C1': '753.556.725',  
  'L9C2': '83,21',  
  'L10C1': '752.534.506',  
  'L10C2': '83,09',  
  'L11C1': '0',  
  'L11C2': '0,00',  
  'L12C1': '1.022.219',  
  'L12C2': '0,11',  
  'L13C1': '0',  
  'L13C2': '0,00',  
  'L14C1': '225.648.300',  
  'L14C2': '24,92',  
  'L15C1': '543.395.154',  
  'Z_aux1': '24,92',  
  'ano_anterior': 2018,  
  'ano_atual': 2019,  
  'quadrimestre': 3,  
  'Z_opcred_conclusao': 'O percentual das Operações de Crédito em relação à  
RCL foi de 24,92, portanto, inferior a 90% do limite de referência, que é  
60% da RCL.',  
  'Z1_1': 'R$ 979,21 bilhões',  
  'Z1_2': 'R$ 285,60 bilhões',  
  'Z1_3': 'R$ 693,61 bilhões',  
  'Z1_4': '242,86',  
  'Z1_5': 'diminuiu',
```

'Z1\_6': 'redução de',  
'Z2\_1': 'R\$ 976,65 bilhões',  
'Z2\_2': 'R\$ 283,67 bilhões',  
'Z2\_3': 'R\$ 692,98 bilhões',  
'Z2\_4': '244,29',  
'Z2\_5': 'diminuiu',  
'Z2\_6': 'redução de',  
'Z3\_1': 'R\$ 957,99 bilhões',  
'Z3\_2': 'R\$ 270,97 bilhões',  
'Z3\_3': 'R\$ 687,02 bilhões',  
'Z3\_4': '253,54',  
'Z3\_5': 'diminuiu',  
'Z3\_6': 'redução de',  
'Z4\_1': 'R\$ 18,66 bilhões',  
'Z4\_2': 'R\$ 12,70 bilhões',  
'Z4\_3': 'R\$ 5,95 bilhões',  
'Z4\_4': '46,88',  
'Z4\_5': 'diminuiu',  
'Z4\_6': 'redução de',  
'Z5\_1': 'R\$ 2,56 bilhões',  
'Z5\_2': 'R\$ 1,93 bilhão',  
'Z5\_3': 'R\$ 633,56 milhões',  
'Z5\_4': '32,89',  
'Z5\_5': 'diminuiu',  
'Z5\_6': 'redução de',  
'Z6\_1': 'R\$ 2,56 bilhões',  
'Z6\_2': 'R\$ 1,93 bilhão',  
'Z6\_3': 'R\$ 633,56 milhões',  
'Z6\_4': '32,89',  
'Z6\_5': 'diminuiu',  
'Z6\_6': 'redução de',  
'Z7\_1': '905.658.589,59',  
'Z7\_2': '76.501.297,52',  
'Z7\_3': '829.157.292,08',  
'Z7\_4': '1083,85',  
'Z7\_5': 'diminuiu',  
'Z7\_6': 'redução de',  
'Z8\_1': '',  
'Z8\_2': '',  
'Z8\_3': '',  
'Z8\_4': '',  
'Z8\_5': 'permaneceu inalterado',  
'Z8\_6': 'manutenção de valores',  
'Z9\_1': 'R\$ 753,56 bilhões',  
'Z9\_2': 'R\$ 154,01 bilhões',  
'Z9\_3': 'R\$ 599,55 bilhões',  
'Z9\_4': '389,3',  
'Z9\_5': 'diminuiu',  
'Z9\_6': 'redução de',  
'Z10\_1': 'R\$ 752,53 bilhões',  
'Z10\_2': 'R\$ 153,74 bilhões',  
'Z10\_3': 'R\$ 598,79 bilhões',  
'Z10\_4': '389,47',  
'Z10\_5': 'diminuiu',  
'Z10\_6': 'redução de',  
'Z11\_1': '',  
'Z11\_2': '',  
'Z11\_3': '',  
'Z11\_4': '',  
'Z11\_5': 'permaneceu inalterado',  
'Z11\_6': 'manutenção de valores',

```
'Z12_1': 'R$ 1,02 bilhão',
'Z12_2': 'R$ 263,26 milhões',
'Z12_3': 'R$ 758,96 milhões',
'Z12_4': '288,3',
'Z12_5': 'diminuiu',
'Z12_6': 'redução de',
'Z13_1': '',
'Z13_2': '',
'Z13_3': '',
'Z13_4': '',
'Z13_5': 'permaneceu inalterado',
'Z13_6': 'manutenção de valores',
'Z14_1': 'R$ 225,65 bilhões',
'Z14_2': 'R$ 131,59 bilhões',
'Z14_3': 'R$ 94,06 bilhões',
'Z14_4': '71,48',
'Z14_5': 'diminuiu',
'Z14_6': 'redução de',
'Z15_1': 'R$ 543,40 bilhões',
'Z15_2': 'R$ 45,90 bilhões',
'Z15_3': 'R$ 497,49 bilhões',
'Z15_4': '1083,85',
'Z15_5': 'diminuiu',
'Z15_6': 'redução de'}
```

In [81]:

```
df_opcred2 = docxtpl.DocxTemplate("templates_rgf/template_opcred.docx")

#renderiza documento
df_opcred2.render(template_dict)
#salva o documento
df_opcred2.save("df_opcred2.docx")
```