UNIVERSIDADE FEDERAL DA PARAÍBA (UFPB)
CENTRO DE CIÊNCIAS SOCIAIS APLICADAS (CCSA)
DEPARTAMENTO DE FINANÇAS E CONTABILIDADE (DFC)
CURSO DE BACHARELADO EM CIÊNCIAS ATUARIAIS (CCA)

JOÃO GUILHERME PEREIRA DOS SANTOS

APLICAÇÃO WEB PARA CRÍTICA DE BASES DE DADOS PREVIDENCIÁRIOS EM FUNDOS DE PENSÃO

JOÃO PESSOA, PB
2025

JOÃO GUILHERME PEREIRA DOS SANTOS

APLICAÇÃO WEB PARA CRÍTICA DE BASES DE DADOS PREVIDENCIÁRIOS EM FUNDOS DE PENSÃO

Trabalho de Conclusão de Curso para o curso de Ciências Atuariais na Universidade Federal da Paraíba, como requesito para a obtenção do título de bacharel em Ciências Atuariais.

Área de concentração: Previdência complementar fechada.

Orientador: Prof. Dr. Luiz Carlos Santos Júnior.

JOÃO PESSOA, PB

2025

Catalogação na publicação
Seção de Catalogação e Classificação

S237a Santos, João Guilherme Pereira dos.
Aplicação web para crítica de bases de dados previdenciários em fundos de pensão / João Guilherme Pereira dos Santos. - João Pessoa, 2025.
100 f. : il.

Orientação: Luiz Carlos Santos Júnior.
TCC (Graduação) - UFPB/CCSA.

1. Previdência complementar fechada. 2. Avaliação atuarial. 3. Qualidade dos dados. 4. Linguagem R. 5. Crítica de base de dados. 6. Shinyapps. I. Santos Júnior, Luiz Carlos. II. Título.

UFPB/CCSA CDU 368

Elaborado por ANA CLAUDIA LOPES DE ALMEIDA - CRB-15/108

JOÃO GUILHERME PEREIRA DOS SANTOS

APLICAÇÃO WEB PARA CRÍTICA DE BASES DE DADOS PREVIDENCIÁRIOS EM FUNDOS DE PENSÃO

Trabalho de Conclusão de Curso (TCC) para o curso de Ciências Atuariais na UFPB, como requisito à obtenção do título de bacharel em Ciências Atuariais.

BANCA EXAMINADORA

Documento assinado digitalmente
gov.br
LUIZ CARLOS SANTOS JUNIOR
Data: 07/05/2025 18:05:57-0300
Verifique em https://validar.iti.gov.br

Prof. Dr. Luiz Carlos Santos Júnior

Orientador

Universidade Federal da Paraíba (UFPB)

Documento assinado digitalmente
gov.br
FILIPE COELHO DE LIMA DUARTE
Data: 07/05/2025 18:09:40-0300
Verifique em https://validar.iti.gov.br

Prof. Dr. Filipe Coelho de Lima Duarte

Membro avaliador

Universidade Federal da Paraíba (UFPB)

Documento assinado digitalmente
gov.br
YURI MARTI SANTANA SANTOS
Data: 07/05/2025 18:34:43-0300
Verifique em https://validar.iti.gov.br

Me. Yuri Martí Santana Santos

Membro avaliador

Fundação de Previdência Complementar do Servidor Público da União (FUNPRESP)

AGRADECIMENTOS

Agradeço, primeiramente, aos meus pais, por todo o apoio, incentivo e amor ao longo dessa caminhada.

Ao professor Luiz, meu orientador, por confiar no meu potencial e por estar sempre disposto a ajudar dentro e fora da sala de aula. Sua didática é excepcional, e me sinto lisonjeado por ter tido a oportunidade de cursar diversas disciplinas sob sua instrução.

À professora Vera, pela generosidade e prontidão em ajudar, sempre atenta às necessidades dos alunos e presente nos momentos em que mais precisei.

Ao professor Herick, que não apenas sugeriu o tema deste trabalho, como também esteve presente em momentos significativos da minha trajetória acadêmica.

A todos os professores que fizeram parte da minha formação, meu sincero reconhecimento. Se eu pudesse voltar no tempo, aproveitaria ainda mais cada aula, cada conversa e cada oportunidade de aprendizado.

À Aline, por quem nutro grande admiração, tanto pelo domínio técnico na área quanto pela generosidade em compartilhar seu conhecimento no dia a dia.

Ao Cleo, pela companhia nos dias de “home office” dentro da universidade e por ser uma presença que me incentivou a continuar nos estudos.

Aos colegas de curso, que compartilharam comigo os desafios e as pequenas vitórias da graduação, meu carinho e gratidão.

RESUMO

A integridade e a consistência das bases cadastrais são fundamentais para garantir a fidedignidade das avaliações atuariais e a sustentabilidade dos planos de benefícios administrados por Entidades Fechadas de Previdência Complementar (EFPC). Neste contexto, este trabalho propõe um processo estruturado de crítica de base de dados utilizando a linguagem R, visando identificar inconsistências, duplicidades e outros aspectos que possam comprometer a qualidade das informações utilizadas na avaliação atuarial. A metodologia adotada combina práticas consolidadas do mercado, incorporando também uma análise descritiva da base de dados e etapas de validação relacionadas à movimentação cadastral. Além da análise técnica, o trabalho contempla a exportação automatizada dos resultados em formato .xlsx e a implementação de uma interface interativa via Shinyapps, com o objetivo de tornar o processo acessível a usuários com diferentes níveis de familiaridade com programação. Os resultados demonstram que a automação e sistematização da crítica de dados pode contribuir significativamente para a melhoria da qualidade das informações utilizadas nas avaliações atuariais, promovendo maior confiabilidade e eficiência na gestão dos planos de benefícios.

Palavras-chave: Previdência Complementar Fechada. Avaliação Atuarial. Qualidade dos Dados. R. Crítica de base de dados. Shinyapps.

ABSTRACT

The integrity and consistency of registry databases are essential to ensure the reliability of actuarial valuations and the sustainability of benefit plans managed by Pension Funds. In this context, this study proposes a structured data review process using the R programming language, aiming to identify inconsistencies, duplications, and other issues that may compromise the quality of information used in actuarial valuations. The adopted methodology combines established market practices, also incorporating a descriptive analysis of the dataset and validation steps related to changes in participant records. In addition to the technical analysis, the study includes the automated export of results in .xlsx format and the implementation of an interactive interface via Shinyapps, with the goal of making the process accessible to users with varying levels of programming experience. The results demonstrate that automating and systematizing the data review process can significantly contribute to improving the quality of information used in actuarial valuations, promoting greater reliability and efficiency in the management of benefit plans.

Keywords: Pension Funds. Actuarial Valuation. Data Quality. R. Data Review. Shinyapps.

LISTA DE QUADROS

LISTA DE TABELAS

LISTA DE FIGURAS

SUMÁRIO

1 INTRODUÇÃO

1.1 CONTEXTO E PROBLEMA

As Entidades Fechadas de Previdência Complementar (EFPC) desempenham um papel fundamental na segurança financeira dos seus participantes, oferecendo planos de benefícios que visam complementar as rendas advindas da Previdência Social - PS (BRASIL, 2021). Para cumprir essa missão, as EFPC contam com uma vasta quantidade de informações e base de dados, que são essenciais para a gestão adequada de seus planos. No entanto, a utilização corriqueira de bases de dados também traz consigo desafios consideráveis, que podem afetar a integridade e a eficácia dessas entidades (PREVIC, 2022).

No final do terceiro trimestre de 2024, o valor total dos ativos das EFPC atingiu a marca de R$ 1.281 trilhões. Dentre essas entidades, 143 apresentaram superávit, 102 estavam em situação deficitária e 27 mantinham um equilíbrio financeiro (PREVIC, 2025). Segundo a Associação Brasileira das Entidades Fechadas de Previdência Complementar (ABRAPP), o valor total dos ativos em domínio dessas entidades representou, em setembro de 2024, 11,3% do Produto Interno Bruto (PIB) referente ao 4º trimestre de 2023 e 1º, 2º, 3º trimestres de 2024.

De acordo com a Previc (2022), os principais riscos atuariais apresentados pelas EFPC são o risco biométrico, o risco de mercado, o risco de liquidez e, por fim, o risco operacional. Este último, inclusive, apresenta relevância especial no contexto dos desequilíbrios técnicos — seja em forma de superávits ou déficits — que podem ser atribuídos não apenas aos riscos atuariais tradicionais, mas também a fatores como a qualidade questionável da base de dados utilizada na avaliação atuarial, a qual pode carecer de confiabilidade e consistência (CORRÊA, 2018). O risco operacional, nesse sentido, refere-se à possibilidade de perdas decorrentes de falhas nos processos, sistemas ou controles internos das EFPC, envolvendo desde a concessão indevida de benefícios — por erros de cálculo ou inconsistências cadastrais — até fraudes ou deficiências na própria condução da avaliação atuarial (RODRIGUES, 2008; PREVCOM, 2021; FERREIRA, 2006). Assim, esse tipo de risco pode estar intrinsecamente ligado aos desequilíbrios técnicos, exigindo uma análise minuciosa e cuidadosa, frequentemente conduzida com o apoio de empresas de consultoria contratadas pela entidade.

Nesse cenário, a atuária Andrea Vanzillotta se destaca ao discutir, em cursos e palestras, a importância da qualidade e da criticidade das bases de dados utilizadas nas avaliações atuariais. Embora suas recomendações ainda não tenham sido formalizadas por órgãos reguladores, como a PREVIC, elas são amplamente reconhecidas por profissionais do setor

como uma valiosa referência para identificar e corrigir falhas nos dados cadastrais. Vanzillotta (2022) enfatiza que o trabalho do atuário deve ter início com a compreensão detalhada do regulamento do plano, visto que é a partir dele que se definem os dados necessários e os procedimentos para seu tratamento. A autora ressalta, ainda, que a análise crítica da base de dados é fundamental para garantir a consistência e a adequação das informações utilizadas nos cálculos atuariais. Para Vanzillotta (2022), o atuário tem a responsabilidade direta de validar a qualidade das informações recebidas, não podendo aceitar os dados fornecidos pelas entidades sem uma verificação rigorosa. Essa validação envolve a aplicação de testes e cruzamentos de dados, a fim de identificar inconsistências e lacunas, garantindo que a modelagem atuarial reflita com precisão as obrigações estabelecidas no regulamento do plano.

Em contraste, observando os procedimentos internacionais, o *The Pensions Regulator* (TPR), órgão do governo do Reino Unido responsável pela supervisão de esquemas de pensão, oferece um modelo que pode servir de referência. O TPR define boas práticas na manutenção de registros, incluindo a revisão periódica de bases de dados e o desenvolvimento de um *Data Improvement Plan* (DIP) para corrigir problemas identificados, conforme orientações disponíveis em seu site institucional (TPR, [202-?]). Além disso, utiliza um sistema de pontuação (*data scoring*) para avaliar a consistência dos dados, focando em *common data* (dados essenciais como nome, data de nascimento e número de seguridade social) e *scheme-specific data* (dados relacionados ao esquema, como contribuições e benefícios). A pontuação envolve a verificação de presença, precisão e formatação dos dados.

Anualmente, o TPR publica um relatório compilando os resultados gerais da qualidade das bases de dados das entidades que submeteram suas pontuações, incentivando melhorias contínuas na gestão das informações (TPR, [202-?]).

A qualidade da base de dados, portanto, não depende apenas de boas práticas regulatórias, mas também do suporte oferecido por ferramentas adequadas à complexidade do sistema previdenciário. Nesse contexto, a linguagem R surge como uma alternativa eficiente, consolidando-se como uma solução viável diante do avanço tecnológico e da crescente demanda por maior eficiência na análise e tratamento de grandes volumes de dados. Sua capacidade de automatizar processos e manipular bases extensas reduz a necessidade de intervenção manual, minimizando erros operacionais e fortalecendo a confiabilidade das informações atuariais.

Além disso, o uso do Shinyapps tem se mostrado uma solução eficaz para ampliar o acesso às análises realizadas no R¸ proporcionando uma nova forma de interação com os dados. Essa ferramenta possibilita a criação de interfaces interativas, permitindo que atuários, gestores

e clientes explorem os resultados de forma intuitiva, sem a necessidade de manipular diretamente scripts ou códigos. Dessa forma, a combinação do R com o Shinyapps oferece um ambiente mais acessível e eficiente para a análise de dados de fundos de pensão brasileiros, que, em alguns casos, ainda enfrentam desafios na adoção de novas tecnologias.

Diante da importância de uma base de dados consolidada e confiável na gestão financeira dessas entidades, bem como na tomada de decisões estratégicas, apresenta-se a questão desta pesquisa: Como o uso do *software* R e da plataforma Shinyapps pode contribuir para a melhoria da análise de dados cadastrais de fundos de pensão no Brasil?

1.2 OBJETIVOS

1.2.1 Objetivo geral

Contribuir para a melhoria da análise de dados cadastrais de fundos de pensão no Brasil por meio do desenvolvimento de uma aplicação web que utiliza a linguagem de programação R e a plataforma Shinyapps.

1.2.2 Objetivos Específicos

- Avaliar criticamente as bases de dados simuladas de fundos de pensão com base em metodologias tradicionais do mercado e sugestões da atuária Andrea Vanzillotta, considerando a movimentação dos dados ao longo do tempo;
- Explorar a estatística descritiva dos dados, fornecendo uma visão geral das informações contidas nas bases de dados;
- Desenvolver uma interface web interativa utilizando a plataforma Shinyapps e a linguagem de programação R, que permita ao usuário realizar a análise e crítica dos dados de forma intuitiva, mesmo sem um conhecimento técnico aprofundado;
- Implementar o processo de exportação dos resultados gerados pela aplicação em formato .xlsx, facilitando a análise posterior dos dados e a integração com outras ferramentas;
- Manter um código estruturado e acessível, com comentários seguindo boas práticas de desenvolvimento, de modo a facilitar a manutenção, reaproveitamento e compreensão por outros interessados.

1.3 JUSTIFICATIVA

As EFPC têm a responsabilidade de oferecer planos de benefícios que visam complementar a previdência social. Neste sentido, a segurança financeira dos participantes está diretamente ligada ao desempenho eficaz dessas entidades na gestão de seus recursos e na concessão de benefícios adequados. Neste sentido, qualquer falha ou imprecisão nas bases de dados usadas pode afetar a qualidade e a sustentabilidade desses benefícios.

O processo para aquisição de base de dados de qualidade não é simples, pois, dentre outras coisas, exige que premissas sejam adotadas e que os dados manuseados estejam alinhados com as necessidades e as características da entidade e de seus participantes.

> Existe uma arte na limpeza de dados e na análise estatística que envolve a aplicação de anos de sabedoria e experiência [...]. Cada conjunto de dados apresenta oportunidades e desafios únicos e a análise estatística não pode ser reduzida a uma abordagem simples baseada em fórmulas (OSBORNE, 2012, p. 11, tradução nossa).

O aprimoramento da crítica das bases de dados, além de mitigar riscos financeiros, pode contribuir para a eficiência operacional das EFPC. A automatização do processo de crítica e a aplicação de técnicas de manipulação de base de dados podem economizar tempo e recursos, tornando a gestão mais eficaz.

Em suma, considerando o papel vital das EFPC na segurança financeira dos participantes e a complexidade dos riscos atuariais envolvidos, a investigação sobre o processo de crítica de bases de dados é essencial para promover uma gestão sólida e eficiente dessas entidades, assegurando benefícios adequados e a sustentabilidade a longo prazo. Portanto, a pesquisa proposta busca apresentar possíveis abordagens para contribuir de forma prática com a crítica de dados em fundos de pensão, por meio do desenvolvimento de uma ferramenta *open source*[1] que, além de apoiar o trabalho técnico e atuarial, poderá auxiliar auditorias independentes e reduzir barreiras para estudantes e profissionais que não possuem expertise na área.

[1] *Open source* refere-se a softwares cujo código-fonte é disponibilizado publicamente, permitindo que qualquer pessoa possa visualizar, modificar e distribuir o conteúdo livremente, promovendo transparência, colaboração e acessibilidade.

2 REFERENCIAL TEÓRICO

2.1 PREVIDÊNCIA COMPLEMENTAR NO BRASIL

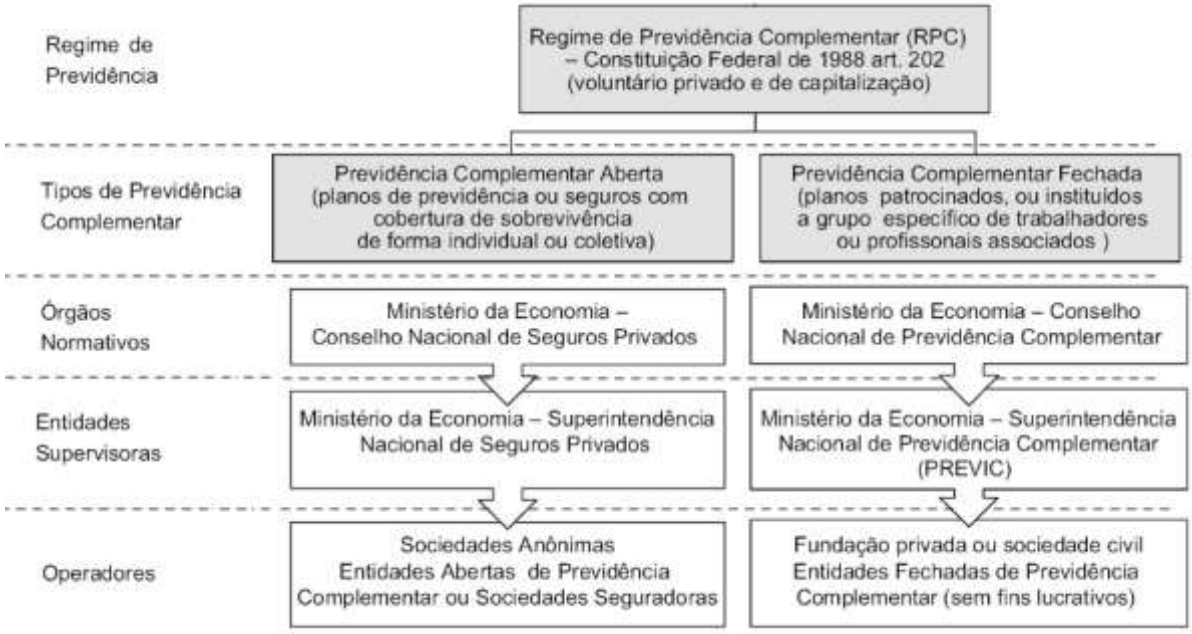
A previdência complementar no Brasil constitui um elemento essencial do sistema de previdência social do país, que, por sua vez, está integrado ao âmbito mais amplo da seguridade social brasileira, tal como estabelecido no Artigo 194 da Constituição de 1988.

O Regime de Previdência Complementar (RPC) é um sistema privado e facultativo que compõe o Sistema Previdenciário Brasileiro juntamente dos regimes públicos, como o Regime Geral de Previdência Social (RGPS) e os Regimes Próprios de Agentes Públicos, também conhecido como Regime Próprio dos Servidores Públicos (RPPS). Enquanto os regimes públicos são compulsórios e geridos pelo Poder Público, o regime complementar é privado, facultativo e gerido por entidades de previdência fiscalizadas pelo governo (LAZZARI *et al.*, 2021). Portanto, o RPC é desvinculado da previdência pública, conforme previsto no artigo 202 da Constituição Federal, e possui regras específicas estabelecidas pelas Leis Complementares n.º 108 e 109, ambas de 29/05/2001, e por demais normativos (BRASIL, 2021).

O RPC se subdivide em dois segmentos principais: o segmento aberto, operado pelas Entidades Abertas de Previdência Complementar (EAPC), e o segmento fechado, administrado pelas Entidades Fechadas de Previdência Complementar (EFPC). De acordo com os dados disponíveis no Ministério da Previdência Social em setembro de 2024, o RPC era composto por 268 entidades no ramo fechado e 43 entidades no ramo aberto.

Cada um dos segmentos do RPC possui suas próprias características e particularidades, sendo regulados e supervisionados por órgãos governamentais específicos. Essa divisão tem o objetivo de garantir o adequado funcionamento e supervisão do sistema de previdência complementar no país. A Figura 1 sintetiza a estrutura geral do RPC.

Figura 1 – Sujeitos referentes ao regime de previdência complementar no Brasil



Fonte: Adaptado de Nese e Giambiagi (2019).

Conforme ilustrado na Figura 1, as entidades abertas e fechadas, embora pertencentes ao mesmo sistema de previdência complementar, possuem órgãos normativos e entidades supervisoras distintos, o que impacta diretamente o seu funcionamento. Como o escopo deste estudo é referente às entidades fechadas, essa diferenciação se faz necessária, ao passo que o presente trabalho busca seguir as práticas atuariais fornecidas pela PREVIC, bem como as normas estabelecidas pelo CNPC.

2.2 EFPC

As EFPC, também conhecidos como fundos de pensão, são instituições de direito privado sem fins lucrativos, cuja principal missão é a gestão de planos coletivos de previdência (NESE; GIAMBIAGI, 2019). A regulamentação e a estruturação desses fundos são estabelecidas pelo Art. 35 da Lei Complementar nº 109/2001, que determina a obrigatoriedade de uma estrutura básica composta por um conselho deliberativo, um conselho fiscal e uma diretoria-executiva (BRASIL, 2022).

A característica fundamental dos Fundos de Pensão é que eles estão acessíveis exclusivamente a grupos de trabalhadores vinculados a empresas específicas ou a órgãos de classe, sejam eles de natureza profissional ou setorial. Além disso, para garantir a transparência e o funcionamento eficiente dessas entidades, é recomendável a elaboração de um regimento interno para o órgão de governança da EFPC. Neste documento, devem constar o maior

detalhamento das atribuições, a periodicidade das reuniões, o controle dos registros em ata para acompanhamento e ajustes, e a identificação dos responsáveis pelas decisões em suas atribuições e devido fundamento (NESE; GIAMBIAGI, 2019).

Conforme mencionado por Nese e Giambiagi (2019), os princípios que determinam o sucesso da governança corporativa em fundos de pensão são semelhantes aos aplicados em empresas. Em ambas as organizações, é essencial que as decisões tomadas pelos gestores estejam em plena conformidade com os interesses dos representantes da entidade, sobre os quais recaem direitos e responsabilidades. Esse alinhamento é fundamental para assegurar a transparência, a responsabilidade e a proteção dos interesses envolvidos, garantindo, assim, a eficácia na gestão e a minimização de riscos.

A estrutura organizacional das EFPC é definida pelo estatuto da entidade, que prevê tanto a composição quanto as atribuições do conselho deliberativo e do conselho fiscal. O conselho deliberativo é responsável pela tomada de decisões estratégicas e de longo prazo, enquanto o conselho fiscal exerce a função de fiscalização e controle interno das atividades da entidade. Além disso, os tipos de benefícios oferecidos, como Benefício Definido (BD), Contribuição Definida (CD) e Contribuição Variável (CV), são regulamentados pelo Art. 7º da Lei Complementar Nº 109, de 29 de maio de 2021. Cada plano de benefício é regido por um regulamento específico que estabelece os direitos e responsabilidades dos participantes, assistidos, patrocinadores e instituidores, bem como as regras de concessão, carência, cálculo e formas de pagamento dos benefícios.

2.3 IMPORTÂNCIA DOS DADOS EM EFPC

No campo atuarial, a relevância dos dados é amplamente reconhecida, sendo destacada por diversos autores e órgãos em seus estudos. Um exemplo claro são os RPPS municipais, que frequentemente enfrentam deficiências significativas em suas bases de dados. Essas falhas não decorrem apenas do tamanho reduzido das bases, mas, principalmente, da falta de manutenção adequada, o que agrava ainda mais a situação. A ausência de um gerenciamento eficaz compromete o trabalho do atuário, dificultando a condução de estudos subsequentes (FERREIRA, 2008; GUIMARÃES, 2012; MINISTÉRIO DA PREVIDÊNCIA SOCIAL, 2009 apud CORRÊA, 2014).

Conforme discutido no capítulo anterior, o gerenciamento eficiente dos dados nas EFPC é de extrema importância, sendo uma prática que não difere significativamente de outros

sistemas previdenciários. Essas instituições são responsáveis pela administração de um volume substancial de ativos financeiros, o que tem impacto direto sobre a massa de seus participantes.

De acordo com o consolidado estatístico da ABRAPP de setembro de 2024, com dados referentes a dezembro de 2023, o número de participantes ativos nas EFPC totaliza 3.005.225, com 4.147.667 dependentes e 867.231 assistidos. Essa ampla base de participantes está distribuída entre diversas entidades e, por conseguinte, está repartida em uma variedade de tipos de planos de benefícios. Adicionalmente, segundo dados da PREVIC extraídos em abril de 2024, o número de participantes ativos é de 3.136.171, havendo ainda 691.870 aposentados e 203.696 beneficiários de pensão.

Nesse contexto, é incumbência das entidades manterem sua base cadastral atualizada. Esse cuidado é essencial para garantir uma interação eficiente com os atuários, profissionais que utilizam esses dados para realizar cálculos e projeções fundamentais à gestão financeira dos planos previdenciários.

A precisão e a integridade dos dados são importantes para as EFPC e os atuários, pois formam a base de cálculos e projeções que asseguram a viabilidade dos planos e o cumprimento dos compromissos com os participantes. Nesse ínterim, a sustentabilidade dos planos é afetada tanto pela realização de um censo previdenciário atualizado – que facilita a tomada de decisões estratégicas, a alocação eficiente de recursos e assegura que as premissas atuariais se fundamentem em informações confiáveis – quanto pela realização de auditorias, que contribuem para validar a qualidade dos dados e a conformidade dos processos utilizados.

A auditoria independente, responsável pela emissão de opinião sobre as demonstrações financeiras da EFPC, deve verificar a consistência das reservas matemáticas apuradas, bem como a adequação das hipóteses atuariais e das bases de dados utilizadas, validando os relatórios que subsidiam a avaliação. Ademais, recomenda-se que esse processo seja periódico, realizado por atuários devidamente habilitados que não tenham elaborado as avaliações atuariais nos exercícios recentes do plano, garantindo maior isenção e confiabilidade no processo. Esses mecanismos de controle são essenciais para assegurar a integridade das informações e reforçar a governança dos planos administrados pelas EFPC (PREVIC, 2022).

2.4 PROCESSO DE AVALIAÇÃO ATUARIAL

No contexto da previdência complementar fechada, a partir do conjunto de dados disponibilizado pela entidade e previamente validado por um atuário, realiza-se o estudo técnico

de adequação das hipóteses atuariais e a avaliação atuarial (GAMA Consultores Associados, 2015). De acordo com a Resolução CNPC nº 30, de 10 de outubro de 2018:

> Art. 2º Para fins desta Resolução, entende-se por:
> I - avaliação atuarial: o estudo técnico desenvolvido por atuário, que deverá ter registro junto ao Instituto Brasileiro de Atuária, que terá por base a massa de participantes, de assistidos e de beneficiários do plano de benefícios de caráter previdenciário, admitidas hipóteses biométricas, demográficas, econômicas e financeiras, e será realizado com o objetivo principal de dimensionar os compromissos do plano de benefícios e estabelecer o plano de custeio de forma a manter o equilíbrio e a solvência atuarial, bem como o montante das reservas matemáticas e fundos previdenciais.

A escolha das premissas atuariais é um processo delicado, pois escolhas inadequadas podem levar a custos incorretos, déficit ou superávit técnico, ou mesmo à incapacidade de pagamento. Por isso, é importante que as premissas sejam escolhidas com bom senso, evitando excessos de margens de segurança ou excessos de riscos (RODRIGUES, 2008).

O estudo técnico de adequação é um instrumento técnico de responsabilidade das EFPC. Esse estudo tem como objetivo principal assegurar que os planos de benefícios de caráter previdenciário gerenciados por essas entidades estejam em conformidade com as hipóteses e projeções financeiras estabelecidas, de forma a garantir a sua viabilidade e a sustentabilidade ao longo do tempo. O estudo deve ser elaborado por um atuário habilitado e legalmente responsável pelo plano de benefícios, utilizando informações fornecidas pela EFPC e pelo patrocinador ou instituidor do plano, sendo especificamente o Administrador Responsável pelo Plano de Benefícios (ARPB) responsável por prover os dados cadastrais e o Administrador Estatutário Tecnicamente Qualificado (AETQ) responsável por prover informações relativas aos investimentos (PREVIC, 2020).

A respeito da obrigatoriedade do estudo técnico de adequação, o Art. 2 da Instrução PREVIC Nº 23, de 26 de junho de 2015, prevê que:

> Art. 2º A comprovação, por meio de estudo técnico, da adequação das hipóteses biométricas, demográficas, econômicas e financeiras às características da massa de participantes e assistidos e do plano de benefícios de caráter previdenciário é exigida para os planos que, independentemente de

> sua modalidade, possuam obrigações registradas em provisão matemática de benefício definido.

Essa obrigatoriedade reforça a relevância de um acompanhamento técnico detalhado para os planos de benefício definido, uma vez que esse tipo de plano está sujeito a riscos atuariais mais complexos, conforme será explorado no próximo tópico.

No processo de avaliação atuarial, conforme descrito anteriormente, a validade e precisão dos dados utilizados no estudo técnico de adequação são essenciais para garantir que as hipóteses atuariais reflitam de maneira precisa a realidade do plano de benefícios. De acordo com a Instrução Previc nº 33, de 23 de outubro de 2020, o estudo técnico deve possuir um atestado de validação, expedido pelo Administrador Responsável pelo Plano de Benefícios (ARPB), para os dados cadastrais e demais informações referentes ao passivo atuarial.

Essa exigência destaca a relevância da etapa de "Crítica e Manipulação de Dados", uma vez que é nesta fase que se busca assegurar a consistência e integridade dos dados fornecidos. A crítica de dados, portanto, vai além de uma simples verificação de erros, ela é um componente fundamental para garantir que o estudo atuarial seja baseado em informações confiáveis, evitando distorções que possam comprometer a solvência e o equilíbrio do plano de benefícios.

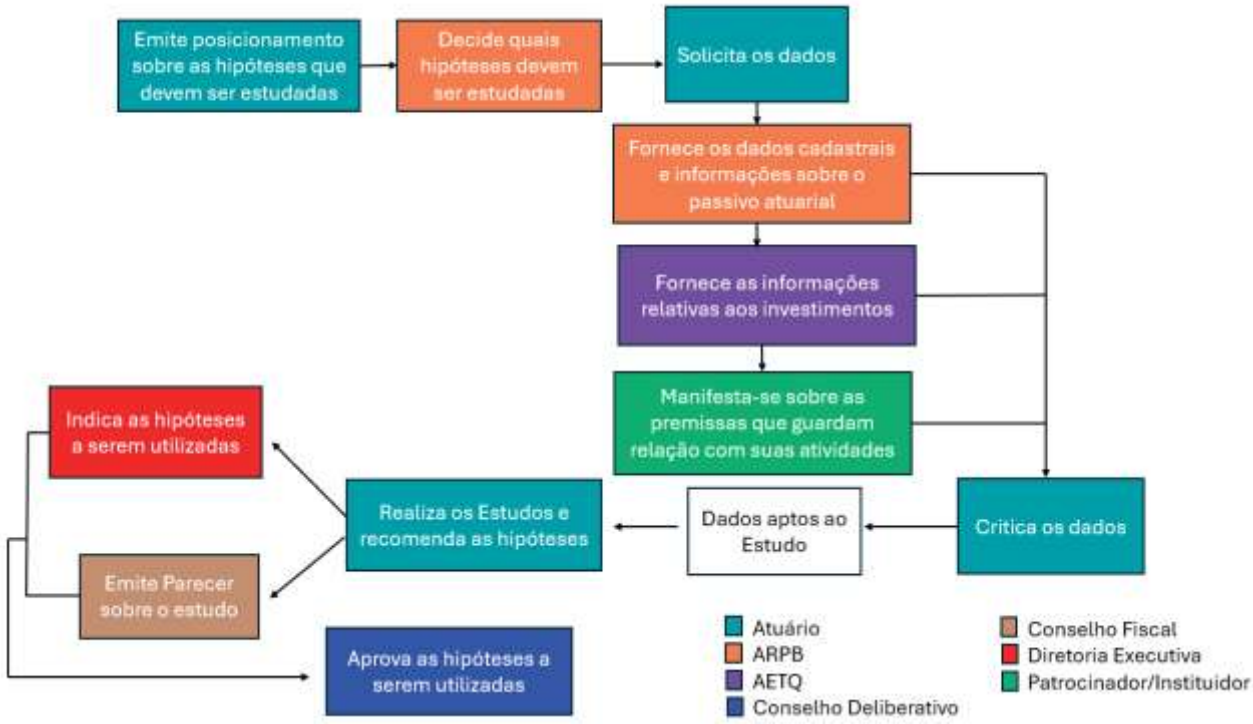
Portanto, antes da realização da avaliação atuarial propriamente dita, é imprescindível que os dados passem por um processo rigoroso de crítica e validação, conforme preconizado pelas normativas vigentes. Somente com uma base de dados sólida e validada é possível prosseguir com as análises atuariais necessárias para o dimensionamento preciso dos compromissos do plano e a definição de um plano de custeio adequado, como será discutido nos próximos tópicos.

2.5 CRÍTICA E MANIPULAÇÃO DE BASE DE DADOS

A fase de crítica da base de dados é uma etapa da avaliação atuarial realizada pelo atuário. Para que o atuário possa realizar uma avaliação precisa, é necessário que ele trabalhe com uma base de dados atualizada e confiável. Para isso, o atuário deve realizar um processo de crítica das bases de dados da entidade recebidas do ARPB. Se o atuário identificar inconsistências na base de dados, ele deverá encaminhá-las para a entidade, que será responsável por ratificá-los ou retificá-los (PREVIC, 2022).

A Figura 2 expressa bem o papel do processo de crítica da base de dados no processo de elaboração das hipóteses atuariais, com os respectivos agentes que fazem parte do processo:

Figura 2 – Procedimentos necessários para o estabelecimento de hipóteses atuariais



Fonte: Adaptado de GAMA Consultores Associados (2015).

A partir das etapas apresentadas, podemos concluir que a base de dados é o principal insumo utilizado pelo atuário para conduzir os estudos subsequentes. Sua importância é evidente, pois se a base de dados não refletir corretamente o perfil dos participantes, os resultados emitidos pelo atuário poderão estar equivocados. Embora, inicialmente, essa incongruência possa parecer pouco significativa, com o passar do tempo, os impactos podem se tornar substanciais, como no caso de um benefício vitalício, que, se calculado incorretamente, pode resultar em perdas consideráveis tanto para o beneficiário quanto para a entidade. Nesse sentido, conforme reforça Vanzillotta (2022), é essencial que o atuário valide a base de dados com apoio do regulamento do plano, complementando com os testes que julgar necessários até que se sinta seguro quanto à fidedignidade das informações utilizadas.

2.6 ESTUDOS CORRELATOS

No contexto atuarial brasileiro, ainda são escassas as referências práticas sobre a crítica e validação de bases de dados aplicadas à previdência, especialmente quando associadas ao uso de linguagens de programação. Nesse cenário, destaca-se a contribuição de Vanzillotta (2022), que propõe uma metodologia estruturada de validação de dados, dividida em três etapas principais. A primeira etapa consiste na análise individual dos registros, com foco na

identificação de campos vazios, formatos inválidos ou registros duplicados. A segunda etapa envolve a verificação de dados com base em limites predefinidos, como idade fora de intervalos esperados, salários nulos, abaixo do mínimo ou acima de um teto estipulado, além de inconsistências em datas — por exemplo, concessão de benefício posterior à data atual. Por fim, a terceira etapa contempla a comparação com a base de dados do ano anterior, permitindo a identificação de entradas, saídas e permanências, o que auxilia na detecção de alterações relevantes na composição da massa de participantes. Apesar dessa abordagem prática, os órgãos reguladores no Brasil ainda não estipulam um procedimento padronizado a ser seguido pelo atuário em relação à crítica de dados.

Por esse motivo, é necessário buscar tais tipos de informações em estudos internacionais ou em áreas de estudo correlatas que também dependem de uma boa consistência de base cadastral. O presente tópico não tem como intenção copiar os modelos já utilizados, mas referenciá-los, uma vez que é importante investigar e disponibilizar este tipo de informação no meio atuarial, ainda mais quando se utilizam ferramentas de linguagem de programação, cuja adoção ainda não é amplamente discutida na literatura ou evidenciada em práticas divulgadas por consultorias e auditores independentes.

O manual da *Society of Actuaries* (SOA, 2011), *Experience Data Quality: How to Clean and Validate Your Data*, apresenta uma série de procedimentos que o atuário pode adotar, com foco no setor de seguros de vida. O objetivo é orientar a limpeza e a validação de dados, identificando erros comuns, suas causas, métodos de detecção e soluções. O documento também destaca o impacto negativo que dados imprecisos podem ter no cálculo de prêmios, no design de produtos e, consequentemente, na rentabilidade e solvência das empresas. De acordo com o manual, uma base de dados de baixa qualidade resultará em estudos imprecisos, o que poderá gerar as seguintes implicações: risco para lucratividade; anti-seleção de tipos de risco; resseguro inapropriado; risco de adequação de reservas; e risco para avaliação da empresa.

O objetivo dos estudos de experiência (*experience studies*) é desenvolver taxas ou probabilidades baseadas em dados coletados de fontes como companhias de seguros, empresas de consultoria ou órgãos governamentais. Essas taxas e probabilidades podem ser utilizadas por atuários para desenvolver premissas para planejamento financeiro, avaliação, precificação e gestão de riscos. Neste contexto, e devido à exposição das seguradoras a diversos tipos de risco, a SOA (Society of Actuaries) destaca a importância da limpeza (*data cleansing*) e validação de dados (*data validation*) para a realização de estudos precisos no campo dos seguros de vida. Esses processos são fundamentais para garantir a qualidade e a confiabilidade dos dados

utilizados nas análises atuariais, permitindo que os atuários tomem decisões informadas e desenvolvam estratégias eficazes para mitigar riscos e otimizar o desempenho das seguradoras.

Embora existam diversas ferramentas no mercado voltadas para a limpeza e validação de dados, o manual da SOA destaca que, em geral, essas soluções não são adequadas para o setor de seguros de vida, devido à sua especificidade. Em resposta a essa necessidade, a SOA desenvolveu a "*SOA Data Quality Tool*", uma metodologia específica para o tratamento de dados nesse segmento. Seu objetivo é detectar erros comuns nas bases de dados e oferecer diferentes tipos de visualizações que facilitam a identificação de erros mais complexos pelos analistas.

O processo de qualidade de dados apresentado pela SOA baseia-se em dois pilares principais: limpeza de dados (data cleansing) e validação de dados (data validation). Embora estejam inter-relacionados, esses conceitos possuem distinções importantes. O processo de limpeza de dados concentra-se na identificação e correção de erros básicos e inconsistências nos dados brutos, como a remoção de valores ausentes ou a correção de formatação inadequada, estabelecendo assim uma base sólida para análises posteriores. Por exemplo, em uma base de dados de participantes de um fundo de pensão, a limpeza pode envolver a remoção de registros com campos obrigatórios em branco ou a padronização de formatos de data. Já a validação de dados foca na conformidade dos dados, garantindo que os valores estejam corretos, dentro do esperado e adequados para o contexto específico em que serão utilizados. Em um caso prático, a validação pode envolver a verificação de que o campo de idade do participante não tenha valores negativos ou superiores a 120 anos, conforme os critérios definidos no regulamento do plano. Essas etapas são fundamentais para garantir que os dados utilizados em cálculos atuariais sejam precisos e confiáveis.

A SOA identifica três tipos comuns de erros: campos não preenchidos (*Missing Value Errors),* formato de dados inadequado (*Data Format Errors*) e erros de codificação (*Coding Errors*). Para exemplificar, campos não preenchidos podem resultar em valores 'NA' (*not available);* erros de formato ocorrem quando tipos de variáveis são usados de forma indevida, como utilizar uma variável textual em um campo destinado a formato numérico; já os erros de codificação ocorrem quando os valores inseridos em determinados campos não seguem o padrão esperado, como o campo de sexo do participante preenchido com algo diferente de 'M' ou 'F'.

A primeira técnica para identificar erros, conforme apresentada no manual da SOA, é a Automação (*Automation*), amplamente utilizada por meio de linguagens de programação. Embora essa técnica exija um tempo considerável no início, sua manutenção é relativamente

simples. No entanto, a automação não é indicada para dados incertos ou mutáveis. Essa técnica é a que melhor se adequa à proposta deste trabalho.

A segunda técnica para detectar erros é conhecida como '*100 Records*'. Ela consiste em inspecionar visualmente os 100 primeiros registros da base de dados. Embora seja uma abordagem eficiente para identificar grandes erros ou padrões de falhas, sua aplicação em bases de dados extensas é limitada. Além de ser suscetível a erros humanos na detecção, essa técnica pode demandar muito tempo do analista, se tornando inapropriada para conjuntos de dados maiores.

A terceira técnica apresentada é a distribuição de frequência (*Frequency Distribution*). Em bases de dados numéricas com grandes volumes de amostras, a aplicação de estatísticas pode ser eficaz na identificação de erros. Dependendo da natureza da variável, é possível agrupar os dados em classes para uma análise mais precisa. Essa abordagem pode economizar tempo, mas o atuário deve dosar entre a precisão dos resultados e o tempo dedicado à análise.

Por fim, a última técnica apresentada é intitulada como *Sampling*, ou Amostragem, essa abordagem é uma alternativa indicada para uma base de dados considerável, dentro dessa técnica temos métodos como: *simple random sampling*, *stratification*, *systemic* e *cluster sampling*, os quais não serão explorados neste trabalho. Tais técnicas geralmente são utilizadas quando há restrições de tempo ou recursos computacionais que impedem a análise completa de grandes volumes de dados. No entanto, no contexto deste trabalho, a proposta é realizar uma crítica automatizada e completa da base de dados dos fundos de pensão, garantindo que todas as inconsistências sejam verificadas sem a necessidade de seleção amostral.

Quanto ao processo de validação de dados (*data validation*), o manual da SOA estabelece três fases principais. A primeira delas é a consistência dos dados (*consistency*), que busca identificar variações inesperadas ao longo do tempo, garantindo que as informações analisadas apresentem um comportamento coerente entre diferentes estudos. Em seguida, a razoabilidade (*reasonability*) avalia se os resultados obtidos são compatíveis com padrões esperados, detectando possíveis erros por meio da análise de tendências e variações abruptas. Por fim, a completude (*completeness*) assegura que os dados utilizados sejam suficientes e confiáveis para a análise, evitando a exclusão indevida de informações que possam comprometer a credibilidade dos resultados.

Além das técnicas de detecção de erros, o manual da SOA também apresenta um conjunto de soluções possíveis para lidar com inconsistências encontradas nas bases de dados. Essas soluções variam conforme o tipo e a gravidade do erro identificado, assim como a quantidade de tempo e recursos disponíveis. São elas: deleting (exclusão do dado inconsistente),

ignoring (manutenção do dado na base, mas sem utilizá-lo nas análises relevantes em que esteja envolvido), correcting (correção dos dados) e filling in (preenchimento de dados ausentes ou incorretos com base em informações de outras variáveis). Essas abordagens demonstram que, uma vez detectadas, as inconsistências podem ser tratadas com critérios técnicos adequados. Embora este trabalho não tenha aplicado essas soluções, por se concentrar apenas na análise crítica dos dados, a menção a essas alternativas é relevante para destacar as possíveis abordagens que poderiam ser adotadas.

A metodologia da SOA não foi utilizada diretamente, pois foi empregada em um escopo distinto. No entanto, a ideia geral da fase de consistência, assim como a técnica de automação, está presente neste estudo por meio dos testes lógicos empregados para identificar erros. Além disso, o princípio da razoabilidade foi incorporado ao comparar duas bases de dados distintas, visando assegurar que os registros apresentem um padrão consistente.

Complementarmente, podemos citar alguns pacotes do R que têm como proposta a validação e limpeza de dados. Entre eles, destacam-se o *janitor* e o *data.validator*. O pacote *janitor* oferece ferramentas simples para a limpeza de dados, incluindo funções para remover colunas vazias, padronizar nomes de variáveis e realizar tabelas de frequência de forma eficiente. Já o pacote *data.validator* fornece um conjunto de funções para validar dados de acordo com regras predefinidas, permitindo aos usuários definir e aplicar critérios específicos de validação, facilitando assim a detecção de inconsistências e erros nos dados. Esses pacotes podem ser valiosos no processo de preparação e verificação de dados, embora não se diferenciem muito do que pode ser feito manualmente, com um pouco mais de esforço.

Neste compêndio, o foco principal será o processo de validação de dados, conforme descrito no material da SOA, embora aspectos relacionados à limpeza de dados possam ser abordados dependendo do desenvolvimento do trabalho. É importante ressaltar que, apesar da ênfase na validação, os erros comumente identificados durante o processo de limpeza de dados não serão negligenciados. Muitos desses erros, relacionados à natureza dos dados, serão detectados pelo software R, o que poderá impedir a execução completa do programa. Ao longo do desenvolvimento, pretende-se realizar esses apontamentos de maneira integrada, sem interromper o fluxo da execução do programa. Essa abordagem garante que problemas fundamentais na qualidade dos dados sejam identificados precocemente, mesmo que não sejam o foco principal do processo de validação.

O processo e as técnicas de validação foram implementados com base no método de automação descrito pela SOA (2011). Os critérios adotados para validar os dados foram definidos conforme a adequação ao conjunto de dados utilizado e serão detalhados na seção de

metodologia. A abordagem busca ser suficientemente genérica para atender às expectativas das entidades de fundos de pensão. No entanto, é importante reforçar a necessidade de adaptar esses critérios às particularidades de cada entidade.

3 METODOLOGIA

Método é o conjunto de atividades que possibilita a produção de conhecimentos válidos; é o caminho percorrido pelo pesquisador (MARCONI; LAKATOS, 2017). Neste capítulo, são abordados o tipo de pesquisa realizada, a base de dados utilizada, os métodos de análise empregados e os pacotes da linguagem R que auxiliaram na execução das análises.

Inicialmente, é apresentado o processo de crítica da base de dados, que combina aspectos tradicionalmente avaliados no mercado com as sugestões propostas por Vanzillotta (2022), nos quais também foram comtemplados aspectos referentes à movimentação da base de dados. Adicionalmente, exploramos a estatística descritiva da base como recurso complementar para compreender padrões e características gerais dos dados.

Por fim, detalha-se o processo de exportação dos resultados, que são gerados em formato .xlsx. Para tornar a aplicação dos métodos mais acessível ao usuário, foi desenvolvida uma interface web utilizando a linguagem de programação R, com suporte da plataforma Shinyapps. Vale destacar que, embora os próximos tópicos apresentem trechos do código utilizados na implementação do servidor e da interface do usuário, além do uso de tratamentos de erros para uma melhor experiência do usuário - o que pode causar um pouco de estranheza para usuários habituais do R -, a explicação textual de cada parte do código por meio dos comentários, garante sua compreensão sem necessidade de um conhecimento técnico aprofundado.

3.1 TIPO DE PESQUISA

A coleta de dados é um elemento fundamental em qualquer pesquisa, independentemente dos métodos ou técnicas utilizados. Esta fase de levantamento de informações é crucial para estabelecer uma base sólida de conhecimento sobre o campo de interesse (MARCONI; LAKATOS, 2017). No contexto específico deste trabalho, essa importância se traduz na análise criteriosa de dados atuariais, formando o suporte principal da metodologia empregada.

A pesquisa aqui apresentada, adota predominantemente uma abordagem quantitativa, focada na análise rigorosa de dados cadastrais de fundos de pensão brasileiros, mais especificamente, nos dados referentes a massa de aposentados. O cerne do trabalho é o desenvolvimento de um script computacional que busca complementar as funcionalidades do Excel, suprindo suas limitações na manipulação de grandes volumes de dados e reduzindo a ocorrência de erros operacionais durante o processo da metodologia, mantendo, contudo, um

relatório em formato .xlsx para facilitar a visualização dos achados. Dessa forma, a proposta não substitui o uso do Excel, mas amplia suas capacidades ao oferecer uma análise mais precisa e ágil das bases de dados atuariais. Esta perspectiva alinha-se com o método científico contemporâneo, que, como discutido por Severino (2013), fundamenta-se em parâmetros quantitativos essenciais para a formulação de relações funcionais e para a compreensão aprofundada dos fenômenos estudados.

Adicionalmente, esta pesquisa tem um caráter aplicado, direcionado à solução de desafios práticos no campo atuarial. O trabalho visa criar ferramentas concretas que possam ser utilizadas diretamente pelos profissionais da área, auxiliando na resolução de problemas cotidianos e promovendo melhorias no processo de análise de dados.

Apesar das vantagens significativas proporcionadas pelo *script*, como a otimização do tratamento de grandes volumes de dados, ele ainda está sujeito a desafios semelhantes aos enfrentados por outras metodologias, como o Excel. Entre esses desafios, destaca-se a necessidade de constante atualização, especialmente em resposta às mudanças regulatórias no setor e às diretrizes específicas de cada entidade, como também destacado pela metodologia empregada pelo SOA, conforme discutido no tópico anterior.

Dessa forma, espera-se que este trabalho contribua não apenas para a prática profissional, mas também para o avanço do conhecimento na área atuarial. Ele promove uma reflexão crítica sobre a importância da tecnologia na análise de dados atuariais e ressalta a necessidade fundamental de uma base de dados consistente e confiável. Ao fazê-lo, o estudo busca preencher uma lacuna importante entre a teoria atuarial e sua aplicação prática, potencialmente abrindo caminho para futuras inovações no campo.

3.2 BASE DE DADOS E PARÂMETROS

Para a realização desta pesquisa, buscou-se inicialmente obter dados reais de fundos de pensão. Diversas entidades foram contatadas com o intuito de coletar uma amostra representativa para a realização da análise. No entanto, o processo de obtenção de dados enfrentou desafios significativos:

- Resposta limitada: Apesar dos esforços de contato, apenas uma entidade forneceu um retorno que poderia ser considerado para o estudo.
- Insuficiência de dados: A base de dados fornecida, embora útil, não apresentou a abrangência necessária para atender plenamente aos objetivos da pesquisa. Em

particular, as variáveis disponibilizadas foram insuficientes para realizar uma crítica robusta, o que impediu a construção de uma análise sólida sobre as bases de dados dos fundos de pensão.

Diante dessas limitações, adotou-se uma estratégia complementar para a composição da base de dados, combinando dados reais e simulação. A base de dados real foi expandida com um número maior de participantes, buscando fornecer maior diversidade de situações a serem analisadas, permitindo a observação de um espectro mais amplo de possíveis inconsistências.

Complementarmente, é importante destacar que, na construção da base simulada, não foram empregados métodos sofisticados de imputação de dados. Essa escolha metodológica foi intencional, uma vez que o objetivo da simulação era, sobretudo, reproduzir de forma de forma suficientemente representativa as características gerais de uma base de dados de aposentados, sem a pretensão de refletir fielmente a complexidade estatística de uma população real.

Para a geração das matrículas dos participantes, utilizou-se uma sequência numérica crescente apenas com a finalidade de identificação. No que tange aos dados pessoais, como data de nascimento, foram definidos limite inferior e superior com base no participante mais jovem e mais velho da base real, dentro dos quais foram sorteadas datas aleatórias de nascimento, utilizando a função =DATA(ALEATÓRIOENTRE(min;max); ALEATÓRIOENTRE(1;12); ALEATÓRIOENTRE(1;28)) garantindo uma distribuição etária coerente com o perfil de aposentados da base real. O campo de sexo foi definido com base em uma distribuição uniforme contínua por meio da função =SE(ALEATÓRIO()<0,5;"M";"F"), o que também foi replicado para os dependentes. Para os valores de benefício complementar, aplicou-se a função =ALEATÓRIOENTRE(min;max), utilizando como referência os valores mínimo e máximo observados na base real. Os tipos de benefício foram atribuídos aleatoriamente entre três categorias fictícias com a função =ÍNDICE({"TIPO 1";"TIPO 2";"TIPO 3"}; ALEATÓRIOENTRE(1;3)). A base de dados atual, por sua vez, foi construída a partir da base anterior, com alterações manuais intencionais para possibilitar a avaliação do desempenho do script na identificação de inconsistências, como inclusão ou remoção de registros e modificações nos valores de benefício — algumas em conformidade com o reajuste esperado, outras propositalmente inconsistentes, com o intuito de testar as críticas do modelo.

Esta metodologia híbrida, combinando informações dos dados reais para construção dos dados simulados, foi considerada a mais apropriada para prosseguir com a pesquisa, permitindo uma análise mais abrangente e adequada para que o intuito de generalização do modelo fosse facilitado, apesar das dificuldades iniciais na obtenção de dados do setor.

A base de dados em fundos de pensão geralmente é subdividida em três categorias distintas: Ativos, Aposentados e Pensionistas. Essa segmentação é fundamental para a análise, pois cada grupo possui características e necessidades específicas no contexto dos fundos de pensão. No entanto, a análise desta pesquisa concentrou-se na categoria de Aposentados, pois a crítica da base de dados desse grupo já reflete, de maneira geral, o que seria feito nas bases de pensionistas e ativos. Como o objetivo do estudo é comparar diferentes bases, verificando as movimentações e variações de benefícios envolvidas, a base dos Aposentados se mostrou a mais adequada para desenvolver o modelo abrangente proposto no objetivo do trabalho. Além disso, considerando que a análise da massa de aposentados pode ser extrapolada para os demais grupos, optou-se por focar nesse segmento, sendo possível, para desenvolvimentos futuros, realizar críticas específicas sobre outros tipos de massa.

3.3 PROCESSO DE ANÁLISE DE DADOS

Neste trabalho se realizam duas análises distintas. A primeira, e mais relevante, consiste na crítica da base de dados, abordada em tópicos anteriores. A segunda análise é descritiva e foca na massa de participantes aposentados, demandando menos esforço, mas também apresentando importância significativa.

Este tópico tem como objetivo detalhar o processo utilizado para a construção do script em linguagem R, abordando os pontos essenciais para o seu funcionamento e enfatizando a importância de garantir a integridade dos dados durante a análise. O modelo desenvolvido tem a finalidade de identificar e corrigir inconsistências nos dados, assegurando a qualidade e a confiabilidade das informações. Essa abordagem não apenas minimiza o risco operacional, comum em modelos tradicionalmente utilizados em Excel, mas também oferece uma solução prática para uma atividade rotineira dos fundos de pensão no Brasil. A metodologia implementada foi projetada para detectar discrepâncias e lacunas nos dados, facilitando a sua correção e permitindo que os usuários tomem decisões informadas com base em dados consistentes. Ajustes de menor relevância feitos ao longo do desenvolvimento do script, assim como arquivos auxiliares responsáveis pela construção da aplicação web, foram omitidos neste texto, mas podem ser verificados no apêndice e também no repositório público disponível na plataforma GitHub, através do seguinte link: https://github.com/jgpds/guisa. Além do código-fonte e dos arquivos complementares, o repositório também contém uma breve apresentação em vídeo, que ilustra o funcionamento da aplicação, facilitando a sua compreensão.

3.3.1 Pacotes utilizados no R

Para a crítica e análise dos dados na linguagem de programação R, com exceção dos pacotes utilizados na construção da aplicação web com Shinyapps – os quais não são essenciais para a realização da crítica –, foram empregados os seguintes pacotes:

- *Rbcb* (FREITAS, 2024): O pacote rbcb permite acessar diversos conjuntos de dados fornecidos pela API do Banco Central do Brasil, relacionados à atividade econômica, por meio de suas funções e estruturas de dados compatíveis com o R. No presente trabalho, foi essencialmente utilizado para identificar variações indevidas no benefício complementar, com base em um indexador específico e uma data de reajuste hipotética.
- *Openxlsx* (SCHAUBERGER; WALKER, 2025): Utilizado para a criação e formatação de planilhas Excel. Nesta análise, foi empregado para gerar e estilizar o relatório em Excel, resultante da análise crítica realizada no R.
- *Readxl* (WICKHAM; BRYAN, 2023): O pacote readxl no R é utilizado para importar arquivos Excel (.xls e .xlsx) diretamente para o ambiente R, e esse foi o principal propósito de sua aplicação neste trabalho.
- *Tidyverse (dplyr)* (WICKHAM et al., 2019): Esse pacote é amplamente utilizado para manipulação eficiente de dados. No contexto deste trabalho, foi empregado com o mesmo objetivo, auxiliando no processamento dos dados por meio do *script* em R.
- *Lubridate* (GROLEMUND; WICKHAM, 2011): O pacote lubridate facilita o trabalho com datas e períodos de tempo, permitindo a manipulação eficiente desses dados. No *script*, ele foi utilizado para esse propósito específico.
- *Lookup* (WRIGHT, 2021): O pacote lookup no R é utilizado para realizar buscas e associações entre diferentes conjuntos de dados. Ele facilita a correspondência entre variáveis, permitindo a busca de valores específicos em uma tabela e a substituição ou associação desses valores a outro conjunto de dados. Embora não seja essencial para a construção do script, já que funções nativas do R poderiam ser utilizadas para obter o mesmo resultado, optou-se por sua utilização devido à facilidade de leitura do código, uma vez que sua sintaxe é semelhante a funções PROCs amplamente conhecidas no Excel.

3.3.2 Análise crítica da base de dados

O processo de crítica de base de dados, conforme detalhado no tópico 2 deste trabalho, inicia-se com a solicitação da base de dados pelo atuário à entidade. Um atuário que mantém uma relação profissional adequada com a entidade deve solicitar a base de dados em um *layout* apropriado, visando otimizar seu trabalho subsequente.

Este procedimento envolve a especificação de um formato de arquivo preferencial, que pode ser .xlsx, .txt, .json, entre outros. A escolha do formato deve considerar tanto a facilidade de manipulação pelo atuário quanto a capacidade de exportação da entidade. No exemplo em questão foi considerado o envio dos dados em formato .xlsx através de uma pasta de trabalho do *software* Excel.

Além de estabelecer um formato específico para a entrada de dados, esta abordagem pode antecipar o processo de limpeza de dados. O atuário, ao definir o *layout*, pode especificar os tipos de variáveis para cada campo, diferenciando entre variáveis de texto, numéricas e datas. Esta prática não apenas facilita a organização dos dados, mas também minimiza erros de entrada e incompatibilidades de formato.

O Quadro 1 exemplifica um possível *layout* para a solicitação de base de dados, mais especificamente da massa de aposentados, utilizando as variáveis empregadas neste trabalho.

Quadro 1 - Estrutura do pedido de base de dados (aposentados)

Campo	Descrição	Tipo	Formato
Matrícula	Identificação do assistido.	Texto	-
Data de Nascimento	Data de nascimento do assistido.	Data	dd/mm/aaaa
Data de Início do Benefício	Data de início do pagamento de benefício ao assistido.	Data	dd/mm/aaaa
Sexo	Sexo do assistido.	Texto	M ou F
Tipo do benefício	Tipo de benefício recebido pelo assistido.	Texto	-
Benefício Complementar	Valor do benefício complementar pago no mês ao assistido.	Numérico	-
Contribuição	Valor da contribuição paga no mês pelo assistido, quando aplicável.	Numérico	-
Data de Nascimento Dep. 1	Data de nascimento do primeiro dependente.	Data	dd/mm/aaaa
Sexo Dep. 1	Sexo do primeiro dependente.	Texto	M ou F
Data de Nascimento Dep. 2	Data de nascimento do segundo dependente.	Data	dd/mm/aaaa
Sexo Dep. 2	Sexo do segundo dependente.	Texto	M ou F
Data de Nascimento Dep. 3	Data de nascimento do terceiro dependente.	Data	dd/mm/aaaa
Sexo Dep. 3	Sexo do terceiro dependente.	Texto	M ou F

Fonte: Elaborado pelo autor (2025).

Este *layout* serve como um guia tanto para a entidade que fornece os dados quanto para o atuário que os recebe, assegurando uma transferência de informações eficiente e precisa.

Considerando o Quadro 1 e as discussões anteriores, este trabalho não focou a limpeza de dados, especificamente no que se refere ao tratamento de variáveis e à remoção de valores ausentes. Isso ocorre porque, dependendo do tipo de dado ausente ou incorreto, o processamento do script poderia ser comprometido, exigindo uma revisão pela entidade responsável e impactando o fluxo de execução. Contudo, apesar desses obstáculos, a estrutura da aplicação foi desenvolvida para que esses erros sejam identificados eventualmente. Por exemplo, a ausência de valores na variável de matrículas ou a presença de matrículas em um formato desconhecido são interpretadas pelo programa como saídas e entradas, respectivamente, sendo analisadas no processo de movimentação da base de dados, conforme detalhado mais adiante.

Além disso, inconsistências em outras variáveis, como a presença de um formato textual na variável Benefício Complementar, comprometeriam a execução do programa e já seriam sinalizadas no momento da importação dos arquivos. Diante desses fatores, a limpeza de dados em sua forma mais rigorosa não foi abordada neste estudo, pois não se alinha diretamente ao seu propósito, ficando como uma possibilidade para pesquisas futuras.

Utilizando o R, o presente trabalho importou, essencialmente, a base de dados utilizando a função ‘*read_excel’* do pacote ‘*readxl’*, conforme a Figura 3.

Figura 3 - Leitura da base de dados

```
# Importação base de dados

# Leitura Base 1 (Anterior)

  observeEvent(input$base1, {
    req(input$base1)
    tryCatch({ # Tratamento de erro
      rv$base1_data <- read_excel(input$base1$datapath, col_types = c("numeric", "date",
"text", "date", "text", "numeric", "numeric", "date", "text", "date", "text", "date",
"text"))
      showNotification("Base 1 carregada com sucesso!", type = "message")
    }, error = function(e) {
      showNotification(paste("Erro ao carregar Base 1:", e$message), type = "error")
    })
  })

# Leitura Base 2 (Atual)
  observeEvent(input$base2, {
    req(input$base2)
    tryCatch({ # Tratamento de erro
      rv$base2_data <- read_excel(input$base2$datapath, col_types = c("numeric", "date",
"text", "date", "text", "numeric", "numeric", "date", "text", "date", "text", "date",
"text"))
      showNotification("Base 2 carregada com sucesso!", type = "message")
    }, error = function(e) {
      showNotification(paste("Erro ao carregar Base 2:", e$message), type = "error")
    })
  })
```

Fonte: Elaborado pelo autor (2025).

É importante ressaltar que, embora essa função tenha a capacidade de identificar automaticamente o tipo de variável de cada campo, é uma boa prática especificar explicitamente o tipo de variável para cada coluna utilizando o parâmetro *col_types*. Essa precaução é especialmente relevante considerando a disposição das informações dos dependentes. No campo de dependentes, é comum haver vários campos vazios. Se o tipo de variável não for definido para esses campos, a função read_excel converterá essas colunas em variáveis booleanas[2]. Por exemplo, se houver campos vazios na variável "Sexo Dependente 1", a função transformará a coluna em uma variável booleana. Isso pode levar a equívocos, como identificar o sexo de um dependente como uma variável booleana F (false) quando, na verdade, se trata de uma variável do tipo texto “F” (feminino).

[2] Variáveis booleanas são variáveis capazes de conter apenas um de dois valores: verdadeiro ou falso. Como as condições no IF retornam verdadeiro ou falso, são chamadas booleanas. (ROMAN, Norton Trevisan. *MC102 - Algoritmos e Programação de Computadores, Parte X*).

Já o parâmetro *sheets* se refere a qual planilha dentro da pasta de trabalho estamos retirando os dados. No exemplo da base de dados utilizada, os dados referentes à massa de ativos estão localizados na primeira planilha; para os aposentados e pensionistas, as planilhas são a segunda e a terceira, respectivamente.

Da mesma forma, o tratamento será aplicado à importação da base de dados anterior, que deve estar posicionada em um período anterior à data de referência da base atual. Esse processo garante a correta comparação entre as bases e a identificação de variações nos dados ao longo do tempo.

Para verificar se as variações dos benefícios complementares estão em conformidade com o esperado, é necessário importar as variações do indexador atuarial do plano utilizando o pacote rbcb. No código implementado, o indexador é definido pelo usuário e armazenado na variável *index_code*, que contém o número serial correspondente fornecido pelo Banco Central. Esse código é então utilizado para recuperar as variações do indexador dentro do período especificado, considerando a data inicial (*start_date*) e a data final (*end_date*).

Figura 4 - Importação da variação do indexador

```
# Série indexador
get_accumulated_index <- function(index_code, start_date, end_date) {
  tryCatch({ # Tratamento de erro
    serie <- rbcb::get_series(
      index_code, # Número do índice escolhido pelo usuário (IPCA= 433, IGPM= 189, INPC=
188, etc)
      start_date = start_date, # Data de início (representa a data da base anterior)
      end_date = end_date # Data final (representa a data da base atual)
    )
    accumulated <- prod(1 + (serie[2] / 100)) - 1 # Fórmula para acumular o ínidice
    return(list(success = TRUE, value = accumulated, error = NULL))
  }, error = function(e) {
    return(list(success = FALSE, value = NULL, error = e$message))
  })
```

Fonte: Elaborado pelo autor (2025).

Com base nas variações do indexador no período, podemos calcular a variação acumulada utilizando a seguinte formulação, que também está destacada na Figura 4:

$$Var.\ Acumulada = \prod_{i=1}^{n}\left(1 + \frac{Var.mensal_i}{100}\right) - 1 \qquad (1)$$

$Var.\ mensal_i$ é o valor da variação mensal para o mês i; n é o número total de meses na série.

Outro dado importante para construção da crítica da base de dados é o salário mínimo, que é utilizado para identificar benefícios que estão abaixo desse valor.

Figura 5 - Importação do salário mínimo

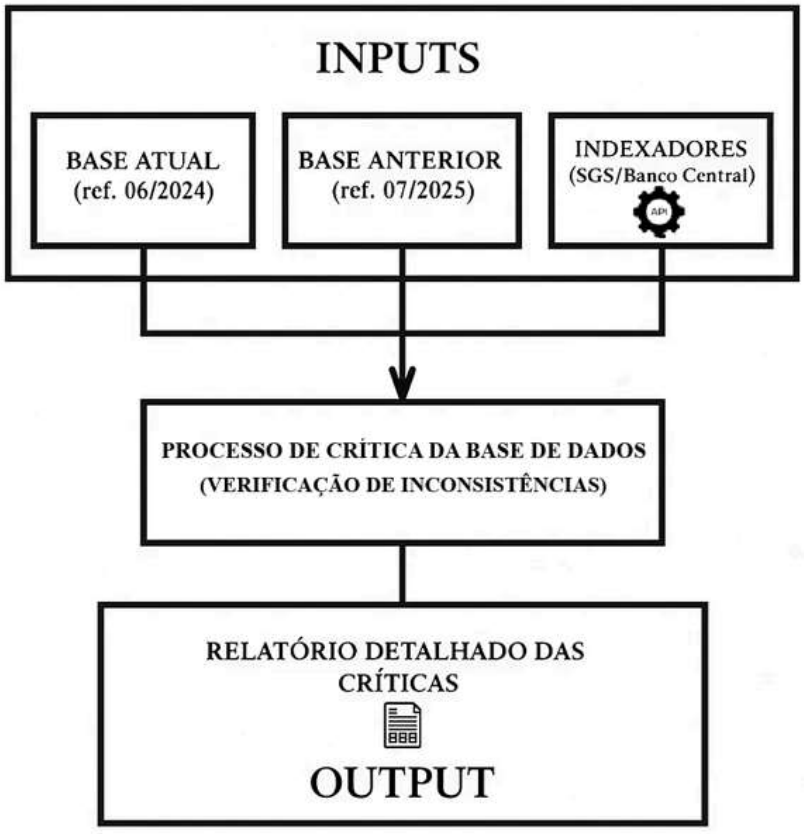
```
# Salário Mínimo

sal_minimo <- rbcb::get_series(1619
                               , start_date = input$data_inicio, end_date = input$data_fim)
sal_minimo <- tail(sal_minimo$`1619`, n = 1)
```

Fonte: Elaborado pelo autor (2025).

A importação do salário mínimo segue um processo semelhante ao da variação do indexador, com a diferença de que o código para sua importação é fixo (1619).

Informados os indexadores, verificam-se as inconsistências das bases de dados e se gera o relatório detalhado das críticas. Todo o processo de verificação das inconsistências, do início ao fim, é apresentado na Figura 6.

Figura 6 - Processo de análise crítica da base de dados



Fonte: Elaborado pelo autor (2025).

Assim, tem-se que o processo de crítica é formado: a) pela fase de *input*, isto é, a fase em que se importam as bases de dados anteriores e a atual, e que se informam os indexadores; b) pela fase de verificação de inconsistências das bases; c) pela fase de *outputs*, se consume, ou seja, a fase em que se geram os relatórios detalhados das críticas.

3.3.3 Análise descritiva

A análise estatística descritiva da massa de aposentados, embora não forneça conclusões definitivas ou específicas, oferece informações valiosas sobre a maturidade do plano e permite uma avaliação inicial da conformidade da base de dados com as expectativas do setor. As estatísticas utilizadas neste estudo foram selecionadas com base no que foi apresentado por Pinheiro (2007), com as devidas modificações. Estas estatísticas, resumidas no Quadro 2, fornecem uma visão panorâmica das características demográficas e financeiras dos participantes. O Quadro 2 fornece uma visão geral dos aposentados, permitindo uma análise preliminar das características da base de dados. Ele não apenas oferece uma visão descritiva dos dados, mas também funciona como uma ferramenta inicial para identificar possíveis inconsistências, auxiliando o administrador responsável pelo plano de benefícios a identificá-las antecipadamente.

Quadro 2 - Estatística descritiva das informações dos participantes do plano

Descrição	Base Anterior	Base Atual
1. Participantes Aposentados	-	-
1.2 Homens	-	-
1.3 Mulheres	-	-
2. Idade Média	-	-
3. Número médio de dependentes	-	-
4. Benefício Complementar Total	-	-
5. Benefício Complementar Médio	-	-

Fonte: Elaborado pelo autor (2025).

A análise das idades médias, por exemplo, permite avaliar o envelhecimento da população de beneficiários, o que é essencial para projeções futuras e o planejamento do fundo. Além disso, o número de dependentes registrado fornece informações importantes sobre os compromissos futuros do plano, já que a inclusão de dependentes pode prolongar a duração dos pagamentos e aumentar os passivos no plano.

Outro aspecto relevante é o benefício complementar total e médio. O benefício complementar total reflete o impacto financeiro dos compromissos, enquanto o benefício médio

oferece uma noção relação a necessidade de cada participante. No R, a metodologia utilizada para preencher cada um desses campos está ilustrada na Figura 7.

Figura 7 – Cálculo das estatísticas

```
# 1. Participantes Aposentados
estatísticas[1, 2] <- nrow(base_anterior)
estatísticas[1, 3] <- nrow(base_atual)
# 1.2. Homens
estatísticas[2, 2] <- sum(ifelse(base_anterior$`Sexo` == "M", 1, 0))
estatísticas[2, 3] <- sum(ifelse(base_atual$`Sexo` == "M", 1, 0))
# 1.3. Mulheres
estatísticas[3, 2] <- sum(ifelse(base_anterior$`Sexo` == "F", 1, 0))
estatísticas[3, 3] <- sum(ifelse(base_atual$`Sexo` == "F", 1, 0))
# 2. Idade Média
data_atual <- as.Date(input$data_fim, origin = "1899-12-30") # mesma base de data
datas_nascimento_anterior <- as.Date(base_anterior$`Data de Nascimento`, format = "%d/%m/%Y") # para o sistema de data do Excel (base 1900)
datas_nascimento_atual <- as.Date(base_atual$`Data de Nascimento`, format = "%d/%m/%Y") # para o sistema de data do Excel (base 1900)
estatísticas[4, 2] <- mean(as.numeric(difftime(data_atual, datas_nascimento_anterior, units = "weeks")) %/% 52)
estatísticas[4, 3] <- mean(as.numeric(difftime(data_atual, datas_nascimento_atual, units = "weeks")) %/% 52)
# 3. Número médio de dependentes
estatísticas[5, 2] <- sum(rowSums(!is.na(base_anterior[, c("Sexo Dep. 1", "Sexo Dep. 2", "Sexo Dep. 3")])))
estatísticas[5, 3] <- sum(rowSums(!is.na(base_atual[, c("Sexo Dep. 1", "Sexo Dep. 2", "Sexo Dep. 3")])))
# 4. Benefício Complementar Total
estatísticas[6, 2] <- sum(base_anterior$`Benefício Complementar`)
estatísticas[6, 3] <- sum(base_atual$`Benefício Complementar`)
# 5. Benefício Complementar Médio
estatísticas[7, 2] <- mean(base_anterior$`Benefício Complementar`)
estatísticas[7, 3] <- mean(base_atual$`Benefício Complementar`)
```

Fonte: Elaborado pelo autor (2025).

É importante destacar que, conforme ilustrado na Figura 7 e ao longo de todo o script, a técnica de vetorização[3] foi adotada sempre que possível. Funções como *nrow()*, *sum()* e *mean()*, dentre outras, foram utilizadas para otimizar o processamento dos dados, evitando o uso de laços ou estruturas de repetição e, consequentemente, melhorando a eficiência do código.

[3] A vetorização reduz o tempo de computação em várias ordens de magnitude, e a diferença em relação aos laços aumenta com o tamanho dos dados. Portanto, se for necessário lidar com grandes volumes de dados, reescrever o algoritmo como operações em matrizes pode levar a ganhos significativos de desempenho (R-bloggers, 2018).

3.3.4 Movimentação cadastral

A movimentação também é uma ferramenta importante para identificar inconsistências na base de dados. Ao comparar duas bases de dados, em datas diferentes, é possível identificar mudanças de status, neste caso, os participantes que se aposentaram e as matrículas que saíram da situação de aposentadoria. Tais movimentações são importantes, pois impactam as provisões a serem estimadas, conforme a base de participantes, e por esse motivo se faz necessária a sua confirmação por parte da entidade. Aqui, utilizou-se do *layout* da Tabela 1 para resumir essas movimentações.

Tabela 1 - Movimentação da base de dados

Entradas				**Saídas**			
Matrícula	**Sexo**	**Data de Nascimento**	**Benefício Complementar**	**Matrícula**	**Sexo**	**Data de Nascimento**	**Benefício complementar**
-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-

Fonte: Elaborado pelo autor (2025).

Como ilustrado na Tabela 1, as movimentações da massa de participantes estão organizadas em duas seções da tabela: a) uma para as entradas, que contém informações sobre os participantes que ingressaram no plano como aposentados durante o período analisado; e outra para as saídas, que apresenta aqueles que deixaram essa condição. Cada movimentação é detalhada por meio de informações básicas – como matrícula, sexo, data de nascimento e benefício complementar – que facilitam o trabalho de validação pela entidade em uma primeira instância. A Figura 8 descreve a análise da movimentação da base cadastral no script.

Figura 8 – Movimentação da base de dados

```
# Movimentação

entradas <- dplyr::anti_join(base_atual, base_anterior, by = "Matrícula") # Quem está na base atual e não está na base anterior
entradas <- entradas %>% select("Matrícula", "Sexo", "Data de Nascimento", "Benefício Complementar")

saídas <- dplyr::anti_join(base_anterior, base_atual, by = "Matrícula") #  Quem está na base anterior e não está na base atual
saídas <- saídas %>% select("Matrícula", "Sexo", "Data de Nascimento", "Benefício Complementar")
```

Fonte: Elaborado pelo autor (2025).

Para criação da tabela foi utilizado a função *anti_join* da biblioteca *dplyr*, permitindo identificar tanto os participantes que estão na base atual, mas não na base anterior, quanto aqueles que estavam na base anterior, mas não na base atual. A função *anti_join* foi escolhida por simplificar a filtragem de registros ausentes em relação a uma base de dados, eliminando a necessidade de operações manuais complexas com funções base do R, sem comprometer o desempenho.

3.3.5 Inconsistências

A verificação de inconsistências na base de dados é um passo essencial para garantir a qualidade e confiabilidade das informações utilizadas na análise atuarial. Além das abordagens exploradas em tópicos anteriores, que também desempenham esse papel, é possível, a partir da leitura do regulamento do plano específico, formular premissas básicas ou regras que auxiliem na identificação de novas falhas.

Como o objetivo deste trabalho é desenvolver um modelo que generalize melhor esses conceitos, foi utilizado um conjunto de regras que melhor atendesse a esse propósito. Tais inconsistências na base de dados podem resultar em impactos financeiros significativos para o fundo de pensão, tornando essencial a sua detecção e correção. As principais inconsistências consideradas neste estudo estão resumidas no Quadro 3:

Quadro 3 - Inconsistências consideradas para a base de dados

Aposentados	Descrição
Inconsistência nº 1	Matrículas Repetidas
Inconsistência nº 2	Idade maior que 90 anos
Inconsistência nº 3	Benefício menor que o salário mínimo
Inconsistência nº 4	Data de Início de Benefício menor que a Data de Nascimento
Inconsistência nº 5	Data de Nascimento divergente
Inconsistência nº 6	Sexo divergente
Inconsistência nº 7	Tipo de Benefício divergente
Inconsistência nº 8	Benefício Complementar com variações significativas

Fonte: Elaborado pelo autor (2025).

As inconsistências listadas representam potenciais erros nos registros, os quais demandam verificação e correção, conforme descrito a seguir:

- **Matrículas Repetidas**: Indicam duplicidade de cadastro, o que pode gerar inconsistências na contabilização dos benefícios.

- **Idade maior que 90 anos**: Embora possa existir beneficiários nessa faixa etária, casos extremos podem sinalizar erros de digitação ou registros desatualizados.
- **Benefício menor que o salário mínimo**: Pode sugerir falhas no cálculo do benefício ou erros de digitação, a depender do tipo de benefício e regulamento do plano.
- **Data de Início de Benefício menor que a Data de Nascimento**: Um erro lógico que inviabiliza a consistência dos dados cadastrais.
- **Sexo divergente**: Pode indicar troca de informações entre beneficiários distintos ou erros na base de dados anterior ou atual.
- **Data de Nascimento divergente**: Pode indicar troca de informações entre beneficiários distintos ou erros na base de dados anterior ou atual.
- **Tipo de Benefício divergente**: Pode indicar erro de cadastro ou alteração indevida.
- **Benefício Complementar com variações significativas**: Alterações significativas nos valores do benefício mesmo considerando mês de reajuste podem sugerir algum tipo de erro na base de dados.

A metodologia utilizada para detectar essas inconsistências baseou-se na automação da análise dos dados por meio do software R, conforme a Figura 9. Foram aplicados testes lógicos para identificar registros que não atendem aos critérios esperados, permitindo uma verificação eficiente e objetiva. Assim, o uso da linguagem R possibilita a rápida identificação dessas inconsistências, otimizando o processo de análise.

O código da Figura 9 estrutura as inconsistências na base de dados por meio de uma matriz chamada “críticas”, onde cada linha corresponde a um tipo específico de inconsistência e a respectiva quantidade de ocorrências identificadas. Essa matriz facilita a organização e interpretação dos resultados.

Para identificar divergências entre bases de diferentes períodos, o código utiliza a função *vlookup*, que busca informações correspondentes em outra tabela com base na matrícula dos participantes. Isso possibilita a comparação de dados como data de nascimento, sexo, tipo de benefício e valores financeiros, permitindo a detecção de alterações inesperadas ou erros de cadastro.

Figura 9 - Obtenção das críticas utilizando testes lógicos

```
# 1. Matrículas Repetidas
criticas[1, 2] <- sum(ifelse(duplicated(rv$base2_data$Matrícula) | duplicated(rv$base2_data$Matrícula, fromLast = TRUE), 1, 0)) # Ocorrências

# 2. Idade maior que 90 anos
criticas[2, 2] <- sum(ifelse(idades > 90, 1, 0)) # Ocorrências

# 3. Benefício menor que o salário mínimo
criticas[3, 2] <- sum(ifelse(base_atual$`Benefício Complementar` < sal_minimo, 1, 0))

# 4. Data de Início de Benefício menor que a Data de Nascimento
criticas[4, 2] <- sum(ifelse(base_atual$`Data de Início de Benefício` < base_atual$`Data de Nascimento`, 1, 0)) # Ocorrências

# 5. Data de Nascimento divergente
atual <- vlookup(base_atual_x_anterior$`Matrícula`, base_atual, "Matrícula", "Data de Nascimento")

anterior <- vlookup(base_atual_x_anterior$`Matrícula`, base_anterior, "Matrícula", "Data de Nascimento")

divergencias_atual_anterior$`Data de Nascimento divergente` <- ifelse(atual != anterior, 1, 0)

criticas[5, 2] <- sum(divergencias_atual_anterior$`Data de Nascimento divergente`)

# 6. Sexo divergente
atual <- vlookup(base_atual_x_anterior$`Matrícula`, base_atual, "Matrícula", "Sexo")

anterior <- vlookup(base_atual_x_anterior$`Matrícula`, base_anterior, "Matrícula", "Sexo")

divergencias_atual_anterior$`Sexo divergente` <- ifelse(atual != anterior, 1, 0)

criticas[6, 2] <- sum(ifelse(atual != anterior, 1, 0)) # Ocorrências

# 7. Tipo de Benefício divergente
atual <- vlookup(base_atual_x_anterior$`Matrícula`, base_atual, "Matrícula", "Tipo do Benefício")

anterior <- vlookup(base_atual_x_anterior$`Matrícula`, base_anterior, "Matrícula", "Tipo do Benefício")

divergencias_atual_anterior$`Tipo de Benefício divergente` <- ifelse(atual != anterior, 1, 0)

criticas[7, 2] <- sum(ifelse(atual != anterior, 1, 0)) # Ocorrências

# 8. Benefício complementar com divergências significativas
atual <- vlookup(base_atual_x_anterior$`Matrícula`, base_atual, "Matrícula", "Benefício Complementar")

anterior <- vlookup(base_atual_x_anterior$`Matrícula`, base_anterior, "Matrícula", "Benefício Complementar")

acumulado = rv$current_params$acumulado

limite_superior <- acumulado / 100 + 1 / 100  # acumulado + 1%

limite_inferior <- acumulado / 100 - 1 / 100  # acumulado - 1%

variacao <- (atual - anterior) / anterior

divergencias_atual_anterior$`Benefício Complementar com variações significativas` <- ifelse(abs(variacao) > limite_superior, 1, 0)

criticas[8, 2] <- sum(divergencias_atual_anterior$`Benefício Complementar com variações significativas`)
```

Fonte: Elaborado pelo autor (2025).

Além disso, foram aplicados testes lógicos para verificar critérios específicos, como idades superiores a 90 anos, benefícios abaixo do salário mínimo, entre outros. No caso das variações no benefício complementar, foi estabelecido um limite percentual (margem),

juntamente com o valor do indexador acumulado para identificar mudanças significativas que possam indicar falhas.

3.3.4 Exportação do relatório para arquivo .xlsx

A exportação do relatório para o formato .xlsx é uma etapa essencial dentro da metodologia, pois torna os resultados gerados no R mais acessíveis e compreensíveis para aqueles que não têm familiaridade com programação. Esse formato permite uma comparação direta com os resultados obtidos no Excel, utilizando fórmulas convencionais, o que facilita a validação dos resultados gerados pelo script em R. Com a utilização do pacote *openxlsx*, o próprio script gera um relatório no Excel, permitindo que todos os dados, tabelas e estatísticas sejam consolidados em um único arquivo. Isso facilita a visualização e interpretação das informações, especialmente para profissionais, que não têm a expectativa de rodar scripts para acessar os resultados. O formato .xlsx torna o processo mais acessível e compreensível para os profissionais, sem comprometer a integridade das informações obtidas no R. Em outras palavras, é uma forma de passar tudo que foi obtido com o R de forma prática e mais palpável para aquele que irá consumir tais tipos de informação. As abas que foram construídas na pasta de trabalho utilizando o script estão dispostas no Quadro 4.

Para essa etapa do trabalho houve o uso extensivo do pacote '*openxlsx*' no R, possibilitando a edição das planilhas com um formato mais familiar para os resultados obtidos. A seguir, detalha-se a função de cada aba.

Quadro 4 - Abas da pasta de trabalho do arquivo .xlsx

Aba	Descrição
Parâmetros	Informações preliminares inseridas pelo usuário, que serviram de apoio para construção das críticas.
Base 1	Base anterior
Base 2	Base atual
Movimentação	Movimentação da base de dados
Divergências	Resumo das divergências encontradas
Anexo	Detalhamento das divergências

Fonte: Elaborado pelo autor (2025).

A aba "Parâmetros" contém as informações iniciais inseridas pelo usuário, servindo como referência para a construção das críticas. Nela, são apresentados o indexador utilizado, a data de início e término da análise, o valor acumulado do índice e, complementarmente, os nomes e caminhos dos arquivos importados.

A aba Base 1 representa a base de dados anterior, sendo a mesma importada no processo. Embora sua exportação para o arquivo .xlsx não seja essencial, essa aba serve como um meio de conferência, permitindo ao usuário verificar a disposição de uma matrícula específica por meio de filtros, caso necessário.

A aba Base 2 representa a base de dados atual, correspondendo à versão mais recente importada para a análise. Embora sua exportação para o arquivo .xlsx não seja essencial, ela pode servir como um recurso adicional de conferência, permitindo ao usuário verificar os dados cadastrais de uma matrícula específica e comparar diretamente com a base anterior.

A aba Movimentação apresenta as alterações identificadas entre as duas bases de dados, permitindo a visualização de entradas, saídas. Com essas informações, torna-se possível avaliar, de forma preliminar, se há uma tendência de redução ou aumento dos compromissos do plano a depender da quantidade de entradas e saídas observadas. Além disso, essa aba inclui os resultados das estatísticas descritivas mencionadas anteriormente pelo Quadro 2, fornecendo um panorama mais detalhado das variações observadas.

A aba Divergências apresenta um resumo das inconsistências identificadas, conforme detalhado no Quadro 3 anteriormente. Ela oferece uma visão geral de todas as inconsistências apontadas, bem como a frequência de ocorrência em relação à base de dados. Dessa forma, serve como um indicador relevante para avaliar o nível de conformidade dos dados.

Por fim, a aba Anexo fornece um detalhamento mais aprofundado das matrículas em que foram identificadas divergências. Para cada inconsistência encontrada, as matrículas correspondentes são listadas junto às variáveis que influenciaram sua classificação como inconsistentes. Por exemplo, no caso da inconsistência relacionada a variações significativas no benefício complementar, a aba apresentará as matrículas envolvidas, acompanhadas dos valores do benefício complementar anterior e atual, permitindo uma análise mais precisa das mudanças ocorridas.

4 RESULTADOS

Este tópico tem como objetivo apresentar os resultados obtidos a partir da análise da base de dados, seguindo os passos descritos na metodologia. Como mencionado anteriormente, a base de dados utilizada nesta pesquisa não é real, mas sim simulada, permitindo testar o script e verificar se cada inconsistência está sendo corretamente detectada. Para isso, a base de dados foi propositalmente modificada, simulando erros que poderiam ocorrer em um cenário real.

Vale destacar que a base de dados atual contém 20.738 aposentados, enquanto a base anterior contém 20.742 aposentados. Já o período de análise considera julho de 2023 como data de referência para a base anterior e junho de 2024 para a base atual. Por fim, o indexador utilizado para o estudo foi o IPCA (Índice Nacional de Preços ao Consumidor Amplo), cuja variação nesse período foi de 4,23%. Esse valor foi obtido de forma automatizada por meio da API do Banco Central, utilizando a formulação do índice acumulado descrita na Equação 1.

4.1 ESTATÍSTICAS

O primeiro ponto a ser observado é em relação à estatística descritiva da base de dados. Após importar a base de dados e gerar o relatório, foram obtidos os seguintes resultados, descritos pelo Quadro 5.

Quadro 5 - Estatística descritiva das informações dos aposentados do plano (Resultados)

Descrição	Base Anterior	Base Atual
1. Participantes Aposentados	20742	20738
1.2 Homens	10392	10393
1.3 Mulheres	10350	10345
2. Idade Média	74,29	75,22
3. Número médio de dependentes	1,99	1,57
4. Benefício Complementar Total	R$ 177.581.564,00	R$ 185.060.130,50
5. Benefício Complementar Médio	R$ 8.561,49	R$ 8.923,72

Fonte: Elaborado pelo autor (2025).

Foi constatado que na base anterior havia um total de 20.742 aposentados, enquanto na base atual esse número se reduziu para 20.738. Ao analisar a demografia por sexo, observa-se um equilíbrio entre participantes homens e mulheres. A idade média dos participantes na base atual é aproximadamente um ano superior à da base anterior, refletindo o impacto natural da passagem do tempo no período analisado, de julho de 2023 a junho de 2024.

Em relação ao número médio de dependentes, observou-se uma redução significativa na base atual. Essa diminuição abrupta deve ser questionada junto à entidade, pois não é comum que ocorra sem justificativa plausível. Possíveis causas incluem falhas na atualização dos registros, erros na migração de dados ou mudanças nos critérios de elegibilidade dos dependentes. Tal inconsistência pode comprometer projeções atuariais, especialmente no cálculo de benefícios continuados e provisões matemáticas, sendo essencial investigar suas origens para garantir a confiabilidade da base.

Por fim, no que se refere ao benefício complementar, o montante total acumulado na base anterior foi de R$ 177.581.564,00, enquanto na base atual esse valor aumentou para R$ 185.060.130,50. Esse crescimento expressivo se deve, em grande parte, à correção monetária do período, que, conforme mencionado anteriormente, reflete a variação do IPCA, aproximadamente 4,22% entre julho de 2023 e junho de 2024.

4.2 MOVIMENTAÇÃO

Em relação à movimentação da base de dados, conforme observado na subseção 4.1, a base anterior registrava 20.742 participantes, enquanto a base atual apresenta 20.738. No entanto, essa diferença (de quatro participantes) não significa necessariamente que houve exatamente 4 (quatro) saídas, uma vez que o total de participantes resulta tanto das exclusões quanto das novas adesões. Para uma análise mais precisa, a tabela abaixo apresenta um resumo detalhado da movimentação ocorrida no período, que justifica os números encontrados.

Tabela 2 - Movimentação da base de dados (Resultados)

Entradas (5)				**Saídas (11)**			
Matrícula	Sexo	Data de Nascimento	Benefício Complementar	Matrícula	Sexo	Data de Nascimento	Benefício Complementar
99	F	24/05/1967	3444,23	60	F	13/11/1957	14157
7181	F	21/06/1968	1989,22	88	F	01/08/1949	14260
20747	M	05/03/1970	4213,21	126	F	23/11/1954	2540
20748	M	17/10/1978	8600	154	F	01/08/1941	3488
20749	M	17/05/1971	2450,22	178	M	19/03/1942	6680
				238	M	19/01/1956	15556
				312	M	14/05/1939	1635
				4961	F	13/11/1952	3522
				16984	F	15/01/1946	4796
				16985	F	11/08/1943	15117
				18663	M	25/10/1942	4810

Fonte: Elaborado pelo autor (2025).

No total, foram verificadas 5 entradas de novos participantes aposentados e, simultaneamente, a saída de 11 aposentados. Esse é um indicador relevante, pois auxilia na projeção das variações das provisões ao longo do tempo. Além disso, cabe à entidade confirmar se essas movimentações ocorreram conforme esperado.

A movimentação observada é coerente com a dinâmica real observada em planos previdenciários maduros, onde é comum que o número de saídas supere o número de entradas na base de aposentados. Esse comportamento está alinhado ao ciclo de vida dos beneficiários: à medida que os participantes falecem, eles deixam de compor a base de aposentados e, caso tenham dependentes elegíveis, originam registros na base de pensionistas. Assim, a maior quantidade de saídas observada na simulação reflete a natural transição de um plano com maturidade avançada, em que o envelhecimento da população tende a aumentar a migração para a base de pensionistas.

4.3 INCONSISTÊNCIAS

Durante a verificação dos dados, foram identificadas 44 inconsistências, que podem indicar possíveis erros de cadastro ou necessidade de revisão de determinadas informações. Essas inconsistências foram categorizadas e estão apresentadas no Quadro 6, permitindo uma visão geral da frequência e do percentual de registros afetados em relação ao total da base.

Quadro 6 - Inconsistências consideradas para a base de dados (Resultados)

Críticas	Frequência	Percentual Sobre a Base
Matrículas Repetidas	4	0,02%
Idade maior que 90 anos	5	0,02%
Benefício menor que o salário mínimo	2	0,01%
Data de Início de Benefício menor que a Data de Nascimento	1	0%
Data de Nascimento divergente	9	0,04%
Sexo divergente	7	0,03%
Tipo de Benefício divergente	9	0,04%
Benefício Complementar com variações significativas	7	0,03%

Fonte: Elaborado pelo autor (2025).

Embora as inconsistências identificadas representem um percentual reduzido da base de dados, é fundamental que sejam analisadas e devidamente verificadas, considerando seu potencial impacto sobre a qualidade das informações. Ao comparar com avaliações atuariais

publicamente disponíveis — ainda que pertencentes a Regimes Próprios de Previdência Social (RPPS) — é possível constatar que falhas semelhantes são relativamente comuns, sendo tratadas com soluções práticas e recorrentes. Por exemplo, inconsistências relacionadas a matrículas repetidas, que representaram 0,02% da base analisada neste estudo, também foram observadas na avaliação atuarial de 2017 do PBPREV, cuja base contava com 38.791 aposentados, com um valor próximo (0,01%), sendo tratadas com a exclusão dos registros excedentes. Em outros casos, como nos relatórios de 2025 do RioPretoPrev e de 2024 do Município de Lages/SC, optou-se pela adoção de matrículas hipotéticas. Já registros com valores de benefício inferiores ao salário mínimo, que corresponderam a 0,01% na base deste trabalho, tiveram maior incidência no PBPREV (1,39%), sendo corrigidos com a imputação do valor do Salário Mínimo Nacional, solução também adotada em outras entidades, como o Município de Lages/SC.

Estabelecendo um paralelo com as soluções propostas pela Society of Actuaries (SOA) — brevemente discutidas no referencial teórico — observa-se que as medidas adotadas por esses RPPS se alinham com algumas das abordagens recomendadas. O tratamento das matrículas repetidas pode ser compreendido como uma combinação entre as técnicas de *deleting* (exclusão dos registros inválidos) e *filling in* (preenchimento com dados, como a matrícula hipotética). Já a imputação de benefícios abaixo do salário mínimo pode ser relacionada às estratégias de *filling in*, por envolver a substituição por um valor mínimo estabelecido, e *correcting*, quando se considera que a correção foi baseada em uma norma legal vigente.

Importante ressaltar que, no âmbito deste trabalho, nenhuma alteração foi realizada nos dados analisados. As soluções aqui mencionadas são utilizadas apenas como referência, com o intuito de ilustrar como inconsistências semelhantes são tratadas em avaliações atuariais reais e de relacionar essas práticas com recomendações técnicas reconhecidas internacionalmente. O foco deste estudo está centrado exclusivamente na identificação e análise crítica da base de dados, não abrangendo a avaliação atuarial como um todo. A seção a seguir apresenta uma análise detalhada das inconsistências encontradas, destacando seus possíveis impactos e a necessidade de atenção específica para cada caso.

4.4 ANEXO

Este tópico apresenta as informações geradas pelo R no relatório detalhado da base de dados, disponíveis na aba “Anexo”, onde é possível visualizar quais matrículas apresentaram algum tipo de inconsistência.

No que se refere às matrículas repetidas, as ocorrências identificadas estão listadas na Tabela 3.

Tabela 3 - Matrículas repetidas (resultados)

Código 1 **Matrículas Repetidas**	
Matrícula	**Data de Nascimento**
19	25/11/1946
19	25/11/1946
311	19/10/1958
311	19/10/1958

Fonte: Elaborado pelo Autor (2025).

A duplicidade de matrículas pode impactar diretamente a provisão matemática de benefícios concedidos aos participantes, pois pode levar à contabilização indevida de valores. Além disso, essa inconsistência gera uma série de problemas na base, pois impacta o uso de funções de correspondência, como o PROCV no Excel, as funções nativas do R e o pacote *vlookup*, utilizado neste trabalho. Ou seja, a presença de matrículas repetidas pode resultar em múltiplas correspondências, levando ao retorno de informações incorretas para determinadas matrículas.

Em relação às inconsistências na idade dos participantes aposentados, a Tabela 4 apresenta as matrículas com registros incoerentes.

Tabela 4 - Idade maior que 90 anos (resultados)

Código 2 **Idade maior que 90 anos**		
Matrícula	**Data de Nascimento**	**Idade**
13	10/11/1913	110
654	12/11/1900	123
667	05/11/1913	110
1004	22/03/1918	106
1040	15/08/1922	102

Fonte: Elaborado pelo Autor (2025).

Essa crítica reflete a necessidade de atualizar a base cadastral continuamente, com o objetivo de confirmar junto à entidade se não há registros que não deveriam mais constar como

aposentados, mas que permanecem na base por algum motivo. A idade de 90 anos, em específico, foi utilizada nessa crítica de forma arbitrária, apenas para representar uma idade elevada que requereria verificação por parte da entidade.

No que tange às inconsistências relacionadas ao valor do benefício, a crítica de "Benefício menor que o salário mínimo" identificou 2 matrículas, dispostas na Tabela 5.

Tabela 5 - Benefício menor que o salário mínimo

Código 3 **Benefício menor que o salário mínimo**		
Matrícula	**Benefício Complementar**	**Salário mínimo**
9	1332,03	1412
1117	901,11	1412

Fonte: Elaborado pelo autor (2025).

Dependendo do tipo de benefício, essa situação pode indicar um erro no registro do valor do benefício na base de dados. Como o valor identificado está abaixo do esperado, é fundamental que a entidade responsável revise esses registros com atenção, a fim de corrigir eventuais falhas e garantir que os participantes recebam o valor adequado conforme esteja previsto no regulamento do plano.

Em relação à inconsistência de data de início de benefício ser menor que a data de nascimento do participante aposentado, foi identificada 1 matrícula com essa falha, conforme indicado na Tabela 6.

Tabela 6 - Data de início de benefício menor que a data de nascimento (resultados)

Código 4 **Data de Início de Benefício menor que a Data de Nascimento**		
Matrícula	**Data de Início de Benefício**	**Data de Nascimento**
99	17/11/1965	24/05/1967

Fonte: Elaborado pelo autor (2025).

Essa crítica evidencia um erro de registro, pois é impossível que a data de início de um benefício seja anterior à data de nascimento do participante.

Também no que se refere à data de nascimento do participante, é possível comparar os dados cadastrais do aposentado entre a base atual e a base anterior. Nesse processo, foram identificadas 9 matrículas com divergências na data de nascimento em relação à base anterior, conforme indicado na Tabela 7.

Tabela 7 - Data de nascimento divergente (resultados)

Código 5 Data de Nascimento divergente		
Matrícula	**Data de Nascimento Anterior**	**Data de Nascimento Atual**
1	11/10/1957	10/10/1957
13	16/12/1939	10/11/1913
654	12/11/1950	12/11/1900
667	05/11/1944	05/11/1913
729	27/10/1953	11/12/1956
779	22/03/1939	05/08/1955
820	22/02/1955	02/02/1939
1004	22/03/1938	22/03/1918
1040	15/08/1939	15/08/1922

Fonte: Elaborado pelo autor (2025).

Essa inconsistência pode ter origem em um erro na base anterior ou na base atual. No entanto, considerando que a base anterior já passou por validação, é mais provável que a inconsistência esteja na base atual. Cabe à entidade responsável realizar a devida verificação no cadastro do aposentado para garantir a precisão desse registro.

Conforme indicado na Tabela 8, foram identificadas 7 matrículas com divergência no campo "sexo" em relação à base anterior.

Tabela 8 - Sexo divergente (resultados)

Código 6 Sexo divergente		
Matrícula	**Sexo Anterior**	**Sexo Atual**
9	F	M
68	M	F
124	M	F
144	F	M
746	M	F
794	M	F
804	F	M

Fonte: Elaborado pelo Autor (2025).

Assim como na Tabela 7, esse tipo de inconsistência cadastral deve ser comunicado à entidade responsável, para que seja investigada a causa da divergência e, se necessário, realizada a devida correção.

Relativo às inconsistências no "Tipo de Benefício Divergente", foram identificadas 9 matrículas, conforme indicado na Tabela 9.

Tabela 9 - Tipo de benefício divergente (resultados)

Código 7 Tipo de Benefício divergente		
Matrícula	**Tipo de Benefício Anterior**	**Tipo de Benefício Atual**
29	TIPO 3	TIPO 2
90	TIPO 3	TIPO 1
105	TIPO 1	TIPO 3
112	TIPO 2	TIPO 3
176	TIPO 1	TIPO 2
177	TIPO 3	TIPO 2
181	TIPO 1	TIPO 2
182	TIPO 1	TIPO 2
200	TIPO 3	TIPO 2

Fonte: Elaborado pelo autor (2025).

Uma alteração no tipo de benefício pode impactar diretamente os valores ou as condições do pagamento ao aposentado. Por esse motivo, é essencial que a entidade realize a devida verificação para assegurar a precisão das informações cadastrais.

Por fim, conforme apresentado na Tabela 10, foram identificadas 7 matrículas cuja variação de benefício não ocorreu conforme o esperado.

Tabela 10 - Benefício complementar com variações significativas (resultados)

Código 8 Benefício Complementar com variações significativas				
Matrícula	**Benefício Anterior**	**Benefício Atual**	**Variação**	**Variação Esperada**
9	14634	1332,03	-0,91	0,04
10	6457	5729,98	-0,11	0,04
15	9913	20332,08	1,05	0,04
278	3999	41680,06	9,42	0,04
776	6834	8122,91	0,19	0,04
1012	2536	1643,21	-0,35	0,04
1117	1824	901,11	-0,51	0,04

Fonte: Elaborado pelo autor (2025).

A crítica de variação de benefício é uma das mais relevantes, pois pode revelar diversos erros operacionais, como a aplicação incorreta de indexadores, falhas no período de atualização e até erros de digitação no registro do benefício. Alterações inesperadas no valor do benefício podem ter um impacto significativo nas obrigações do plano, exigindo uma análise cuidadosa.

No caso específico da matrícula 9, o benefício anterior era de R$ 14.634,00, mas atualmente consta como R$ 1.332,03. Essa mesma matrícula também aparece na Tabela 5, pois seu benefício atual está abaixo do salário mínimo. A variação registrada foi de aproximadamente -90,90%, representando uma perda expressiva para o aposentado.

Por outro lado, a matrícula 278 apresentou uma variação de aproximadamente 942,26%, um aumento significativo que deve ser analisado com atenção. Diante de oscilações tão

elevadas, é essencial que tais matrículas sejam questionadas para entidade avaliar a origem dessas variações.

4.5 DESEMPENHO

Em relação ao desempenho da interface web e do script, ambos demonstraram ser eficazes. O tempo de processamento foi satisfatório, sem apresentar problemas técnicos ou computacionais significativos. A importação da base de dados anterior e da base de dados atual ocorreu de forma eficiente, sem apresentar dificuldades. A navegação entre as abas do site e o uso de suas funcionalidades também transcorreu sem problemas notáveis durante a fase final de implementação[4].

A geração do relatório detalhado em Excel também não apresentou dificuldades, sendo possível baixá-lo em poucos segundos, já que o tamanho final foi de 2.458 KBytes, o que também não gera nenhum obstáculo no seu manuseio, mantendo uma boa responsividade.

Além disso, foi realizada uma avaliação no tempo de execução para avaliar como o script se comportaria com uma base de dados mais robusta. Para isso, foi alterado o tamanho da base de dados, passando a base de dados atual para 100 mil matrículas e a base anterior para 90 mil matrículas. Por padrão, o Shiny limita a importação de arquivos a 5MB por arquivo. Como as bases de dados excediam esse limite, foi realizada uma modificação no código, utilizando a linha: *options(shiny.maxRequestSize=30*1024^2)*. Essa alteração permitiu a importação de arquivos de até 30MB. A implementação dessa modificação garantiu que a análise da base de dados fosse realizada com sucesso. Os resultados obtidos, em relação ao tempo de importação dos arquivos, estão descritos no quadro abaixo:

Quadro 7 - Desempenho da importação de arquivos em diferentes ambientes

N° de Registros	Tempo de Importação (Shinyapps) (s)	Tempo de Importação (Ambiente local[5]) (s)	Dif. Relativa (%)	Tamanho (KB)
20.738	0,09	0,09	0%	1.679
100.000	0,96	0,75	+28%	10.740

Fonte: Elaborado pelo autor (2025).

[4] A aplicação está disponível para acesso em: https://guisa.shinyapps.io/guisa.

[5] Ambiente local refere-se à máquina utilizada nos testes realizados localmente: processador AMD Ryzen 5 7600, armazenamento SSD com velocidade de leitura de até 3500 MB/s e gravação de até 2100 MB/s.

É possível observar que, com o incremento no número de registros, o tempo de importação também aumentou. No entanto, esse aumento foi compatível com a ampliação da base de dados e não comprometeu a experiência de uso em nenhum dos ambientes de execução. Além disso, o desempenho do relatório detalhado, bem como das demais ferramentas e gráficos disponíveis no site, manteve-se satisfatório, confirmando a eficácia do sistema.

Observa-se ainda uma diferença de desempenho entre os testes realizados no ambiente local e na plataforma Shinyapps em relação ao arquivo com mais registros. No ambiente local, o tempo de execução está diretamente relacionado à capacidade da máquina do usuário — principalmente ao desempenho do processador e à velocidade de leitura e gravação do armazenamento —, já que esses componentes influenciam diretamente nos cálculos feitos ao longo do script como também na leitura dos arquivos importados e na geração do relatório final em formato .xlsx. Para mensuração desses tempos, recorreu-se à função *system.time*, aplicada diretamente no código durante os testes.

Vale destacar que o tamanho dos arquivos, em relação à quantidade de registros, foi relativamente reduzido. Na prática, porém, isso nem sempre é observado, uma vez que as planilhas de bases de dados podem conter múltiplas abas, fórmulas, referências externas, links ou formatações que aumentam consideravelmente o tamanho do arquivo.

Como mencionado ao longo do trabalho, o modelo adotado buscou ser o mais generalista possível, sem comprometer a funcionalidade. Nas bases utilizadas, não foram inseridos links externos nem formatações excessivas, e as fórmulas presentes nas planilhas serviram apenas para simulação dos dados. Além disso, essas fórmulas não ficaram registradas nos arquivos finais, uma vez que foi utilizada a função de “colar valores”, garantindo, assim, arquivos mais leves.

5. CONSIDERAÇÕES FINAIS

Este estudo teve como principal objetivo desenvolver uma aplicação que pudesse servir como base para a automatização das críticas às bases de dados de fundos de pensão no Brasil. Para isso, foi adotada a linguagem de programação R, combinada com a plataforma Shinyapps, proporcionando uma interface mais intuitiva e acessível para os usuários. A metodologia utilizada foi detalhada ao longo do trabalho, abordando desde a escolha das ferramentas até a implementação das etapas essenciais para o processamento e análise dos dados. A proposta buscou não apenas viabilizar a automatização dessas críticas, mas também oferecer uma estrutura adaptável a diferentes planos de previdência, garantindo flexibilidade e aplicabilidade em diversos contextos.

A construção da aplicação seguiu uma metodologia estruturada, iniciando com a apresentação dos principais pacotes da linguagem de programação R utilizados neste estudo. Em seguida, descreveu-se o processo de importação da base de dados, que, para fins de demonstração, foi simulada. Nesse contexto, foram utilizadas duas bases de dados, cada uma com datas de referência distintas, permitindo a análise da movimentação dos registros. Por fim, para que os requisitos necessários ao funcionamento do script fossem estabelecidos, foram definidos os parâmetros relacionados ao indexador do plano e ao período de análise, viabilizando o cálculo do indexador acumulado.

As tarefas foram subdivididas em etapas para estruturar a análise de forma clara e objetiva. Primeiramente, realizou-se a análise estatística descritiva da base de dados, possibilitando uma visão inicial sobre a quantidade de participantes aposentados em ambas as bases e a evolução dos benefícios ao longo do tempo. Em seguida, foi detalhada a

movimentação da base, explicando as variações observadas nas estatísticas e identificando as matrículas que ingressaram ou saíram do grupo na condição de aposentado. Por fim, foram analisadas as inconsistências da base de dados por meio de testes lógicos, além da comparação entre as bases atual e anterior, a fim de identificar registros que não estavam dentro do esperado.

Conforme destacado ao longo do trabalho, desde a estruturação da base de dados até a metodologia aplicada para a realização das críticas, buscou-se desenvolver um modelo o mais genérico possível. O objetivo foi tanto facilitar a compreensão do processo quanto permitir a adaptação da aplicação a diferentes planos de fundos de pensão ou outras bases de dados. Assim, aqueles que se interessarem pelo tema poderão realizar as devidas alterações no script,

que foi disponibilizado no apêndice deste trabalho, bem como no repositório público referenciado anteriormente.

Entretanto, a generalização do modelo também traz limitações. Como mencionado anteriormente, cada plano possui suas especificidades, o que exige adaptações. Além disso, este trabalho concentrou-se exclusivamente na análise da massa de aposentados, podendo ser expandido futuramente para incluir pensionistas e participantes ativos. Dessa forma, reforça-se a importância da adequação dos critérios às particularidades de cada entidade.

As limitações relacionadas exclusivamente à funcionalidade do script podem ser divididas em dois grupos. O primeiro abrange fatores externos que não podem ser controlados, como a disponibilidade da API do Banco Central, utilizada para a importação dos indexadores, que pode eventualmente ficar indisponível. Outra restrição refere-se ao limite de 30 MB para a importação de arquivos no Shiny. No entanto, considerando o layout da planilha proposta, esse limite se mostra adequado para grandes bases de dados. Embora essas limitações não tenham sido eliminadas neste trabalho, a criação de alternativas é totalmente plausível.

O segundo grupo de limitações envolve possíveis erros operacionais cometidos durante a construção do script. Tais erros podem ser identificados e corrigidos a partir de revisões críticas e sugestões de melhoria. Para isso, o repositório público do projeto pode servir como um espaço para registrar e discutir eventuais problemas, facilitando sua resolução e aprimoramento contínuo.

Adicionalmente, embora tenham sido implementados diversos mecanismos para tratamento de erros, sempre há a possibilidade de falhas não previstas, exigindo do usuário atenção ao interpretar os resultados. Vale destacar que todas as ferramentas utilizadas neste estudo são gratuitas e de código aberto, o que facilita tanto a replicação quanto a adaptação do modelo para diferentes contextos sem custos adicionais, tornando a proposta acessível a um maior número de interessados.

Além dos aspectos funcionais e metodológicos, também foi considerada a preocupação com a privacidade e a proteção dos dados utilizados na aplicação. A importação das bases ocorre de forma temporária durante a sessão do usuário, não havendo qualquer mecanismo que armazene, compartilhe ou transmita os arquivos carregados para terceiros. O processamento das informações acontece exclusivamente no ambiente da aplicação, respeitando os princípios previstos na Lei Geral de Proteção de Dados (LGPD – Lei nº 13.709/2018). Ainda que a aplicação esteja hospedada em ambiente online, não foi implementado nenhum artifício que armazene os dados no servidor após o uso. Recomenda-se, contudo, que os usuários evitem inserir informações sensíveis em ambientes públicos. Além disso, o formato proposto para a

base de dados foi desenvolvido com esse cuidado, garantindo que não haja colunas com informações pessoais que possam levar à identificação de indivíduos.

De maneira geral, a aplicação desenvolvida demonstrou-se eficaz em seu propósito, oferecendo uma alternativa viável e de fácil usabilidade para otimizar o trabalho já realizado na área atuarial. Como sugestão para estudos futuros, destaca-se o potencial de incorporar tecnologias baseadas em inteligência artificial, como os LLMs (Large Language Models), com o objetivo de aprimorar a análise dos dados e fornecer recomendações mais inteligentes e contextualizadas para solucionar tais inconsistências. Esses modelos podem ser treinados ou ajustados com base em regras validadas por profissionais atuariais e instituições especializadas, como a TPR e a SOA como também métodos tradicionais utilizados no cenário brasileiro, ampliando a qualidade e a precisão das críticas automatizadas. Em relação à limitação de tamanho da base de dados, recomenda-se explorar alternativas como a utilização de formatos mais compactos em alternativa ao formato .xlsx ou a migração para outras plataformas, sejam elas gratuitas ou comerciais, que comportem volumes maiores de dados e ofereçam maior escalabilidade para aplicações mais complexas.

Por fim, espera-se que este estudo sirva como incentivo para novas pesquisas e estimule a aplicação prática da programação no campo da atuária, promovendo o desenvolvimento de ferramentas mais inteligentes, acessíveis e alinhadas às necessidades reais do setor.

REFERÊNCIAS

ASSOCIAÇÃO BRASILEIRA DAS ENTIDADES FECHADAS DE PREVIDÊNCIA COMPLEMENTAR (Abrapp). **Consolidado Estatístico de setembro de 2024**, 2025. Disponível em: https://www.abrapp.org.br/consolidado-estatistico/. Acesso em: 25 fev. 2025.

BRASIL. Ministério da Economia. **Instrução n°23, de 26 de junho de 2015.** [*S. l.*]: Ministério da Economia, 2015. Atualizado em 29 de dez. de 2020. Disponível em: https://www.gov.br/economia/pt-br/orgaos/entidades-vinculadas/autarquias/previc/regulacao/normas/instrucoes/instrucoes-previc/2015/instrucao-ndeg23-de-26-de-junho-de-2015.pdf/view Acesso em: 06 out. 2023.

BRASIL. **Lei Nº 13.709, de 14 de agosto de 2018**. Lei Geral de Proteção de Dados Pessoais (LGPD). Diário Oficial da União, Brasília, 14 ago. 2018. Disponível em: https://www.planalto.gov.br/ccivil_03/_ato2015-2018/2018/lei/l13709.htm. Acesso em: 05 mai. 2025.

BRASIL. Ministério da Previdência Social. **O que é Previdência Complementar.** [*S. l.*]: Ministério da Previdência Social, 08 jun. 2021. Atualizado em 19 de jul. de 2021. Disponível em: https://www.gov.br/previdencia/pt-br/assuntos/previdencia-complementar/mais-informacoes/o-que-previdncia-complementar. Acesso em: 05 out. 2023.

BRASIL. Ministério da Previdência Social. **Perguntas Frequentes de Previdência Complementar.** [*S. l.*]: Ministério da Previdência Social, 16 jul. 2021. Atualizado em 19 de jul. de 2021. Disponível em: https://www.gov.br/previdencia/pt-br/assuntos/previdencia-complementar/mais-informacoes/perguntas-frequentes-de-previdencia-complementar Acesso em: 23 set. 2023.

BRASIL. Ministério da Previdência Social. **Entidades Fechadas de Previdência Complementar (EFPC).** [*S. l.*]: Ministério da Previdência Social, 2022. Atualizado em 19 de jul. de 2022. Disponível em: https://www.gov.br/previc/pt-br/previdencia-complementar-fechada/entidades-fechadas-de-previdencia-complementar-efpc Acesso em: 09 out. 2023.

BRASIL. Ministério da Previdência Social. **Relatório Gerencial de Previdência Complementar.** [*S. l.*]: Ministério da Previdência Social, 2024. Atualizado em dezembro de 2024. Disponível em: https://www.gov.br/previdencia/pt-br/assuntos/previdencia-complementar/rgpc/2024/rgpc_2024_3tri.pdf Acesso em: 25 fev. 2025.

BRASIL. Ministério da Previdência Social. **A demografia dos fundos de pensão**. Brasília: MPS, 2007. 292 f. Disponível em: http://sa.previdencia.gov.br/site/arquivos/office/3_081014-111404-315.pdf. Acesso em: 13 out. 2024.

BRASILIS CONSULTORIA ATUARIAL. **Relatório de avaliação atuarial 2017: Regime Próprio de Previdência Social do Estado da Paraíba – PBPREV**. [*S. l.: s. n.*]: 2018. Disponível em: http://www.pbprev.pb.gov.br/midia/exibir/Avalia%C3%A7%C3%A3o%20Atuarial%20PBPREV%202018.pdf. Acesso em: 06 mai. 2025.

BRASILIS CONSULTORIA ATUARIAL. **Relatório de avaliação atuarial 2025: Regime Próprio de Previdência Social do Município de São José do Rio Preto – RIOPRETOPREV**. Versão 01. [*S. l.: s. n.*]: 18 fev. 2025. Disponível em: https://novopainel.riopreto.sp.gov.br/uploads/avaliacao_atuarial_2025_edad60aee8.pdf. Acesso em: 06 mai. 2025.

CORRÊA, Cristiane Silva. **Premissas Atuariais em Planos Previdenciários**. Curitiba, Editora: Appris, 2018.

CORRÊA, Cristiane Silva. **Tamanho populacional e aleatoriedade de eventos demográficos na solvência de RPPS municipais capitalizados**. 2014. 274 f. Tese (Doutorado) – Curso de Demografia, Universidade Federal de Minas Gerais, Belo Horizonte, 2014.

CURSO-R. **Ciências de Dados em R**. Disponível em: https://livro.curso-r.com/12-1-arrumando-banco-de-dados-o-pacote-janitor.html. Acesso em: 14 out. 2024.

DUBEL, Marcin et al. **data.validator: Automatic Data Validation and Reporting**. Versão 0.2.1. [*S. l.*]: CRAN, 2023. Disponível em: https://CRAN.R-project.org/package=data.validator. Acesso em: 14 out. 2024.

FERREIRA, Bruno Pérez. **Análise do risco de não superação da meta atuarial em fundos de previdência**. 2006. 83 f. Dissertação (Mestrado) — Universidade Federal de Minas Gerais, Belo Horizonte, 2006.

FIRKE, Sam et al. **janitor: Simple Tools for Examining and Cleaning Dirty Data**. Versão 2.2.0. [*S. l.*]: CRAN, 2023. Disponível em: https://CRAN.R-project.org/package=janitor. Acesso em: 14 out. 2024.

FREITAS, Wilson. **rbcb: R Interface to Brazilian Central Bank Web Services**. Versão 0.1.14. [*S. l.*]: CRAN, 2024. Disponível em: https://cran.r-project.org/package=rbcb. Acesso em: 14 out. 2024.

GAMA CONSULTORES ASSOCIADOS. **Guia Prático de Utilização da Instrução Previc Nº 23/2015**, 2015. Disponível em: https://www.editoraroncarati.com.br/v2/Artigos-e-

Noticias/Artigos-e-Noticias/Guia-pratico-de-utilizacao-da-instrucao-Previc-IN-n%C2%BA-23.html. Acesso em: 06 out. 2023.

GUIA PREVIC. **Melhores Práticas Atuariais para Entidades Fechadas de Previdência Complementar**, 2022. Disponível em: https://www.gov.br/economia/pt-br/orgaos/entidades-vinculadas/autarquias/previc/centrais-de-conteudo/publicacoes/guias-de-melhores-praticas/novo-guia-previc-melhores-atuariais.pdf

GUILLOT, Antoine. **Machine learning explained: Vectorization and matrix operations.** R-bloggers, 7 mai. 2018. Disponível em: https://www.r-bloggers.com/2018/05/machine-learning-explained-vectorization-and-matrix-operations/. Acesso em: 13 jan. 2025.
IBM. **What is data profiling?** Disponível em: https://www.ibm.com/topics/data-profiling. Acesso em: 24 set. 2023.

LAZZARI, João B.; CASTRO, Carlos Alberto Pereira de. **Direito Previdenciário**. [*S. l.*]: Grupo GEN, 2021. E-book. ISBN 9788530990756. Disponível em: https://integrada.minhabiblioteca.com.br/#/books/9788530990756/ Acesso em: 18 out. 2024.

LÓGICA CONSULTORIA ATUARIAL. **Relatório de avaliação atuarial 2024: Regime Próprio de Previdência Social do Município de Lages – LAGESPREVI**. [*S. l.: s. n.*]: 02 abr. 2024. Disponível em: https://www.lagesprevi.sc.gov.br/arquivos_admin_financeiro/1717680885345.pdf. Acesso em: 06 mai. 2025.

MARCONI, M. A.; LAKATOS, E. M. **Fundamentos de metodologia científica**. 8. ed. São Paulo: Atlas, 2017.

NESE, Arlete; GIAMBIAGI, Fabio. Fundamentos da Previdência Complementar - Da Administração à Gestão de Investimentos. [*S. l.*]: Grupo GEN, 2019. E-book. ISBN 9788595150188. Acesso em: 16 abr. 2025.

OSBORNE, J. **Best Practices in Data Cleaning:** A Complete Guide to Everything You Need to Do Before and After Collecting Your Data. 1º. ed. [*S. l.*]: SAGE, 2012.

PREVCOM. **Relatório Anual de Informações – Exercício de 2021**. Belo Horizonte: PREVCOM-MG, 2022. Disponível em: https://prevcommg.com.br/relatorio-anual/. Acesso em: 01 maio 2025.

RODRIGUES, José Ângelo. **Gestão de risco atuarial.** São Paulo: Saraiva, 2008.

ROMAN, Norton Trevisan. **MC102 - Algoritmos e Programação de Computadores, Parte X.** Disponível em: https://www.ic.unicamp.br/~norton/disciplinas/mc102/pascal10.html. Acesso em: 13 out. 2024.

SCHAUBERGER, P.; WALKER, A. **openxlsx: Read, Write and Edit xlsx Files**. Versão 4.2.7.1. [*S. l.*]: CRAN, 2024. Disponível em: https://CRAN.R-project.org/package=openxlsx. Acesso em: 14 out. 2024.

SEVERINO, J. A. **Metodologia do trabalho científico**. 1. ed. São Paulo: Cortez, 2013.

SHAW, Anirban. **Data Validation With data.validator: An Open-Source Package from Appsilon**. R-bloggers, 13 jul. 2021. Disponível em: https://www.r-bloggers.com/2021/07/data-validation-with-data-validator-an-open-source-package-from-appsilon/. Acesso em: 14 out. 2024.

SOCIETY OF ACTUARIES - SOA. **Experience Data Quality How to Clean and Validate Your Data. 2011.** Disponível em: https://www.soa.org/globalassets/assets/Files/Research/research-2011-12-data-quality.pdf Acesso em: 18 set. 2024.

SOCIETY OF ACTUARIES - SOA. **Experience Study Calculations. 2016.** Disponível em: https://www.soa.org/globalassets/assets/Files/Research/2016-10-experience-study-calculations.pdf. Acesso em: 05 out. 2024.

SPINU, Vitalie et al. **lubridate: Make Dealing with Dates a Little Easier**. Versão 1.9.3. [*S. l.*]: CRAN, 2023. Disponível em: https://CRAN.R-project.org/package=lubridate. Acesso em: 14 out. 2024.

SUPERINTENDÊNCIA NACIONAL DE PREVIDÊNCIA COMPLEMENTAR (PREVIC). **Informe Estatístico – 3 Trimestre 2024**, 2025. Disponível em: https://www.gov.br/previc/pt-br/publicacoes/informe-estatistico-trimestral/2024/informe-estatistico-trimestral-3deg-trimestre/view. Acesso em: 01 abr. 2025.

SUPERINTENDÊNCIA NACIONAL DE PREVIDÊNCIA COMPLEMENTAR (PREVIC). **Instrução Previc nº 33, de 23 de Outubro de 2020**, 2020. Atualizado em 14 de jul. de 2022. Disponível em: https://www.gov.br/previc/pt-br/normas/instrucao/instrucoes-previc/2020/instrucao-previc-no-33-de-23-de-outubro-de-2020.pdf/view Acesso em: 23 set. 2023.

SUPERINTENDÊNCIA NACIONAL DE PREVIDÊNCIA COMPLEMENTAR (PREVIC). **Relatório 2023 da Previdência Complementar Fechada - Maio 2024**, 2024. Disponível em: https://www.gov.br/previc/pt-br/publicacoes/relatorio-de-estabilidade-da-previdencia-complementar-rep Acesso em: 05 mai. 2025.

THE PENSIONS REGULATOR - TPR. **Review your pension scheme data,** [202-?]. Disponível em: https://www.thepensionsregulator.gov.uk/en/trustees/contributions-data-and-transfers/record-keeping/review-your-scheme-data Acesso em: 24 set. 2024.

VANZILLOTTA, Andrea. **Curso Avaliação Atuarial em Planos de Previdência**. YouTube, 23 de novembro de 2022. 1 vídeo (2h39min55s). Disponível em: https://youtu.be/up84i0d99w0. Acesso em: 01 de abril de 2025.

WICKHAM, H. **Mastering Shiny**. [*S. l.*]: O'Reilly Media, 2021. Disponível em: https://mastering-shiny.org/index.html. Acesso em: 21 nov. 2024.

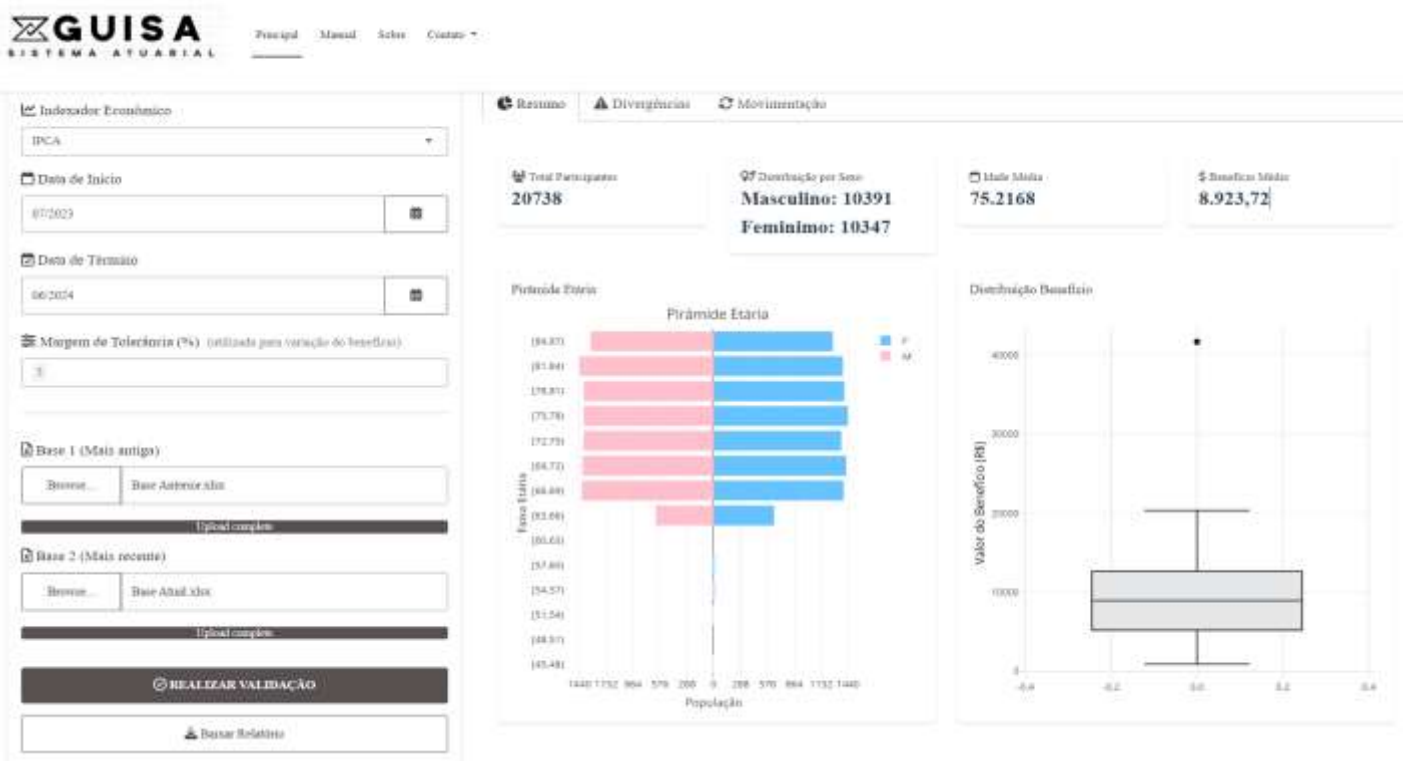
WICKHAM, H.; BRYAN, J. et al. **readxl: Read Excel Files**. Versão 1.4.3. [*S. l.*]: CRAN, 2023. Disponível em: https://CRAN.R-project.org/package=readxl. Acesso em: 14 out. 2024.

WICKHAM, H. **tidyverse: Easily Install and Load the 'Tidyverse'**. Versão 2.0.0. [*S. l.*]: CRAN, 2023. Disponível em: https://CRAN.R-project.org/package=tidyverse. Acesso em: 14 out. 2024.

WRIGHT, K. **lookup: Functions Similar to VLOOKUP in Excel**. Versão 1.0 [*S. l.*]: CRAN, 2021. Disponível em: https://CRAN.R-project.org/package=lookup. Acesso em: 02 abr. 2025.

APÊNDICES

APÊNDICE A – INTERFACE WEB



APÊNDICE B – CÓDIGO

```
# Gerenciamento de pacotes
if (!require("pacman")) install.packages("pacman")
pacman::p_load(
  shiny,
  bslib,
  shinyWidgets,
  DT,
  rbcb,
  lubridate,
  openxlsx,
  readxl,
  dplyr,
  plotly,
  ggplot2,
  lookup,
  shinyBS
)

# Constantes e configurações
CONFIG <- list(
  THEME = bs_theme(
    bg = "#ffffff",
    fg = "#333333",
    primary = "#575152",
    base_font = "Montserrat",
    heading_font = "Montserrat",
    font_scale = 1.1
  ),
  INDEXADORES = c("IPCA" = 433, "IGPM" = 189, "INPC" = 188, "SELIC" = 432, "TR" =
226),
```

```
  DATE_MIN = "2000-01",
  DATE_MAX = format(Sys.Date(), "%Y-%m")
)

# Funções auxiliares
get_accumulated_index <- function(index_code, start_date, end_date) {
  tryCatch({
    serie <- rbcb::get_series(
      index_code,
      start_date = start_date,
      end_date = end_date
    )
    accumulated <- prod(1 + (serie[2] / 100)) - 1
    return(list(success = TRUE, value = accumulated, error = NULL))
  }, error = function(e) {
    return(list(success = FALSE, value = NULL, error = e$message))
  })
}

# UI
ui <- navbarPage(
  theme = CONFIG$THEME,

  tags$head(
    includeCSS("www/styles.css"),
    tags$style(HTML("
      .navbar {
        padding-top: 0px;         /* Reduz o padding superior */
        padding-bottom: 0px;      /* Reduz o padding inferior */
      }
      body {
        overflow-y: auto; /* Remove a barra de rolagem vertical */
```

```
    }
    .navbar-page {
      max-width: 800px; /* Define a largura máxima */
      max-height: 600px; /* Define a altura máxima */
      overflow-y: auto; /* Permite rolagem interna nos painéis se necessário */
    }
    .tab-content {
      max-height: 500px; /* Altura máxima do conteúdo de cada aba */
    }
  ")),
  tags$link(rel = "shortcut icon", href = "favicon.ico")
 ),
 title = tags$div(
  tags$a(
   href = "#",  # Link para onde o logo leva, se aplicável
   tags$img(src = "Screenshot_2.png", height = "75px", style = "margin-right: 15px;")  # Caminho da imagem da logo
  )
 ),
 # Tab Principal
 tabPanel("Principal",
          tags$head(
           includeCSS("www/styles.css")
          ),

          sidebarLayout(
           sidebarPanel(
            selectInput(
             "indexador",
             tags$span(icon("chart-line"), "Indexador Econômico"),
             choices = names(CONFIG$INDEXADORES),
             selected = "IPCA"
```

```
      ),

      airMonthpickerInput(
        "data_inicio",
        tags$span(icon("calendar"), "Data de Início"),
        minDate = CONFIG$DATE_MIN,
        maxDate = CONFIG$DATE_MAX,
        view = "months",
        dateFormat = "MM/yyyy"
      ),

      airMonthpickerInput(
        "data_fim",
        tags$span(icon("calendar-check"), "Data de Término"),
        minDate = CONFIG$DATE_MIN,
        maxDate = CONFIG$DATE_MAX,
        view = "months",
        dateFormat = "MM/yyyy"
      ),

      selectizeInput(
        "margem_tolerancia",
        label = tags$div(
          icon("sliders-h"),
          "Margem de Tolerância (%)",
          tags$small("(utilizada para variação do benefício)", style = "color: gray; margin-
left: 5px; font-weight: normal;")
        ),
        choices = c(1, 2, 3, 4, 5, 10),
        multiple = TRUE,
        selected = 5
      ),
```

```
  hr(),

  fileInput(
    "base1",
    tags$span(icon("file-excel"), "Base 1 (Mais antiga)"),
    accept = c(".xlsx", ".xls")
  ),

  fileInput(
    "base2",
    tags$span(icon("file-excel"), "Base 2 (Mais recente)"),
    accept = c(".xlsx", ".xls")
  ),

  actionButton(
    "validar",
    tags$span(icon("check-circle"), "Realizar Validação"),
    class = "btn-validar"
  ),

  # Adicionando o botão de download aqui
  uiOutput("download_button")
),

mainPanel(
  tabsetPanel(
    tabPanel(
      tags$span(icon("chart-pie"), "Resumo"),
      br(),
      fluidRow(
        column(3,
```

```
                div(class = "info-box",
                    style = "padding: 15px; background: white; border-radius: 5px; box-shadow: 0 2px 4px rgba(0,0,0,0.1); margin-bottom: 20px;",
                    h4(icon("users"), "Total Participantes", style = "font-size: 14px; margin: 0;"),
                    div(style = "font-size: 24px; font-weight: bold; color: #2c3e50;",
                        textOutput("total_participantes")
                    )
                )
            ),
            column(3,
                div(class = "info-box",
                    style = "padding: 15px; background: white; border-radius: 5px; box-shadow: 0 2px 4px rgba(0,0,0,0.1); margin-bottom: 20px;",
                    h4(icon("venus-mars"), "Distribuição por Sexo", style = "font-size: 14px; margin: 0;"),
                    div(style = "font-size: 24px; font-weight: bold; color: #2c3e50;",
                        textOutput("dist_sexo")
                    )
                )
            ),
            column(3,
                div(class = "info-box",
                    style = "padding: 15px; background: white; border-radius: 5px; box-shadow: 0 2px 4px rgba(0,0,0,0.1); margin-bottom: 20px;",
                    h4(icon("calendar"), "Idade Média", style = "font-size: 14px; margin: 0;"),
                    div(style = "font-size: 24px; font-weight: bold; color: #2c3e50;",
                        textOutput("idade_media")
                    )
                )
            ),
            column(3,
                div(class = "info-box",
```

```
                    style = "padding: 15px; background: white; border-radius: 5px; box-
shadow: 0 2px 4px rgba(0,0,0,0.1); margin-bottom: 20px;",
                    h4(icon("dollar-sign"), "Benefício Médio", style = "font-size: 14px;
margin: 0;"),
                    div(style = "font-size: 24px; font-weight: bold; color: #2c3e50;",
                      textOutput("beneficio_medio")
                    )
                  )
                )
              ),

              # Gráficos (mantidos apenas os dois primeiros)
              fluidRow(
                column(6,
                  div(class = "chart-box",
                    style = "padding: 15px; background: white; border-radius: 5px; box-
shadow: 0 2px 4px rgba(0,0,0,0.1); margin-bottom: 20px;",
                    h4("Pirâmide Etária", style = "font-size: 16px; margin-bottom: 15px;"),
                    plotlyOutput("piramide_etaria", height = "500px")
                  )
                ),
                column(6,
                  div(class = "chart-box",
                    style = "padding: 15px; background: white; border-radius: 5px; box-
shadow: 0 2px 4px rgba(0,0,0,0.1); margin-bottom: 20px;",
                    h4("Distribuição Benefício", style = "font-size: 16px; margin-bottom:
15px;"),
                    plotlyOutput("dist_beneficio", height = "500px")
                  )
                )
              )
            ),
            tabPanel(
              tags$span(icon("exclamation-triangle"), "Divergências"),
```

```
            br(),
            DTOutput("divergencias")
          ),
          tabPanel(
            tags$span(icon("sync"), "Movimentação"),
            br(),
            DTOutput("movimentacao")
          )
        )
      )
    )
  ),

  # Tab Manual
  tabPanel("Manual",
           fluidRow(
             column(
               8, offset = 2,
               div(
                 class = "manual-content",
                 style = "padding: 20px;",
                 h1("Manual do Usuário", align = "center"),
                 br(),

                 h2("1. Seleção do Indexador"),
                 p("Escolha o indexador econômico desejado (IPCA, IGPM, INPC, SELIC ou TR). O indexador auxiliará na identificação de variações nos benefícios e se elas ocorreram conforme o esperado."),

                 h2("2. Período de Análise"),
                 p("Defina o período de análise selecionando:"),
                 tags$ul(
```

```
        tags$li(HTML("<strong>Data de Início</strong>: primeiro mês do período a ser analisado.")),

        tags$li(HTML("<strong>Data de Término</strong>: último mês do período a ser analisado. As datas são essenciais, pois a partir delas será calculado o valor acumulado do indexador."))

      ),


      h2("3. Importação das Bases"),

      p("Realize o upload de duas bases de dados em formato Excel (.xlsx ou .xls):"),

      tags$ul(

        tags$li(HTML("<strong>Base 1</strong>: arquivo mais antigo.")),

        tags$li(HTML("<strong>Base 2</strong>: arquivo mais recente."))

      ),

      p("Após carregar a ", HTML("<strong>Base 2</strong>"), ", o aplicativo gerará automaticamente uma ",

        HTML("<strong>pirâmide etária</strong>"), " da população e um ",

        HTML("<strong>gráfico boxplot</strong>"), ", resumindo o comportamento do benefício complementar. Além disso, serão exibidas acima dos gráficos as estatísticas mais relevantes da base."),

      p(HTML("<em>A importação das bases deve seguir o layout padrão, que pode ser baixado clicando no botão \"Baixar modelo de Planilhas\" ao final da página. A partir desse modelo, você poderá ajustar os dados conforme necessário, garantindo a manutenção da formatação inicial e do layout da planilha.</em>")),


      h2("4. Resultados"),

      p("Após importar ambas as bases, selecionar o indexador e definir o período de análise, clique no botão ",

        HTML("<strong>\"Realizar Validação\"</strong>"), " para processar os dados. Em seguida, as seguintes abas estarão disponíveis:"),

      tags$ul(

        tags$li(HTML("<strong>Resumo</strong>: visão geral dos parâmetros e resultados da validação.")),

        tags$li(HTML("<strong>Divergências</strong>: lista detalhada de registros com diferenças.")),

        tags$li(HTML("<strong>Movimentação</strong>: análise das alterações entre as bases."))

      ),
```

```
            p("Para um relatório detalhado, clique no botão ",
              HTML("<strong>\"Baixar Relatório\"</strong>"),
              " ao final da página. Uma planilha no formato .xlsx será gerada para sua análise."),

            br(),
            downloadButton("download_planilhas", "Baixar Modelo de Planilhas")
          )
        )
      )
    ),

    # Tab Sobre
    tabPanel("Sobre",
      fluidRow(
        column(
          8, offset = 2,
          div(
            class = "sobre-content",
            style = "padding: 20px; text-align: justify;",
            h1("Sobre", align = "center"),
            br(),
            p("O GUISA - Sistema Atuarial é uma aplicação web desenvolvida como parte da pesquisa de conclusão do curso de Ciências Atuariais na
    Universidade Federal da Paraíba (UFPB), por João Guilherme Pereira dos Santos, sob a orientação do Prof. Dr. Luiz Carlos Junior Santos.
    A aplicação foi projetada para oferecer uma ferramenta crítica e analítica das bases de dados gerenciadas por fundos de pensão no Brasil.
    O objetivo principal do GUISA é agilizar e otimizar o processo de análise de dados, fornecendo, por meio de uma interface intuitiva e das
    informações fornecidas pelo usuário, três abas principais:"),
            tags$ul(
              tags$li(tags$b("Dashboard Resumo"), " – Exibe uma análise prévia da massa de participantes, permitindo uma visão inicial dos dados."),
```

```
            tags$li(tags$b("Tabela de Inconsistências"), " – Identifica e exibe divergências e inconsistências na base de dados entre períodos."),

            tags$li(tags$b("Aba de Movimentação"), " – Mostra as entradas e saídas dos participantes ao longo do tempo.")

          ),

          p("Além disso, o sistema gera um relatório detalhado em formato .xlsx, permitindo uma visualização aprofundada das críticas e movimentações da base de dados.")

        )

      )

    )

  ),

  navbarMenu("Contato",

      HTML('<li><a href="https://mail.google.com/mail/?view=cm&fs=1&to=joaosantos34537445@gmail.com&su=Assunto&body=Corpo%20do%20email" target="_blank" style="font-size: 14px;">

          <img src="email.svg" width="14px" style="margin-right: 10px;"> E-mail</a></li>'),

      HTML('<li><a href="https://api.whatsapp.com/send?phone=5581999911222" target="_blank" style="font-size: 14px;">

          <img src="whatsapp.svg" width="14px" style="margin-right: 10px;"> WhatsApp</a></li>'),

      HTML('<li><a href="https://github.com/jgpds" target="_blank" style="font-size: 14px;">

          <img src="github.svg" width="14px" style="margin-right: 10px;"> GitHub</a></li>'),

      HTML('<li><a href="https://www.linkedin.com/in/joaoguilhermepsantos/" target="_blank" style="font-size: 14px;">

          <img src="linkedin.svg" width="17px" style="margin-right: 10px;"> LinkedIn</a></li>')

  ),

  position = "fixed-top"

)

# Server

server <- function(input, output, session) {

  # Aumentar o limite de tamanho de importação de arquivo para 30MB

  options(shiny.maxRequestSize=30*1024^2)
```

```
  # Reactive values
  rv <- reactiveValues(
    validation_status = FALSE,
    current_params = NULL,
    validation_results = NULL,
    base1_data = NULL,
    base2_data = NULL
  )

  # Observer para ler Base 1
  observeEvent(input$base1, {
    req(input$base1)
    tryCatch({
      rv$base1_data <- read_excel(input$base1$datapath, col_types = c("numeric", "date",
"text", "date", "text", "numeric", "numeric", "date", "text", "date", "text", "date", "text"))

      # Alternativa: mostrar estrutura completa
      print("Estrutura da Base 1:")
      print(str(rv$base1_data))

      showNotification("Base 1 carregada com sucesso!", type = "message")
    }, error = function(e) {
      showNotification(paste("Erro ao carregar Base 1:", e$message), type = "error")
    })
  })

  # Observer para ler Base 2
  observeEvent(input$base2, {
    req(input$base2)
    tryCatch({
      rv$base2_data <- read_excel(input$base2$datapath, col_types = c("numeric", "date",
"text", "date", "text", "numeric", "numeric", "date", "text", "date", "text", "date", "text"))
```

```
    # Alternativa: mostrar estrutura completa
    print("Estrutura da Base 2:")
    print(str(rv$base2_data))

    showNotification("Base 2 carregada com sucesso!", type = "message")
  }, error = function(e) {
    showNotification(paste("Erro ao carregar Base 2:", e$message), type = "error")
  })
 })

 # Modificação no server para o botão de download
 output$download_button <- renderUI({
  if (rv$validation_status) {
    downloadButton(
      "downloadData",
      label = "Baixar Relatório ",
      class = "btn-download btn-block", # Adicionado btn-block para largura total
      style = "margin-top: 15px; width: 100%;" # Aumentado margin-top e definido width
100%
    )
  }
 })

 output$piramide_etaria <- renderPlotly({
  req(rv$base2_data)
  dados <- rv$base2_data
  datas_nascimento <- as.Date(dados$`Data de Nascimento`, origin = "1899-12-30")
  data_atual <- as.Date(input$data_fim, origin = "1899-12-30") # mesma base de data
  idades <- as.numeric(difftime(data_atual, datas_nascimento, units = "weeks")) %/% 52
  sexo <- dados$Sexo

  # Criar um dataframe com as idades e sexo
```

```
  dados <- data.frame(idade = idades, sexo = sexo)

  # Criar faixas etárias (0-4, 5-9, ..., 95-99) e contar por faixa etária e sexo
  dados <- dados %>%
    mutate(faixa_etaria = cut(idade, breaks = seq(0, 100, by = 3), right = FALSE)) %>%
    group_by(faixa_etaria, sexo) %>%
    summarise(count = n(), .groups = "drop") %>%
    mutate(count = ifelse(sexo == "M", -count, count)) %>%
    filter(count != 0)  # Remover faixas etárias vazias

  # Definir intervalo e limites do eixo X dinamicamente
  tick_max <- max(abs(dados$count))
  tick_interval <- round(tick_max / 5)  # Define intervalos proporcionais ao maior valor

  # Criar o gráfico de pirâmide etária
  plot_ly(dados, x = ~count, y = ~faixa_etaria, color = ~sexo, colors = c("#66C3FF", "pink"),
          type = 'bar', orientation = 'h') %>%
    layout(barmode = 'relative',
           xaxis = list(title = 'População', tickvals = seq(-tick_max, tick_max, by =
tick_interval),
                        ticktext = abs(seq(-tick_max, tick_max, by = tick_interval)),
                        tickangle = 0),  # Mantém os rótulos retos
           yaxis = list(title = 'Faixa Etária'),
           title = "Pirâmide Etária")
 })



 output$dist_beneficio <- renderPlotly({
  req(rv$base2_data)

  tryCatch({
    # Extrai e limpa os dados de benefícios
```

```
    dados <- rv$base2_data
    beneficios <- as.numeric(gsub("[^0-9.]", "", dados[[6]]))

    # Remove NAs e valores negativos ou zero
    beneficios <- beneficios[!is.na(beneficios) & beneficios > 0]

    # Cria o boxplot básico com ggplot2
    p <- ggplot(data.frame(Beneficio = beneficios), aes(y = Beneficio)) +
      geom_boxplot(outlier.colour = "black", fill = "#e6e7e8") +
      labs(title = "", y = "Valor do Benefício (R$)") +
      theme_minimal()

    # Converte para um gráfico interativo com ggplotly
    ggplotly(p)

  }, error = function(e) {
    plot_ly() %>%
      add_annotations(
        text = "Erro ao gerar o gráfico. Verifique os dados.",
        showarrow = FALSE
      )
  })
})

# Calcular o total de participantes
output$total_participantes <- renderText({
  total <- nrow(rv$base2_data)
  return(total)
})

output$idade_media <- renderText({
  # Datas fornecidas
```

```
   datas_nascimento <- as.Date(rv$base2_data[[2]], origin = "1899-12-30") # para o sistema de data do Excel (base 1900)
   data_atual <- as.Date(input$data_fim, origin = "1899-12-30") # mesma base de data

   # Calcular idade
   idade_media_aux <- mean(as.numeric(difftime(data_atual, datas_nascimento, units = "weeks")) %/% 52)
   return(idade_media_aux)
  })

  output$beneficio_medio <- renderText({
   req(rv$base2_data)  # Garante que os dados existem

   # Pega o nome da coluna de benefício (ajuste conforme sua base)
   coluna_beneficio <- names(rv$base2_data)[6]

   # Converte para numérico, removendo caracteres não numéricos
   beneficios <- as.numeric(gsub("[^0-9.]", "", rv$base2_data[[6]]))

   # Calcula a média, removendo NAs
   ben_medio <- mean(beneficios, na.rm = TRUE)

   # Formata o resultado
   return(format(round(ben_medio, 2), big.mark = ".", decimal.mark = ","))
  })

  output$dist_sexo <- renderText({
   req(rv$base2_data)  # Garante que os dados existem

   num_homens <- sum(ifelse(rv$base2_data[[3]] == "M", 1, 0))
   num_mulheres <- sum(ifelse(rv$base2_data[[3]] == "F", 1, 0))

   # Formata o resultado
```

```
  return(paste0("Masculino: ", num_homens, "\nFeminimo: ", num_mulheres))
 })

 # Link download sobre
 output$download_planilhas <- downloadHandler(
  filename = function() {
   paste("Modelos-Planilhas-", Sys.Date(), ".zip", sep = "")
  },
  content = function(file) {
   # Diretório dos arquivos existentes
   file1 <- "datasets/Base Anterior.xlsx"
   file2 <- "datasets/Base Atual.xlsx"

   # Compactando os arquivos em um único ZIP
   zip::zipr(file, files = c(file1, file2))
  }
 )



 # Observe validation button click
 observeEvent(input$validar, {
  # Verificar se as bases foram carregadas
  if (is.null(rv$base1_data) || is.null(rv$base2_data)) {
   showNotification("Por favor, carregue ambas as bases antes de validar", type =
"warning")
   return()
  }

  # Parse dates
  data_inicio <- as.Date(paste0(input$data_inicio, "-01"))
  data_fim <- as.Date(paste0(input$data_fim, "-01"))
```

```
  # Validate date range
  if (data_fim < data_inicio) {
    showNotification("Data de término deve ser posterior à data de início", type = "error")
    return()
  }

  # Get index data
  index_result <- get_accumulated_index(
    CONFIG$INDEXADORES[input$indexador],
    data_inicio,
    data_fim
  )

  if (!index_result$success) {
    showNotification(paste("Erro ao obter dados do indexador:", index_result$error), type =
"error")
    return()
  }

  # Store parameters
  rv$current_params <- list(
    indexador = input$indexador,
    data_inicio = format(data_inicio, "%m/%Y"),
    data_fim = format(data_fim, "%m/%Y"),
    acumulado = index_result$value
  )

  rv$validation_status <- TRUE
 })

 # Download handler
```

```
 output$downloadData <- downloadHandler(
  filename = function() {
   paste0("resultados-", format(Sys.Date(), "%Y%m%d"), ".xlsx")
  },
  content = function(file) {
   req(rv$current_params)
   req(rv$base2_data)
   req(rv$base1_data)

   dados <- rv$base2_data
   datas_nascimento <- as.Date(dados$`Data de Nascimento`, origin = "1899-12-30")
   data_atual <- as.Date(input$data_fim, origin = "1899-12-30") # mesma base de data
   idades <- as.numeric(difftime(data_atual, datas_nascimento, units = "weeks")) %/% 52
   sexo <- dados$Sexo

   # Salário Mínimo
   sal_minimo <- rbcb::get_series(1619
                                  , start_date = input$data_inicio, end_date = input$data_fim)
   sal_minimo <- tail(sal_minimo$`1619`, n = 1)

   base_anterior <- rv$base1_data
   base_atual <- rv$base2_data

   # Interseção
   matrículas_comuns  <- intersect(base_atual$Matrícula, base_anterior$Matrícula)
   base_atual_x_anterior <- data.frame(Matrículas = matrículas_comuns)

   # Bases Auxiliares
   divergencias_atual <- tibble(
    Matrícula = base_atual$Matrícula,
    `Matrículas Repetidas` = ifelse(duplicated(base_atual$Matrícula) |
duplicated(base_atual$Matrícula, fromLast = TRUE), 1, 0),
```

```
    `Idade maior que 90 anos` = ifelse(idades > 90, 1, 0),
    `Benefício menor que o salário mínimo` = ifelse(base_atual$`Benefício Complementar` < sal_minimo, 1, 0),
    `Data de Início de Benefício menor que a Data de Nascimento` = ifelse(base_atual$`Data de Início de Benefício` < base_atual$`Data de Nascimento`, 1, 0))


  divergencias_atual_anterior <- tibble(
    Matrícula = base_atual_x_anterior$`Matrículas`,
    `Data de Nascimento divergente` = rep(NA,nrow(base_atual_x_anterior)),
    `Sexo divergente` = rep(NA,nrow(base_atual_x_anterior)),
    `Tipo de Benefício divergente` = rep(NA,nrow(base_atual_x_anterior)),
    `Benefício Complementar com variações significativas` = rep(NA,nrow(base_atual_x_anterior)))


  # Críticas
  criticas <- data.frame(
    Critica = c(
      "Matrículas Repetidas",
      "Idade maior que 90 anos",
      "Benefício menor que o salário mínimo",
      "Data de Início de Benefício menor que a Data de Nascimento",
      "Data de Nascimento divergente",
      "Sexo divergente",
      "Tipo de Benefício divergente",
      "Benefício Complementar com variações significativas"
    ),
    Frequencia = NA, # inicializando com NA, para depois preencher
    Percentual_Sobre_a_Base = NA # inicializando com NA, para depois preencher
  )
  colnames(criticas) <- c("Críticas", "Frequência", "Percentual Sobre a Base")


  # 1. Matrículas Repetidas
```

```
    criticas[1, 2] <- sum(ifelse(duplicated(rv$base2_data$Matrícula) |
duplicated(rv$base2_data$Matrícula, fromLast = TRUE), 1, 0)) # Ocorrëncias

    criticas[1, 3] <- paste0(round(criticas[1, 2]/nrow(base_atual) * 100, 2), "%") #
Porcentagem sobre a base


    # 2. Idade maior que 90 anos

    criticas[2, 2] <- sum(ifelse(idades > 90, 1, 0)) # Ocorrëncias

    criticas[2, 3] <- paste0(round(criticas[2, 2]/nrow(base_atual) * 100, 2), "%") #
Porcentagem sobre a base


    # 3. Benefício menor que o salário mínimo

    criticas[3, 2] <- sum(ifelse(base_atual$`Benefício Complementar` < sal_minimo, 1, 0)) #
Ocorrëncias

    criticas[3, 3] <- paste0(round(criticas[3, 2]/nrow(base_atual) * 100, 2), "%") #
Porcentagem sobre a base


    # 4. Data de Início de Benefício menor que a Data de Nascimento

    criticas[4, 2] <- sum(ifelse(base_atual$`Data de Início de Benefício` < base_atual$`Data
de Nascimento`, 1, 0)) # Ocorrëncias

    criticas[4, 3] <- paste0(round(criticas[4, 2]/nrow(base_atual) * 100, 2), "%") #
Porcentagem sobre a base


    # 5. Data de Nascimento divergente

    atual <- vlookup(base_atual_x_anterior$`Matrícula`, base_atual, "Matrícula", "Data de
Nascimento")

    anterior <- vlookup(base_atual_x_anterior$`Matrícula`, base_anterior, "Matrícula", "Data
de Nascimento")

    divergencias_atual_anterior$`Data de Nascimento divergente` <- ifelse(atual != anterior,
1, 0)

    criticas[5, 2] <- sum(divergencias_atual_anterior$`Data de Nascimento divergente`) #
Ocorrëncias

    criticas[5, 3] <- paste0(round(criticas[5, 2]/nrow(base_atual) * 100, 2), "%") #
Porcentagem sobre a base


    # 6. Sexo divergente

    atual <- vlookup(base_atual_x_anterior$`Matrícula`, base_atual, "Matrícula", "Sexo")
```

```
    anterior <- vlookup(base_atual_x_anterior$`Matrícula`, base_anterior, "Matrícula",
"Sexo")

    divergencias_atual_anterior$`Sexo divergente` <- ifelse(atual != anterior, 1, 0)

    criticas[6, 2] <- sum(ifelse(atual != anterior, 1, 0)) # Ocorrëncias

    criticas[6, 3] <- paste0(round(criticas[6, 2]/nrow(base_atual) * 100, 2), "%") #
Porcentagem sobre a base


    # 7. Tipo de Benefício divergente

    atual <- vlookup(base_atual_x_anterior$`Matrícula`, base_atual, "Matrícula", "Tipo do
Benefício")

    anterior <- vlookup(base_atual_x_anterior$`Matrícula`, base_anterior, "Matrícula", "Tipo
do Benefício")

    divergencias_atual_anterior$`Tipo de Benefício divergente` <- ifelse(atual != anterior, 1,
0)

    criticas[7, 2] <- sum(ifelse(atual != anterior, 1, 0)) # Ocorrëncias

    criticas[7, 3] <- paste0(round(criticas[7, 2]/nrow(base_atual) * 100, 2), "%") #
Porcentagem sobre a base


    # 8. Benefício complementar com divergências significativas

    atual <- vlookup(base_atual_x_anterior$`Matrícula`, base_atual, "Matrícula", "Benefício
Complementar")

    anterior <- vlookup(base_atual_x_anterior$`Matrícula`, base_anterior, "Matrícula",
"Benefício Complementar")

    acumulado = rv$current_params$acumulado

    limite_superior <- acumulado + 0.5 / 100  # acumulado + 0.5%

    limite_inferior <- acumulado + 0.5 / 100  # acumulado - 0.5%

    variacao <- anterior/ atual - 1

    print(limite_superior)

    print(limite_inferior)

    print(variacao)

    divergencias_atual_anterior$`Benefício Complementar com variações significativas` <-
ifelse(abs(variacao) > limite_superior, 1, 0)

    criticas[8, 2] <- sum(divergencias_atual_anterior$`Benefício Complementar com
variações significativas`) # Ocorrëncias

    criticas[8, 3] <- paste0(round(criticas[8, 2]/nrow(base_atual) * 100, 2), "%") #
Porcentagem sobre a base
```

```
# # FORMATAÇÃO DATAS (P1)

base_atual <- base_atual %>%

  mutate(across(c("Data de Nascimento", "Data de Início de Benefício",

                  "Data de Nascimento Dep. 1", "Data de Nascimento Dep. 2", "Data de Nascimento Dep. 3"),

                ~ format(as.Date(.), "%d/%m/%Y")))


base_anterior <- base_anterior %>%

  mutate(across(c("Data de Nascimento", "Data de Início de Benefício",

                  "Data de Nascimento Dep. 1", "Data de Nascimento Dep. 2", "Data de Nascimento Dep. 3"),

                ~ format(as.Date(.), "%d/%m/%Y")))

# Anexo


# Código 1

df1_anexoI_titulo <- "Código 1\nMatrículas Repetidas"

df1_anexoI <- data.frame(Matrícula = (divergencias_atual %>% filter(`Matrículas Repetidas` == 1))$Matrícula)

df1_anexoI$`Data de Nascimento` <- vlookup(df1_anexoI$`Matrícula`, base_atual, "Matrícula", "Data de Nascimento")


# Código 2

df2_anexoI_titulo <- "Código 2\nIdade maior que 90 anos"

df2_anexoI <- data.frame(Matrícula = (divergencias_atual %>% filter(`Idade maior que 90 anos` == 1))$Matrícula)

df2_anexoI$`Data de Nascimento` <- vlookup(df2_anexoI$`Matrícula`, base_atual, "Matrícula", "Data de Nascimento")

df2_anexoI$`Idade` <- as.numeric(difftime(data_atual, as.Date(df2_anexoI$`Data de Nascimento`, format = "%d/%m/%Y"), units = "weeks")) %/% 52


# Código 3

df3_anexoI_titulo <- "Código 3\nBenefício menor que o salário mínimo"

df3_anexoI <- data.frame(Matrícula = (divergencias_atual %>% filter(`Benefício menor que o salário mínimo` == 1))$Matrícula)
```

```
df3_anexoI$`Benefício Complementar` <- vlookup(df3_anexoI$`Matrícula`, base_atual, "Matrícula", "Benefício Complementar")

df3_anexoI$`Salário mínimo` <- sal_minimo


# Código 4

df4_anexoI_titulo <- "Código 4\nData de Início de Benefício menor que a Data de Nascimento"

df4_anexoI <- data.frame(Matrícula = (divergencias_atual %>% filter(`Data de Início de Benefício menor que a Data de Nascimento` == 1))$Matrícula)

df4_anexoI$`Data de Início de Benefício` <- vlookup(df4_anexoI$`Matrícula`, base_atual, "Matrícula", "Data de Início de Benefício")

df4_anexoI$`Data de Nascimento` <- vlookup(df4_anexoI$`Matrícula`, base_atual, "Matrícula", "Data de Nascimento")


# Código 5

df5_anexoI_titulo <- "Código 5\nData de Nascimento divergente"

df5_anexoI <- data.frame(Matrícula = (divergencias_atual_anterior %>% filter(`Data de Nascimento divergente` == 1))$Matrícula)

df5_anexoI$`Data de Nascimento Anterior` <- vlookup(df5_anexoI$`Matrícula`, base_anterior, "Matrícula", "Data de Nascimento")

df5_anexoI$`Data de Nascimento Atual` <- vlookup(df5_anexoI$`Matrícula`, base_atual, "Matrícula", "Data de Nascimento")


# Código 6

df6_anexoI_titulo <- "Código 6\nSexo divergente"

df6_anexoI <- data.frame(Matrícula = (divergencias_atual_anterior %>% filter(`Sexo divergente` == 1))$Matrícula)

df6_anexoI$`Sexo Anterior` <- vlookup(df6_anexoI$`Matrícula`, base_anterior, "Matrícula", "Sexo")

df6_anexoI$`Sexo Atual` <- vlookup(df6_anexoI$`Matrícula`, base_atual, "Matrícula", "Sexo")


# Código 7

df7_anexoI_titulo <- "Código 7\nTipo de Benefício divergente"

df7_anexoI <- data.frame(Matrícula = (divergencias_atual_anterior %>% filter(`Tipo de Benefício divergente` == 1))$Matrícula)
```

```
df7_anexoI$`Tipo de Benefício Anterior` <- vlookup(df7_anexoI$`Matrícula`, base_anterior, "Matrícula", "Tipo do Benefício")

df7_anexoI$`Tipo de Benefício Atual` <- vlookup(df7_anexoI$`Matrícula`, base_atual, "Matrícula", "Tipo do Benefício")


# Código 8

df8_anexoI_titulo <- "Código 8\nBenefício Complementar com variações significativas"

df8_anexoI <- data.frame(Matrícula = (divergencias_atual_anterior %>% filter(`Benefício Complementar com variações significativas` == 1))$Matrícula)

df8_anexoI$`Benefício Anterior` <- vlookup(df8_anexoI$`Matrícula`, base_anterior, "Matrícula", "Benefício Complementar")

df8_anexoI$`Benefício Atual` <- vlookup(df8_anexoI$`Matrícula`, base_atual, "Matrícula", "Benefício Complementar")

df8_anexoI$`Variação` <- (df8_anexoI$`Benefício Atual` - df8_anexoI$`Benefício Anterior`) / df8_anexoI$`Benefício Anterior`

df8_anexoI$`Variação Esperada` <- acumulado


# Auxílio para loop ao preencher excel

dfs_anexoI_títulos <- c(df1_anexoI_titulo, df2_anexoI_titulo, df3_anexoI_titulo, df4_anexoI_titulo, df5_anexoI_titulo, df6_anexoI_titulo, df7_anexoI_titulo, df8_anexoI_titulo)

dfs_anexoI <- list(df1_anexoI, df2_anexoI, df3_anexoI, df4_anexoI, df5_anexoI, df6_anexoI, df7_anexoI, df8_anexoI)


# Movimentação

entradas <- dplyr::anti_join(base_atual, base_anterior, by = "Matrícula") # Quem está na base atual e não está na base anterior

entradas <- entradas %>% select("Matrícula", "Sexo", "Data de Nascimento", "Benefício Complementar")

saídas <- dplyr::anti_join(base_anterior, base_atual, by = "Matrícula") #  Quem está na base anterior e não está na base atual

saídas <- saídas %>% select("Matrícula", "Sexo", "Data de Nascimento", "Benefício Complementar")


# Estatísticas

estatísticas <- data.frame(

  Descrição = c(
```

```
      "1. Participantes Aposentados",
      "1.2. Homens",
      "1.3. Mulheres",
      "2. Idade Média",
      "3. Número médio de dependentes",
      "4. Benefício Complementar Total",
      "5. Benefício Complementar Médio"
    ),
    Anterior = NA,
    Atual = NA
  )
  colnames(estatísticas) <- c("Descrição", "Base Anterior", "Base Atual")


  # 1. Participantes Aposentados
  estatísticas[1, 2] <- nrow(base_anterior)
  estatísticas[1, 3] <- nrow(base_atual)
  # 1.2. Homens
  estatísticas[2, 2] <- sum(ifelse(base_anterior$`Sexo` == "M", 1, 0))
  estatísticas[2, 3] <- sum(ifelse(base_atual$`Sexo` == "M", 1, 0))
  # 1.3. Mulheres
  estatísticas[3, 2] <- sum(ifelse(base_anterior$`Sexo` == "F", 1, 0))
  estatísticas[3, 3] <- sum(ifelse(base_atual$`Sexo` == "F", 1, 0))
  # 2. Idade Média
  data_anterior <- as.Date(input$data_inicio, origin = "1899-12-30") # mesma base de data
  data_atual <- as.Date(input$data_fim, origin = "1899-12-30") # mesma base de data
  datas_nascimento_anterior <- as.Date(base_anterior$`Data de Nascimento`, format =
"%d/%m/%Y") # para o sistema de data do Excel (base 1900)
  datas_nascimento_atual <- as.Date(base_atual$`Data de Nascimento`, format =
"%d/%m/%Y") # para o sistema de data do Excel (base 1900)
  estatísticas[4, 2] <- mean(as.numeric(difftime(data_anterior, datas_nascimento_anterior,
units = "weeks")) %/% 52)
```

```
    estatísticas[4, 3] <- mean(as.numeric(difftime(data_atual, datas_nascimento_atual, units = "weeks")) %/% 52)

    # 3. Número médio de dependentes

    estatísticas[5, 2] <- sum(rowSums(!is.na(base_anterior[, c("Sexo Dep. 1", "Sexo Dep. 2", "Sexo Dep. 3")])))/nrow(base_anterior)

    estatísticas[5, 3] <- sum(rowSums(!is.na(base_atual[, c("Sexo Dep. 1", "Sexo Dep. 2", "Sexo Dep. 3")])))/nrow(base_atual)

    # 4. Benefício Complementar Total

    estatísticas[6, 2] <- sum(base_anterior$`Benefício Complementar`)

    estatísticas[6, 3] <- sum(base_atual$`Benefício Complementar`)

    # 5. Benefício Complementar Médio

    estatísticas[7, 2] <- mean(base_anterior$`Benefício Complementar`)

    estatísticas[7, 3] <- mean(base_atual$`Benefício Complementar`)


    # Criando pasta excel

    wb <- createWorkbook()


    style_corpo_bordas <- createStyle(halign = "center", fontSize = 10, border = "TopBottomLeftRight", borderColour = "#A6A6A6", fontColour = "#262626")

    style_corpo_bordas_left <- createStyle(halign = "left", fontSize = 10, border = "TopBottomLeftRight", borderColour = "#A6A6A6", fontColour = "#262626")

    style_corpo_bordas_amarelo <- createStyle(halign = "left", fontSize = 10, fgFill = "#ffff66", border = "TopBottomLeftRight", borderColour = "#A6A6A6", fontColour = "#262626")

    style_cabecalho2 <- createStyle(fontColour = "#262626", fgFill = "#D9D9D9", border = "TopBottomLeftRight", borderColour = "#A6A6A6",fontSize = 10, halign = "center", valign = "center", textDecoration = "bold", wrapText = TRUE)

    style_cabecalho3 <- createStyle(fontColour = "#262626", fgFill = "#ffde21", border = "TopBottomLeftRight",fontSize = 10, borderColour = "#A6A6A6", halign = "center", valign = "center", textDecoration = "bold", wrapText = TRUE)

    style_cabecalho4 <- createStyle(fontColour = "#f2eded", fgFill = "#474747", border = "TopBottomLeftRight", borderColour = "#A6A6A6",fontSize = 11, halign = "center", valign = "center", textDecoration = "bold", wrapText = TRUE)

    style_cabecalho5 <- createStyle(fontColour = "#f2eded", fgFill = "#474747", border = "TopBottomLeftRight", borderColour = "#f2eded",fontSize = 12, halign = "center", valign = "center", textDecoration = "bold", wrapText = TRUE)
```

```
    style_cabecalho <- createStyle(fontColour = "#262626", fgFill = "#d3d7de", border =
"TopBottomLeftRight",fontSize = 10, halign = "center", valign = "center", textDecoration =
"bold", wrapText = TRUE)



    # Aba de Parâmetros
    addWorksheet(wb, "Parâmetros")
    writeData(
      wb,
      "Parâmetros",
      data.frame(
        Parâmetro = c("Indexador", "Data Início", "Data Fim", "Acumulado (%)"),
        Valor = c(
          rv$current_params$indexador,
          rv$current_params$data_inicio,
          rv$current_params$data_fim,
          sprintf("%.2f%%", rv$current_params$acumulado * 100)
        )
      )
      , startCol = 2, startRow = 2)

    writeData(wb, "Parâmetros", "Base Anterior", startCol = 5, startRow = 2)
    writeData(wb, "Parâmetros", "Base Atual", startCol = 5, startRow = 3)
    writeData(wb, "Parâmetros", input$base1$name, startCol = 6, startRow = 2)
    writeData(wb, "Parâmetros", input$base2$name, startCol = 6, startRow = 3)

    writeData(wb, "Parâmetros", "Path Anterior", startCol = 5, startRow = 5)
    writeData(wb, "Parâmetros", "Path Atual", startCol = 5, startRow = 6)
    writeData(wb, "Parâmetros", input$base1$datapath, startCol = 6, startRow = 5)
    writeData(wb, "Parâmetros", input$base2$datapath, startCol = 6, startRow = 6)

    addStyle(wb, "Parâmetros", style_cabecalho2, rows = 2, cols = 2:3, gridExpand = TRUE)
```

```
    addStyle(wb, "Parâmetros", style_corpo_bordas_left, rows = 3:6, cols = 2, gridExpand =
TRUE)

    addStyle(wb, "Parâmetros", style_corpo_bordas, rows = 3:6, cols = 3, gridExpand =
TRUE)

    addStyle(wb, "Parâmetros", style_cabecalho2, rows = 2, cols = 5, gridExpand = TRUE)

    addStyle(wb, "Parâmetros", style_cabecalho2, rows = 3, cols = 5, gridExpand = TRUE)

    addStyle(wb, "Parâmetros", style_cabecalho2, rows = 5, cols = 5, gridExpand = TRUE)

    addStyle(wb, "Parâmetros", style_cabecalho2, rows = 6, cols = 5, gridExpand = TRUE)

    addStyle(wb, "Parâmetros", style_corpo_bordas_amarelo, rows = 2, cols = 6, gridExpand
= TRUE)

    addStyle(wb, "Parâmetros", style_corpo_bordas_amarelo, rows = 3, cols = 6, gridExpand
= TRUE)

    addStyle(wb, "Parâmetros", style_corpo_bordas_amarelo, rows = 5, cols = 6, gridExpand
= TRUE)

    addStyle(wb, "Parâmetros", style_corpo_bordas_amarelo, rows = 6, cols = 6, gridExpand
= TRUE)


    setColWidths(wb, "Parâmetros", cols = 1, widths = 3)

    setColWidths(wb, "Parâmetros", cols = 2, widths = 15)

    setColWidths(wb, "Parâmetros", cols = 3, widths = 11)

    setColWidths(wb, "Parâmetros", cols = 4, widths = 3)

    setColWidths(wb, "Parâmetros", cols = 6, widths = 130)


    setRowHeights(wb, "Parâmetros", rows = 2, heights = 14.4)


    # Aba Base 1

    if (!is.null(base_anterior)) {

      addWorksheet(wb, "Base 1")

      writeData(wb, "Base 1", base_anterior, startCol = 2, startRow = 2)

      setColWidths(wb, "Base 1", cols = 1, widths = 2.5)

      setColWidths(wb, "Base 1", cols = 2:ncol(base_anterior), widths = "auto")


      addStyle(wb, sheet = 2, style_cabecalho2, rows = 2, cols = 2:(1+ncol(base_anterior)))

      addStyle(wb, sheet = 2, style_corpo_bordas, rows = 3:(2+nrow(base_anterior)), cols =
2:(1+ncol(base_anterior)), gridExpand = TRUE)
```

```
}

# Aba Base 2
if (!is.null(base_atual)) {
  addWorksheet(wb, "Base 2")
  writeData(wb, "Base 2", base_atual, startCol = 2, startRow = 2)
  setColWidths(wb, "Base 2", cols = 1, widths = 2.5)
  setColWidths(wb, "Base 2", cols = 2:ncol(base_atual), widths = "auto")

  addStyle(wb, sheet = "Base 2", style_cabecalho2, rows = 2, cols =
2:(1+ncol(base_atual)))
  addStyle(wb, sheet = "Base 2", style_corpo_bordas, rows = 3:(2+nrow(base_atual)), cols
= 2:(1+ncol(base_atual)), gridExpand = TRUE)
}

# Aba de Parâmetros
addWorksheet(wb, "Movimentação")
writeData(wb, "Movimentação", paste0("Entradas", " (", nrow(entradas),")"), startCol = 2,
startRow = 2)
writeData(wb, "Movimentação", entradas, startCol = 2, startRow = 3)
writeData(wb, "Movimentação", paste0("Saídas", " (", nrow(saídas),")"), startCol = 7,
startRow = 2)
writeData(wb, "Movimentação", saídas, startCol = 7, startRow = 3)
writeData(wb, "Movimentação", "Estatísticas", startCol = 12, startRow = 2)
writeData(wb, "Movimentação", estatísticas, startCol = 12, startRow = 3)

addStyle(wb, sheet = "Movimentação", style_cabecalho4, rows = 2, cols = 2:5,
gridExpand = TRUE)
addStyle(wb, sheet = "Movimentação", style_cabecalho2, rows = 3, cols = 2:5,
gridExpand = TRUE)
addStyle(wb, sheet = "Movimentação", style_corpo_bordas, rows = 4:(3+nrow(entradas)),
cols = 2:(1+ncol(entradas)), gridExpand = TRUE)
```

```
    addStyle(wb, sheet = "Movimentação", style_cabecalho4, rows = 2, cols = 7:10,
gridExpand = TRUE)

    addStyle(wb, sheet = "Movimentação", style_cabecalho2, rows = 3, cols = 7:10,
gridExpand = TRUE)

    addStyle(wb, sheet = "Movimentação", style_corpo_bordas, rows = 4:(3+nrow(saídas)),
cols = 7:(6+ncol(saídas)), gridExpand = TRUE)


    addStyle(wb, sheet = "Movimentação", style_cabecalho4, rows = 2, cols = 12:14,
gridExpand = TRUE)

    addStyle(wb, sheet = "Movimentação", style_cabecalho2, rows = 3, cols = 12:14,
gridExpand = TRUE)

    addStyle(wb, sheet = "Movimentação", style_corpo_bordas, rows =
4:(3+nrow(estatísticas)), cols = 12:(11+ncol(estatísticas)), gridExpand = TRUE)


    mergeCells(wb, sheet = "Movimentação", cols =  2:(1+ncol(entradas)), rows = 2)

    mergeCells(wb, sheet = "Movimentação", cols =  7:(6+ncol(saídas)), rows = 2)

    mergeCells(wb, sheet = "Movimentação", cols =  12:(11+ncol(estatísticas)), rows = 2)


    setColWidths(wb, sheet = "Movimentação", cols = c(1, 6, 11), widths = 2)

    setColWidths(wb, sheet = "Movimentação", cols = c(2), widths = 9)

    setColWidths(wb, sheet = "Movimentação", cols = c(3), widths = 6)

    setColWidths(wb, sheet = "Movimentação", cols = c(4), widths = 24)

    setColWidths(wb, sheet = "Movimentação", cols = c(5), widths = 23)

    setColWidths(wb, sheet = "Movimentação", cols = c(7), widths = 9)

    setColWidths(wb, sheet = "Movimentação", cols = c(8), widths = 6)

    setColWidths(wb, sheet = "Movimentação", cols = c(9), widths = 24)

    setColWidths(wb, sheet = "Movimentação", cols = c(10), widths = 23)

    setColWidths(wb, sheet = "Movimentação", cols = c(12), widths = 28)


    # Aba Divergências

    addWorksheet(wb, "Divergências")

    writeData(wb, "Divergências", "Resumo - Crítica da Base de Dados", startCol = 2,
startRow = 2)

    writeData(wb, "Divergências", criticas, startCol = 2, startRow = 3)
```

```
    mergeCells(wb, sheet = "Divergências", cols = 2:(1+ncol(criticas)), rows = 2)


    setColWidths(wb, sheet = "Divergências", cols = 2, widths = 57)
    setColWidths(wb, sheet = "Divergências", cols = 3, widths = 11)
    setColWidths(wb, sheet = "Divergências", cols = 4, widths = 21)


    addStyle(wb, sheet = "Divergências", style_cabecalho4, rows = 2, cols =
2:(1+ncol(criticas)), gridExpand = TRUE)
    addStyle(wb, sheet = "Divergências", style_cabecalho2, rows = 3, cols =
2:(1+ncol(criticas)), gridExpand = TRUE)
    addStyle(wb, sheet = "Divergências", style_corpo_bordas_left, rows =
4:(3+nrow(criticas)), cols = 2, gridExpand = TRUE)
    addStyle(wb, sheet = "Divergências", style_corpo_bordas, rows = 4:(3+nrow(criticas)),
cols = 3:(1+ncol(criticas)), gridExpand = TRUE)
    # Aba Anexo
    addWorksheet(wb, "Anexo")


    starting_col <- 2
    for (i in 1:length(dfs_anexoI_títulos)){
      if (nrow(dfs_anexoI[[i]]) > 0 ){
        writeData(wb, sheet = "Anexo", x = dfs_anexoI_títulos[i], startCol = starting_col,
startRow = 2)
        writeData(wb, sheet = "Anexo", x = dfs_anexoI[[i]], startCol = starting_col, startRow =
3)


        mergeCells(wb, sheet = "Anexo", cols =
starting_col:(starting_col+length(dfs_anexoI[[i]])-1), rows = 2) # Unindo colunas do título


        setColWidths(wb, sheet = "Anexo", cols = starting_col, widths = 14) # Largura
primeira coluna (Matricula)
        setColWidths(wb, sheet = "Anexo", cols =
(starting_col+1):(starting_col+length(dfs_anexoI[[i]])-1), widths = 22) # Largura resto dos
subtítulos


        addStyle(wb, sheet = "Anexo", style_cabecalho4, rows = 2, cols =
starting_col:(starting_col+length(dfs_anexoI[[i]])-1))
```

```
        addStyle(wb, sheet = "Anexo", style_cabecalho2, rows = 3, cols =
starting_col:(starting_col+length(dfs_anexoI[[i]])-1))
        addStyle(wb, sheet = "Anexo", style_corpo_bordas, rows =
4:(3+nrow(dfs_anexoI[[i]])), cols = starting_col:(starting_col+length(dfs_anexoI[[i]])-1),
gridExpand = TRUE)


        setColWidths(wb, sheet = "Anexo", cols = starting_col + length(dfs_anexoI[[i]]) + 1,
widths = 1)
        starting_col <- starting_col + length(dfs_anexoI[[i]]) + 1
      }
    }


    setColWidths(wb, sheet = "Anexo", cols = 1, widths = 1)
    setRowHeights(wb, sheet = "Anexo", rows = 1, heights = 9)
    setRowHeights(wb, sheet = "Anexo", rows = 2, heights = 44)


    # Salvar o workbook
    saveWorkbook(wb, file, overwrite = TRUE)
    rv$criticas <- criticas
    rv$entradas <- entradas
    rv$saídas <- saídas
  }
)


 output$divergencias <- renderDT({
  req(rv$criticas)


  datatable(
    rv$criticas,
    options = list(
      pageLength = 8,
      dom = 'rtip',
      language = list(
```

```
      info = "Mostrando _START_ até _END_ de _TOTAL_ registros"
    )
   ),
   class = 'cell-border stripe hover',
   rownames = FALSE
  )
})

output$movimentacao <- renderDT({
 req(rv$entradas, rv$saídas)

 # Cria um dataframe com duas colunas
 resumo <- data.frame(
  Entradas = nrow(rv$entradas),
  Saídas = nrow(rv$saídas)
 )

 # Exibe o resumo como uma tabela com estilo aprimorado
 datatable(
  resumo,
  options = list(
   dom = 'rt',
   ordering = FALSE,
   language = list(
    info = ""
   ),
   columnDefs = list(
    list(className = 'dt-center', targets = '_all')
   ),
   initComplete = JS(
    "function(settings, json) {
    $(this.api().table().container()).css({
```

```
        'background': 'linear-gradient(to right, #f8f9fa, #e9ecef)',
        'border-radius': '8px',
        'box-shadow': '0 4px 6px rgba(0, 0, 0, 0.1)',
        'padding': '20px'
      });
      $(this.api().table().header()).css({
        'background-color': '#4a5568',
        'color': 'white',
        'font-size': '16px'
      });
      $(this.api().table().body()).find('td').css({
        'font-size': '22px',
        'padding': '15px',
        'font-weight': 'bold'
      });
    }"
    )
  ),
  class = 'display',
  rownames = FALSE,
  caption = htmltools::tags$caption(
    style = 'caption-side: top; text-align: center; font-weight: bold; font-size: 24px; margin-bottom: 15px; color: #2c3e50;',
    'Resumo de Movimentação'
  )
) %>%
  formatStyle(
    columns = c('Entradas', 'Saídas'),
    backgroundColor = styleEqual(
      c(0), c('#f5f5f5')
    ),
    color = styleEqual(
```

```
        c(0), c('#6c757d')
      ),
      fontWeight = 'bold',
      border = '1px solid #dee2e6',
      borderRadius = '4px'
    )
  })

}

# Run the application
shinyApp(ui = ui, server = server)
```