

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE CIÊNCIA DA COMPUTAÇÃO

FELIPE FERNANDES BASTOS

**MamInAgro - Alternativas para cálculos
sobre as observações das espécies da fauna
brasileira**

Monografia apresentada como requisito parcial
para a obtenção do grau de Bacharel em Ciência
da Computação

Orientador: Prof^a. Dr^a. Renata Galante
Co-orientador: Prof^o. Dr^a. Pedro Cordeiro Estrela
de Andrade Pinto

Porto Alegre
2024

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. Carlos André Bulhões Mendes

Vice-Reitora: Prof^ª. Patricia Helena Lucas Pranke

Pró-Reitor de Graduação: Prof^ª. Cíntia Inês Boll

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

Coordenador do Curso de Ciência de Computação: Prof. Marcelo Walter

Bibliotecário-chefe do Instituto de Informática: Alexsander Borges Ribeiro

AGRADECIMENTOS

Meu primeiro agradecimento é para a minha orientadora, Renata Galante, que me ajudou muito no trabalho, seja com novas ideias ou apoiando as ideias que eu tive ao longo do projeto. Mesmo em um momento de dificuldade, como as enchentes que enfrentamos, ela sempre esteve à disposição para me ajudar. Quero a todos familiares que estiveram do meu lado nessa caminhada, por todo apoio. Meu pai, Rubens, que sempre fez o possível e o impossível para que eu me dedicasse estritamente aos estudos, e minha mãe, Giane, que sempre me ajudou nos momentos de maior tensão na jornada acadêmica. Quero agradecer à minha namorada, Maria Eduarda, por sempre me apoiar, especialmente nos momentos mais estressantes da faculdade. Um agradecimento especial à minha dinda Isabel, que cuidou de mim e da minha educação durante todo meu período de colégio. Queria agradecer também ao meu primo Eduardo, que em 2015 me deu o incentivo que faltava para eu mudar de curso e ingressar em Ciência da Computação na UFRGS. E um muito obrigado a todos que estiveram comigo e me acompanharam nessa trajetória!

RESUMO

Em um contexto onde a agricultura intensiva e o uso de agrotóxicos ameaçam cada vez mais a biodiversidade, especialmente em paisagens agrícolas, torna-se fundamental entender e mitigar esses impactos. Este trabalho foca na evolução do MamInAgro, uma aplicação desenvolvida para reunir e analisar dados sobre mamíferos em áreas agrícolas no Brasil. Para a evolução dos cálculos foram consideradas três alternativas: (1) transformar os cálculos de R para Python e utilizá-los diretamente em um serviço de backend; (2) rodar os scripts de cálculos em Python utilizando bibliotecas auxiliares para chamar scripts R; e (3) alterar diretamente os cálculos no código R, adaptando-os para que os resultados fossem armazenados em tabelas no banco de dados. Após a análise das alternativas, foram feitas melhorias significativas na aplicação, incluindo a evolução dos cálculos, o armazenamento adequado dos resultados obtidos, a implementação de novos métodos de análise e a criação de um sistema de visualização de dados mais robusto. Essas mudanças não só aprimoram a capacidade da aplicação de processar dados complexos, mas também tornam as informações mais acessíveis para visualização e tomada de decisões. Com isso, o MamInAgro avança para se tornar uma ferramenta útil para pesquisadores que buscam entender melhor as interações entre a agricultura e a vida selvagem, contribuindo para a preservação da biodiversidade.

Palavras-chave: Banco de dados. visualização de dados. agroecossistema. análise de dados.

ABSTRACT

In a context where intensive agriculture and the use of pesticides increasingly threaten biodiversity, especially in agricultural landscapes, understanding and mitigating these impacts becomes crucial. This work focuses on the evolution of MamInAgro, an application developed to gather and analyze data on mammals in Brazilian agricultural areas. Three alternatives were considered for implementing the calculations: (1) transforming the calculations from R to Python and using them directly in a backend service; (2) running the calculation scripts in Python using auxiliary libraries to call R scripts; and (3) directly modifying the calculations in the R code, adapting them to store the results in database tables. After analyzing these alternatives, significant improvements were made to the application, including enhancements to the calculations, proper storage of the obtained results, implementation of new analysis methods, and the creation of a more robust data visualization system. These changes not only improve the application's ability to process complex data but also make the information more accessible for visualization and decision-making. As a result, MamInAgro advances as a valuable tool for researchers seeking to better understand the interactions between agriculture and wildlife, contributing to biodiversity conservation.

Keywords: Database, Data Visualization, Agroecosystem, Data Analysis.

LISTA DE FIGURAS

Figura 2.1	SiBBr	15
Figura 2.2	GBIF	17
Figura 2.3	NatureServe	19
Figura 2.4	Atlas of Living Australia	21
Figura 3.1	MamInAgro	24
Figura 4.1	Python - Exemplo Visualização	39
Figura 4.2	Metabase - Exemplo Visualização	40
Figura 4.3	Banco - Tabelas Cálculos	42
Figura 4.4	Banco - VWM's Cálculos	44
Figura 4.5	Backoffice - Login	47
Figura 4.6	Backoffice - Home	47
Figura 4.7	Metabase - Tela Inicial	49
Figura 4.8	Metabase - SQL Editor	50
Figura 4.9	Metabase - Dashboard 1	51
Figura 4.10	Metabase - Dashboard 2	51
Figura 4.11	Metabase - Sem Filtro	52
Figura 4.12	Metabase - Com Filtro	53
Figura 5.1	Área de Formação dos Participantes	55
Figura 5.2	Nível de Escolaridade dos Participantes	56
Figura 5.3	Identidade de Gênero dos Participantes	56
Figura 5.4	Nível de Experiência com Internet dos Participantes	57
Figura 5.5	Facilidade de Entendimento das Visualizações	58
Figura 5.6	Velocidade de Resposta ao Realizar Consultas	58
Figura 5.7	Facilidade de Criação e Customização de Filtros	59
Figura 5.8	Intuitividade da Interface	59
Figura 5.9	Satisfação Geral com o Metabase	60

LISTA DE TABELAS

Tabela 2.1 Comparação das Funcionalidades de Visualização de Dados	22
--	----

LISTA DE ABREVIATURAS E SIGLAS

ALA	<i>Atlas of Living Australia</i>
AWS	<i>Amazon Web Services</i>
API	<i>Application Programming Interface</i>
BI	<i>Business Intelligence</i>
GBIF	<i>Global Biodiversity Information Facility</i>
JSON	<i>JavaScript Object Notation</i>
JWT	<i>JSON Web Token</i>
MFA	<i>Multi-Factor Authentication</i>
R	<i>Linguagem de Programação R</i>
SiBBR	<i>Sistema de Informação sobre a Biodiversidade Brasileira</i>
SQL	<i>Structured Query Language</i>
VWM	<i>View Materializada</i>

SUMÁRIO

1 INTRODUÇÃO	11
2 TRABALHOS RELACIONADOS	13
2.1 Sistema de Informação sobre a Biodiversidade Brasileira	13
2.1.1 Funcionalidades Principais	13
2.1.2 Impacto e Utilização	14
2.1.3 Comparação com MamInAgro	14
2.1.4 Visualização de Dados no SiBBR	14
2.2 Global Biodiversity Information Facility (GBIF)	15
2.2.1 Funcionalidades Principais	15
2.2.2 Impacto e Utilização	16
2.2.3 Comparação com MamInAgro	16
2.2.4 Visualização de Dados no GBIF	16
2.3 NatureServe	17
2.3.1 Funcionalidades Principais	17
2.3.2 Impacto e Utilização	18
2.3.3 Comparação com MamInAgro	18
2.3.4 Visualização de Dados no NatureServe	18
2.4 Atlas of Living Australia (ALA)	19
2.4.1 Funcionalidades Principais	20
2.4.2 Impacto e Utilização	20
2.4.3 Comparação com MamInAgro	20
2.4.4 Visualização de Dados no ALA	20
2.5 Comparação entre os Trabalhos Relacionados	21
3 MAMINAGRO E REQUISITOS DA APLICAÇÃO	23
3.1 MamInAgro	23
3.2 Limitações e Novas Funcionalidades	24
3.3 Requisitos do Trabalho	25
3.4 Evolução dos Requisitos	26
4 APLICAÇÃO	27
4.1 Métodos Levantados para a Alteração dos Cálculos em R	27
4.1.1 Utilizando rpy2 para Chamada de Cálculos em R via Python	29
4.1.1.1 Integração Python-R com rpy2	29
4.1.1.2 Vantagens e Desvantagens da Integração Python-R	32
4.1.2 Mudança dos Cálculos Diretamente em R e Armazenamento dos Resultados no Banco de Dados	32
4.1.2.1 Execução de Cálculos em R	32
4.1.2.2 Vantagens e Desvantagens de Realizar os Cálculos Diretamente em R	34
4.1.3 Tradução do Código R para Python	35
4.1.3.1 Tradução de Códigos R para Python	35
4.1.3.2 Vantagens e Desvantagens da Tradução para Python	37
4.2 Escolha da Atualização dos Scripts de R	38
4.3 Visualização de Dados	38
4.3.1 Visualização de Dados em Python	38
4.3.1.1 Vantagens e Desvantagens da Visualização em Python	39
4.3.2 Visualização de Dados com Ferramentas de BI	40
4.3.2.1 Vantagens e Desvantagens do Metabase	41
4.3.3 Escolha do Metabase como Ferramenta de Visualização de Dados	41

4.4 Mudanças no Banco de Dados	42
4.4.1 Criação de Tabelas no esquema calculations	42
4.4.2 Escalabilidade e Uso de Visões Materializadas	43
4.4.3 Criação de Visões Materializadas	43
4.5 Execução dos Cálculos na Aplicação	44
4.5.1 Integração com Flask para Chamada de Cálculos R.....	44
4.6 Backoffice.....	45
4.6.1 Segurança com AWS Cognito.....	46
4.7 Nova Arquitetura para o Sistema de Cálculos	47
4.8 Implementação de Dashboards no Metabase	48
4.8.1 Processo de Criação de Dashboards	49
4.8.2 Perguntas e SQL Editor no Metabase	49
4.8.3 Dashboards para Diferentes Públicos	50
4.8.4 Funcionalidades dos Dashboards.....	52
4.8.5 Fluxo Completo do Sistema de Cálculos	53
5 AVALIAÇÃO DA USABILIDADE	55
5.1 Resultados.....	55
6 CONCLUSÃO	61
REFERÊNCIAS.....	62

1 INTRODUÇÃO

Estamos vivendo um momento decisivo para a preservação da biodiversidade em nosso planeta. O acelerado avanço das mudanças climáticas, a expansão contínua das fronteiras agrícolas, a introdução de espécies invasoras e o crescimento descontrolado das áreas urbanas estão entre os principais fatores que ameaçam a diversidade biológica. Esses elementos, somados às intervenções humanas como a construção de infraestrutura e a exploração de recursos naturais, têm levado muitas espécies ao risco de extinção. Muito se discute sobre as medidas necessárias para mitigar esses impactos e preservar a biodiversidade, mas é fundamental, antes de mais nada, compreender o estado atual da diversidade biológica que ainda possuímos.

No Brasil, o uso intensivo de pesticidas representa um grande desafio para a conservação da biodiversidade. O país é um dos maiores consumidores mundiais desses produtos, e o número de pesticidas aprovados para uso aumentou drasticamente nos últimos anos, expondo organismos não-alvo a resíduos e subprodutos prejudiciais (Fundação Heinrich Böll, 2023). O "Atlas dos Agrotóxicos" destaca que o uso de herbicidas como o glifosato está relacionado à redução de organismos vivos no solo, afetando diretamente a biodiversidade e a saúde dos ecossistemas (Fundação Heinrich Böll, 2023). Além disso, as consequências ambientais do uso desenfreado desses produtos incluem a contaminação de recursos hídricos e a degradação de habitats naturais. (ECODEBATE, 2023). Essa situação reforça a necessidade de revisão das políticas regulatórias e de uma abordagem mais sustentável na agricultura para mitigar os impactos negativos sobre a biodiversidade.

A agricultura, embora essencial para atender à crescente demanda global por alimentos, é amplamente reconhecida como uma das principais forças motrizes na redução da biodiversidade em paisagens rurais. O impacto da expansão agrícola sobre a biodiversidade é evidente, especialmente em áreas onde a intensificação do uso da terra leva à fragmentação dos habitats e à perda de espécies nativas. Estudos indicam que práticas agrícolas modernas, como o uso intensivo de fertilizantes e pesticidas, estão diretamente relacionadas à degradação dos ecossistemas e à diminuição da diversidade biológica nesses ambientes (WILLIAMS, 2023).

Diversas bases de dados, tanto no Brasil quanto internacionalmente, organizam informações sobre a ocorrência de espécies. A aplicação, denominada MamInAgro, trabalho de conclusão de curso da aluna Laurien Santin, é uma aplicação web com um banco de dados que permite o armazenamento do maior número possível de informações fre-

quentemente divulgadas em estudos da área(SANTIN, 2024).

O objetivo deste trabalho é projetar e desenvolver a parte de cálculos da aplicação, dando continuidade ao projeto MamInAgro. Foram consideradas três alternativas para a implementação dos cálculos: (1) transformar os cálculos de R para Python e utilizá-los diretamente em um serviço de backend que fornecesse os resultados para a aplicação web; (2) rodar os scripts de cálculos em Python utilizando bibliotecas auxiliares para chamar scripts R; (3) alterar diretamente os cálculos no código R, adaptando-os para que os resultados fossem armazenados em tabelas no banco de dados, permitindo consultas por outras aplicações. Após a conclusão dessa fase, foi implementada uma nova proposta de visualização de dados que reflete os valores consolidados nos cálculos.

Este trabalho está dividido em cinco capítulos. No Capítulo 2, são apresentadas algumas ferramentas que têm finalidades semelhantes ao MamInAgro e como elas exibem suas informações, incluindo o Sistema de Informação sobre a Biodiversidade Brasileira (SiBBR), o Global Biodiversity Information Facility (GBIF), o NatureServe e o Atlas of Living Australia (ALA). No Capítulo 3, discutem-se o MamInAgro, suas limitações e os requisitos das novas funcionalidades. No Capítulo 4, são abordadas a aplicação, as alternativas estudadas para resolver os problemas dos cálculos, a arquitetura desenvolvida para essa nova etapa e a solução de visualização de dados. O Capítulo 5 apresenta os resultados de uma avaliação realizada com usuários por meio de formulários, e, finalmente, o Capítulo 6 traz as conclusões do trabalho e sugestões para trabalhos antigos.

2 TRABALHOS RELACIONADOS

Esse capítulo apresenta ferramentas com funcionalidades similares ao que já foi desenvolvido no MamInAgro e como são feitas suas visualizações de dados. Primeiro, é apresentado o SiBBr, que é o representante brasileiro da GBIF, agregando todos os dados de ocorrências das espécies brasileiras.

2.1 Sistema de Informação sobre a Biodiversidade Brasileira

O Sistema de Informação sobre a Biodiversidade Brasileira (SiBBr) é uma plataforma online que integra dados sobre biodiversidade e ecossistemas brasileiros (Sistema de Informação sobre a Biodiversidade Brasileira, 2024). Ele oferece acesso a coleções biológicas, catálogos de espécies, ocorrências por região e ferramentas geográficas. Além disso, permite que pesquisadores registrem e publiquem conjuntos de dados e que entusiastas participem por meio de projetos de ciência cidadã. A plataforma suporta a conservação da biodiversidade e a formulação de políticas públicas.

2.1.1 Funcionalidades Principais

O SiBBr apresenta várias funcionalidades que facilitam a coleta e análise de dados sobre a biodiversidade:

- **Coleções Biológicas:** Acesso a uma vasta gama de coleções biológicas de diversas instituições brasileiras.
- **Catálogos de Espécies:** Informação detalhada sobre as espécies registradas, incluindo taxonomia, distribuição geográfica e status de conservação.
- **Ocorrências por Região:** Ferramentas para explorar dados de ocorrências de espécies em diferentes regiões do Brasil.
- **Ferramentas Geográficas:** Integração com mapas e outras ferramentas geoespaciais para visualização e análise dos dados.
- **Registro de Dados:** Interface para que pesquisadores e cidadãos possam registrar e publicar novos conjuntos de dados.
- **Ciência Cidadã:** Incentivo à participação de entusiastas e voluntários através de

projetos de ciência cidadã.

2.1.2 Impacto e Utilização

O SiBBr tem desempenhado um papel crucial na conservação da biodiversidade brasileira. Ao centralizar e disponibilizar dados sobre as espécies e seus habitats, a plataforma facilita a pesquisa científica e a formulação de políticas públicas voltadas à preservação ambiental. Além disso, a participação de cidadãos em projetos de ciência cidadã aumenta a conscientização pública sobre a importância da biodiversidade.

2.1.3 Comparação com MamInAgro

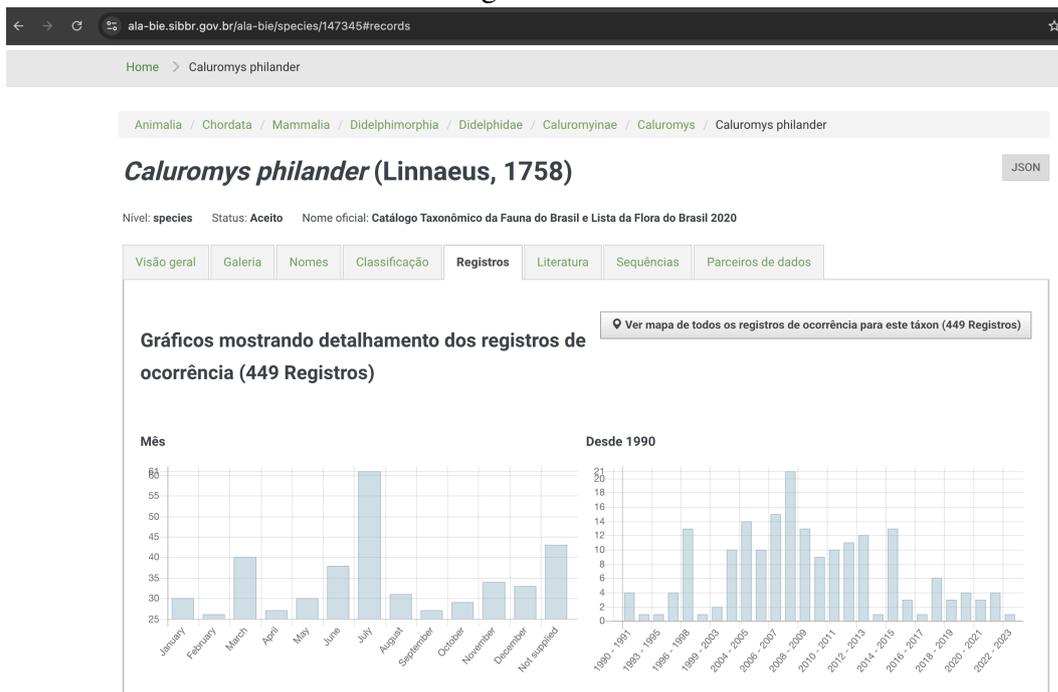
Embora o SiBBr seja uma ferramenta abrangente para a coleta e disseminação de dados sobre biodiversidade, o MamInAgro foca especificamente em mamíferos em paisagens agrícolas. O MamInAgro não apenas agrega dados de ocorrências, mas também caracteriza os ambientes agrícolas onde os animais são encontrados, o que é essencial para estudos sobre o impacto da agricultura e dos agrotóxicos na biodiversidade.

2.1.4 Visualização de Dados no SiBBr

O SiBBr oferece diversas ferramentas de visualização de dados, facilitando a análise e interpretação das informações sobre a biodiversidade. A plataforma possui um **portal espacial**, que integra registros de espécies, áreas e camadas ambientais. Os usuários podem visualizar registros de uma ou mais espécies, explorar relações entre táxons e aplicar filtros diversos, como intervalos de tempo, regiões e grupos taxonômicos.

Além disso, o SiBBr permite a criação de **mapas personalizados**, onde os usuários podem definir áreas de pesquisa, desenhando polígonos ou carregando arquivos em formato shapefile. Isso possibilita uma análise detalhada da distribuição das espécies e suas interações com o ambiente natural. A plataforma também disponibiliza listas de espécies e registros de ocorrências que podem ser refinados e exportados para análise posterior. A Figura 2.1 ilustra uma das interfaces de visualização de dados do SiBBr.

Figura 2.1 – SiBBr



Fonte: <https://www.sibbr.gov.br/>. Acesso em 30/07/2024.

2.2 Global Biodiversity Information Facility (GBIF)

O Global Biodiversity Information Facility (GBIF) é uma rede internacional que fornece acesso a dados sobre biodiversidade de todo o mundo (Global Biodiversity Information Facility, 2024). Ele agrega dados de várias fontes globais, oferecendo uma plataforma robusta para explorar a distribuição de espécies e realizar análises de biodiversidade.

2.2.1 Funcionalidades Principais

O GBIF apresenta várias funcionalidades que facilitam a coleta e análise de dados sobre a biodiversidade:

- **Integração de Dados Globais:** acesso a dados de ocorrência de espécies provenientes de diversas fontes ao redor do mundo, permitindo uma visão abrangente e consolidada da biodiversidade global.

- **Ferramentas de Análise Geoespacial:** disponibiliza mapas interativos e ferramentas de análise espacial que facilitam a visualização e o estudo da distribuição das espécies em diferentes regiões geográficas.
- **Acesso Aberto e Distribuído:** fornece acesso livre e aberto a todos os dados, promovendo a transparência e incentivando a colaboração entre pesquisadores, instituições e o público em geral.
- **Suporte a Publicações Científicas:** oferece infraestrutura e padrões para a publicação e compartilhamento de dados e metadados de biodiversidade, auxiliando na disseminação e reutilização de informações científicas de forma padronizada.

2.2.2 Impacto e Utilização

O GBIF tem um impacto significativo na pesquisa científica global, permitindo que pesquisadores e gestores ambientais acessem dados abrangentes sobre biodiversidade. A plataforma apoia a formulação de políticas públicas e iniciativas de conservação em todo o mundo.

2.2.3 Comparação com MamInAgro

Enquanto o GBIF fornece uma visão global da biodiversidade, o MamInAgro se concentra em dados específicos de mamíferos em paisagens agrícolas brasileiras. O MamInAgro agrega dados de ocorrência e caracteriza ambientes agrícolas, oferecendo uma perspectiva mais focada e relevante para estudos locais.

2.2.4 Visualização de Dados no GBIF

O GBIF oferece ferramentas avançadas de visualização de dados, permitindo que os usuários explorem a distribuição de espécies e realizem análises detalhadas. A Figura 2.2 exemplifica uma das interfaces de visualização disponíveis no GBIF.

Figura 2.2 – GBIF



Fonte: <https://www.gbif.org>. Acesso em 30/07/2024.

2.3 NatureServe

NatureServe desenvolve padrões para coleta e rastreamento do status de espécies ameaçadas e sistemas ecológicos (NatureServe Explorer, 2024). Eles criam ferramentas de visualização de dados que ajudam a monitorar a saúde e as tendências da biodiversidade.

2.3.1 Funcionalidades Principais

NatureServe oferece várias ferramentas para a coleta e análise de dados sobre biodiversidade:

- **Ferramenta de Revisão Ambiental (*Environmental Review Tool*):** plataforma customizável que permite o acesso e a análise interativa de dados sobre recursos naturais, auxiliando na avaliação de impactos ambientais e no planejamento de conservação.
- **Painéis de Biodiversidade (*Biodiversity Dashboards*):** dashboards interativos que

fornecem visualizações detalhadas sobre o estado e as tendências da biodiversidade em diferentes ecossistemas e regiões geográficas.

- **iMapInvasives:** aplicativo especializado no monitoramento e gestão de espécies invasoras, oferecendo dados em tempo real sobre a distribuição e a propagação dessas espécies, facilitando ações de controle e mitigação.
- **Centralização de Dados de Conservação:** consolida informações sobre o status de conservação de espécies e habitats, fornecendo uma base de dados robusta para apoiar decisões e políticas relacionadas à preservação da biodiversidade.

2.3.2 Impacto e Utilização

NatureServe tem um impacto significativo na conservação da biodiversidade, fornecendo ferramentas para monitorar a saúde dos ecossistemas e tomar decisões informadas sobre conservação.

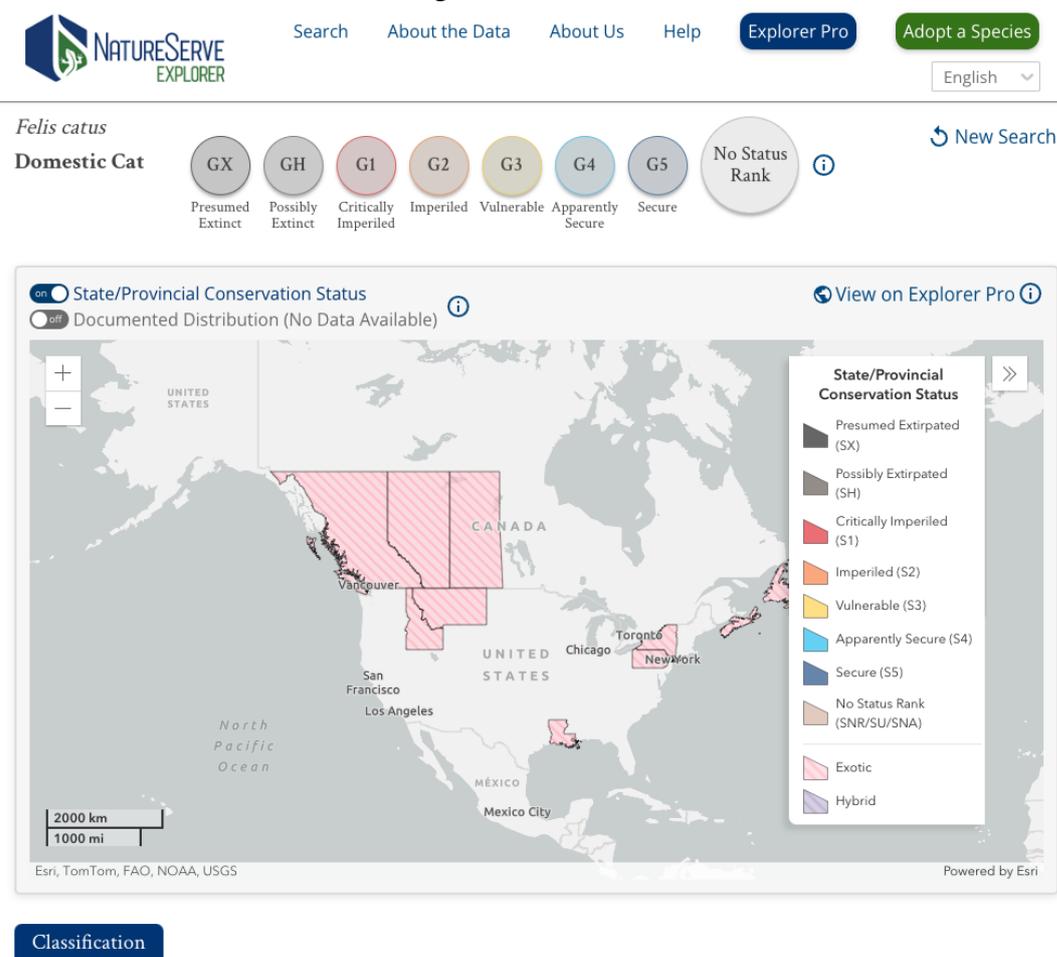
2.3.3 Comparação com MamInAgro

NatureServe oferece uma variedade de ferramentas de visualização de dados e análise geoespacial, enquanto o MamInAgro foca especificamente em mamíferos em paisagens agrícolas, oferecendo uma visão detalhada e específica para esse contexto.

2.3.4 Visualização de Dados no NatureServe

NatureServe oferece dashboards interativos que permitem visualizar a saúde e as tendências da biodiversidade. A Figura 2.3 ilustra uma das interfaces de visualização de dados do NatureServe.

Figura 2.3 – NatureServe



Scientific Name: *Felis catus* Linnaeus, 1758

Other Common Names: Chat (FR), Chat domestique (FR), Chat haret (FR), Feral Cat (EN), Gato (PT), Gato Doméstico (ES)

Kingdom: Animalia

Phylum: Craniata

Class: Mammalia

Order: Carnivora

Family: Felidae

Genus: Felis

Fonte: <https://explorer.natureserve.org/>. Acesso em 30/07/2024.

2.4 Atlas of Living Australia (ALA)

O Atlas of Living Australia (ALA) é uma infraestrutura de pesquisa que integra dados de diversas fontes australianas, oferecendo ferramentas de mapeamento e visualização de dados de biodiversidade (Atlas of Living Australia, 2024).

2.4.1 Funcionalidades Principais

O ALA apresenta várias funcionalidades que facilitam a coleta e análise de dados sobre a biodiversidade:

- **Ferramentas de Mapeamento Interativo:** oferece mapas interativos que permitem explorar e visualizar a distribuição de espécies em todo o território australiano, facilitando a compreensão de padrões espaciais de biodiversidade.
- **Acesso a Coleções Biológicas:** integra dados de coleções biológicas de diversas instituições, disponibilizando informações detalhadas sobre espécimes, incluindo características, localização e condições de coleta.
- **Ferramentas de Pesquisa e Análise:** disponibiliza uma série de ferramentas que permitem realizar pesquisas avançadas e análises geoespaciais, apoiando estudos científicos e projetos de conservação.
- **Integração de Dados Nacionais:** agrega e padroniza dados de biodiversidade provenientes de múltiplas fontes nacionais, proporcionando uma base de dados completa e confiável para pesquisadores e gestores ambientais.

2.4.2 Impacto e Utilização

O ALA apoia a pesquisa científica e a conservação da biodiversidade na Austrália, oferecendo acesso a uma vasta gama de dados e ferramentas de análise.

2.4.3 Comparação com MamInAgro

Enquanto o ALA oferece uma visão abrangente da biodiversidade australiana, o MamInAgro se concentra em mamíferos em paisagens agrícolas brasileiras, oferecendo uma perspectiva mais focada e específica para estudos locais.

2.4.4 Visualização de Dados no ALA

O ALA oferece ferramentas avançadas de visualização de dados, permitindo que os usuários explorem a distribuição de espécies e realizem análises detalhadas. A Figura

2.4 mostra a interface de visualização do Atlas of Living Australia.

Figura 2.4 – Atlas of Living Australia

***Felis catus* Linnaeus, 1758** API

species Aceto Autoridade do nome: AFD

Cat

Budjigarr in Yuwaalaraay
 Bujibuji in Rembarrnga
 Bujigat in Ritharrngu
 Bujigat in Kriol
 Burrjiyan in Gamilaraay / Gamilaroi / Kamilaroi
 Garnbulanyi in Marra

Budjigarr in Yuwaalaraay
 Bujigan in Wubuy
 Bujigat in Ngalakgan
 Bujigat in Ngandi
 Gajakkurru in Rembarrnga
 Nyun.Ngarri in Alawa

Resumo Galeria Nomes Classificação Registros Literatura Publicadores

Mapa registro de ocorrências (40.497 registro)

Ver mapa interativo Ver os registros

Record a sighting Receba alertas quando novos registros forem adicionados

Fonte: <https://biocache.ala.org.au/>. Acesso em 30/07/2024.

Essas ferramentas são essenciais para pesquisadores e gestores ambientais, permitindo uma compreensão abrangente da biodiversidade e auxiliando na tomada de decisões para a conservação e uso sustentável dos recursos naturais.

2.5 Comparação entre os Trabalhos Relacionados

Nesta seção, é apresentada uma tabela comparativa que resume as principais características de visualização de dados das plataformas discutidas anteriormente: SiBBR, GBIF, NatureServe e Atlas of Living Australia (ALA).

Tabela 2.1 – Comparação das Funcionalidades de Visualização de Dados

Funcionalidades de Visualização	SiBBr	GBIF	NatureServe	ALA
Mapas Interativos	X	X	X	X
Dashboards Personalizados			X	
Análise Geoespacial	X	X	X	X
Ferramentas de Mapeamento	X	X	X	X
Visualização de Espécies por Região	X	X		X
Criação de Mapas Personalizados	X			

Fonte: De autoria própria

A Tabela 2.1 destaca as funcionalidades de visualização de dados das plataformas SiBBr, GBIF, NatureServe e ALA. Todas as plataformas oferecem mapas interativos e ferramentas de mapeamento, permitindo aos usuários explorar e analisar a distribuição de espécies. NatureServe se destaca por oferecer dashboards personalizados e ferramentas específicas para o monitoramento de espécies invasoras, enquanto o SiBBr permite a criação de mapas personalizados, uma funcionalidade não encontrada em GBIF e ALA.

3 MAMINAGRO E REQUISITOS DA APLICAÇÃO

Este capítulo apresenta o MamInAgro, uma aplicação web projetada para armazenar e analisar dados sobre mamíferos em paisagens agrícolas brasileiras, detalhando suas funcionalidades iniciais, as limitações identificadas e as melhorias implementadas. São descritas as novas funcionalidades adicionadas ao sistema, incluindo cálculos, visualização de dados e o desenvolvimento de um backoffice para monitoramento e controle. Além disso, são discutidos os requisitos essenciais que guiaram o desenvolvimento e a expansão da aplicação..

3.1 MamInAgro

MamInAgro é uma aplicação web que integra um banco de dados para armazenar informações sobre mamíferos em paisagens agrícolas brasileiras. A aplicação foi desenvolvida como parte de um trabalho de conclusão de curso(SANTIN, 2024), com o objetivo de consolidar dados frequentemente divulgados em estudos da área, permitindo um armazenamento eficiente e uma visualização acessível para pesquisadores e interessados.

A parte web da aplicação foi projetada para ser intuitiva, com uma interface que facilita a inserção e consulta de dados. O banco de dados associado organiza as informações de forma que possam ser facilmente acessadas e manipuladas para diversas análises. A aplicação utiliza tecnologias modernas para garantir a segurança e a integridade dos dados, além de oferecer uma experiência de usuário agradável. A Figura 3.1 ilustra uma das interfaces principais do MamInAgro

Figura 3.1 – MamInAgro

The screenshot shows the 'Paisagem' (Landscape) section of the MamInAgro application. At the top, there is a navigation bar with 'Home', 'Taxonomia', 'Referência', 'Ocorrência', 'Paisagem' (highlighted), 'Cultura', and 'Usuários'. Below the navigation bar, there is a form titled 'Criar Paisagem' (Create Landscape) with the following fields: Latitude, Longitude, Raio (Radius), Descrição (Description), Tipo de Agroecossistema (Agroecosystem Type), Antroma (Antroma), Tipo de Área Amostrada (Sampled Area Type), Referência (Reference), and a checkbox for 'É Área Protegida?' (Is it a Protected Area?). A 'Salvar' (Save) button is located at the bottom right of the form.

Below the form is a table displaying a list of landscapes. The table has the following columns: Latitude, Longitude, Raio, Descrição, Agroecossistema, Antroma, Tipo de Área Amostrada, and É Área Protegida?. The table contains 8 rows of data.

Latitude	Longitude	Raio	Descrição	Agroecossistema	Antroma	Tipo de Área Amostrada	É Área Protegida?
-8.483333	-37.333333		Mosaico de áreas com maior cobertura florestal até áreas com maior dominância da fisionomia ligada à agricultura e pecuária	Cattle Ranching			NÃO
17374	17374		Abc				NÃO
177483	847373	1	And				NÃO
1	1	2222	Lamaçal	Agrofloresta	Ant teste	Área Protegida	SIM
1	1	222	Raios	Plantação de Dinheiro	Zona de Conservação Interplanetária	Amostragem Aleatória Descriteriosa	NÃO
123	-123	1	Test	Agrofloresta	Vila de São Roque	Área Protegida	SIM
32	23	23	Área permanentemente encoberta por neblina.	Plantação de Dinheiro	Zona de Conservação Interplanetária	Amostragem Aleatória Descriteriosa	NÃO

At the bottom of the table, there is a pagination control showing a page number '1' in a blue box, with arrows on either side.

Fonte: De autoria própria

A arquitetura da aplicação foi desenhada para ser escalável. O backend é responsável pelo processamento das requisições, enquanto o frontend exibe as informações de maneira clara e organizada. Esse sistema é essencial para o acompanhamento de como a agricultura impacta a biodiversidade, especialmente em relação aos mamíferos que habitam essas áreas.

3.2 Limitações e Novas Funcionalidades

Apesar de sua eficácia, a versão inicial do MamInAgro apresentou algumas limitações. Entre elas, a ausência de suporte para a realização de métricas utilizadas nos estudos e a falta de uma interface para visualizar os dados que estavam sendo inseridos na aplicação. Estas limitações dificultavam a análise mais aprofundada dos dados e a geração de insights relevantes para a conservação da biodiversidade.

Para superar essas limitações, foram desenvolvidas novas funcionalidades focadas em dois aspectos principais: cálculos e visualização de dados. As novas funcionalida-

des de cálculos foram integradas ao backend da aplicação, permitindo que métricas mais complexas e específicas fossem realizadas diretamente na plataforma. Além disso, uma nova proposta de visualização de dados foi implementada, proporcionando uma interface intuitiva onde os resultados dos cálculos e os dados consolidados podem ser facilmente explorados.

Além dessas melhorias, foi desenvolvido um **backoffice** utilizando a tecnologia React, a mesma utilizada na aplicação web do MamInAgro. Esse backoffice possui funcionalidades essenciais para monitoramento e controle, como um **login** (usuário e senha) para acesso seguro dos principais administradores do projeto, e uma **home** que exibe gráficos sobre a entrada de valores nas tabelas do sistema, como, por exemplo, o número de novos biomas por dia.

Duas funcionalidades principais também foram adicionadas ao backoffice: um botão para rodar os cálculos, que executa scripts em R diretamente do backend, e outro botão que direciona para a URL do Metabase.

Essas melhorias tornam o MamInAgro uma ferramenta ainda mais poderosa para pesquisadores, permitindo não só o armazenamento e a consulta de dados, mas também a análise detalhada de como os fatores agrícolas estão impactando a biodiversidade.

3.3 Requisitos do Trabalho

Com base nas reuniões com o Professor Pedro Cordeiro Estrela de Andrade Pinto, professor da Universidade Federal da Paraíba (UFPB) e curador do Museu de Biodiversidade da UFPB, e sua equipe, foram definidos os seguintes requisitos para o desenvolvimento do MamInAgro:

1. **Integração de Cálculos Específicos:** A aplicação deve suportar a execução de cálculos complexos, inicialmente utilizando os cálculos desenvolvidos em R para gerar resultados para futuras análises. Embora a possibilidade de trocar a linguagem dos cálculos para Python tenha sido considerada, o requisito inicial foi manter os scripts em R, se possível, devido ao conhecimento técnico da equipe responsável pela manutenção futura.
2. **Escalabilidade e Desempenho:** A aplicação deve ser capaz de lidar com grandes volumes de dados e acessos simultâneos, garantindo a integridade dos dados e o desempenho da plataforma.

3. **Flexibilidade na Manutenção e Expansão:** A arquitetura da aplicação deve ser modular e flexível, permitindo futuras expansões e atualizações sem comprometer o funcionamento atual.

Esses requisitos foram estabelecidos para assegurar que o MamInAgro não só supere as limitações da versão inicial, mas também ofereça uma plataforma robusta e eficiente para o estudo e a conservação da biodiversidade em paisagens agrícolas.

3.4 Evolução dos Requisitos

Inicialmente, o projeto MamInAgro foi concebido com três requisitos principais: **Integração de Cálculos Específicos, Escalabilidade e Desempenho, e Flexibilidade na Manutenção e Expansão**. Esses requisitos refletiam a necessidade de uma solução que permitisse a execução de cálculos, e que fosse capaz de lidar com a capacidade atual dos dados e que fosse flexível para futuras expansões.

No entanto, à medida que o desenvolvimento avançava e as primeiras iterações de integração dos cálculos foram validadas, percebeu-se que seria necessária a utilização do banco de dados para guardar os resultados das operações de cálculo. A necessidade de armazenar os resultados de forma estruturada surgiu como uma prioridade para assegurar que esses dados pudessem ser facilmente consultados e analisados posteriormente. Esse requisito foi então adicionado ao escopo do projeto, resultando no **Armazenamento Estruturado dos Resultados**.

Com o armazenamento dos resultados devidamente implementado, em uma nova reunião uma nova proposta foi apresentada: criar uma ferramenta de visualização de dados que permitisse aos usuários explorar de maneira mais útil os dados consolidados dos cálculos. Essa sugestão levou à inclusão do requisito de **Visualização de Dados**, garantindo que o MamInAgro não fosse apenas uma ferramenta de armazenamento e cálculo, mas também uma ferramenta para análise e tomada de decisões.

4 APLICAÇÃO

Neste capítulo, serão apresentadas as bases das soluções propostas para a mudança de cálculos feitos em R e para a visualização desses dados consolidados em um ambiente próprio de visualização de dados. O processo foi dividido em várias etapas, detalhadas nas subseções a seguir.

4.1 Métodos Levantados para a Alteração dos Cálculos em R

Diversas abordagens foram consideradas para a mudança dos cálculos feitos originalmente em R. As principais alternativas levantadas foram: a utilização de `rpy2` para integração entre Python e R, a mudança dos cálculos diretamente no R com armazenamento dos resultados no banco de dados, e a tradução dos códigos R para Python. Cada uma dessas alternativas é detalhada a seguir, juntamente com suas respectivas vantagens e desvantagens. Abaixo será mostrado o código que será usado como base para o teste das alternativas:

```
# Conexão com o banco de dados
library(DBI)
library(RPostgreSQL)

# Conectando ao banco de dados PostgreSQL
con <- dbConnect(RPostgreSQL::PostgreSQL(),
                 dbname = dbname,
                 host = dbhost,
                 port = dbport,
                 user = dbuser,
                 password = dbpass)

# Consulta para obter os dados necessários para o cálculo
query <- "
SELECT b.id AS biome_id, td.risk_classification
FROM Biomes b
JOIN BiomeLandscapes bl ON bl.biome_id = b.id
```

```
JOIN Landscapes l ON l.id = bl.landscape_id
JOIN LandscapeLocalities ll ON ll.landscape_id = l.id
JOIN Localities loc ON loc.id = ll.locality_id
JOIN Occurrence o ON o.locality_id = loc.id
JOIN ThreatDegrees td ON td.species_id = o.species_id
"

# Executando a consulta
resultados <- dbGetQuery(con, query)

# Mapeando classificações de risco para valores numéricos
risk_classification_dict <- c(
  "Baixo Risco" = 1,
  "Moderado Risco" = 2,
  "Alto Risco" = 3
)

# Convertendo classificações de risco para valores numéricos
resultados$risk_score <- risk_classification_dict[
  resultados$risk_classification
]

# Calculando a média de risco por bioma
risk_scores_avg <- aggregate(
  resultados$risk_score,
  by = list(resultados$biome_id),
  FUN = mean
)

colnames(risk_scores_avg) <- c("biome_id", "average_risk_score")

# Gerando o gráfico com ggplot2
library(ggplot2)
biome_names <- c(
  "Amazônia", "Cerrado", "Mata Atlântica",
```

```

    "Caatinga", "Pampa", "Pantanal"
)
risk_scores_avg$biome_name <- factor(
  risk_scores_avg$biome_id,
  levels = 1:6,
  labels = biome_names
)

ggplot(risk_scores_avg, aes(x = biome_name, y = average_risk_score)) +
  geom_bar(stat = "identity") +
  theme_minimal() +
  labs(title = "Risco Médio por Bioma",
        x = "Bioma",
        y = "Risco Médio")

# Fechando a conexão com o banco de dados
dbDisconnect(con)

```

4.1.1 Utilizando rpy2 para Chamada de Cálculos em R via Python

Uma das alternativas propostas é utilizar a biblioteca `rpy2`, que permite a integração entre Python e R, permitindo que códigos R sejam executados diretamente a partir de scripts Python (Rpy2 Documentation, 2024). Essa abordagem possibilita o uso das bibliotecas gráficas de Python para a visualização dos resultados.

4.1.1.1 Integração Python-R com rpy2

A biblioteca `rpy2` facilita a execução de códigos R dentro de um ambiente Python. Com ela, é possível chamar funções R, passar dados entre Python e R e utilizar os pacotes do R diretamente. Esse método, aplicado ao código anterior, é mostrado a seguir:

```

import rpy2.robjects as ro
from rpy2.robjects.packages import importr

```

```
# Importando as bibliotecas R necessárias
ro.r('library(DBI)')
ro.r('library(RPostgreSQL)')
ro.r('library(ggplot2)')

# Código R para conexão e execução da consulta
ro.r(f'''
# Conectando ao banco de dados PostgreSQL
con <- dbConnect(RPostgreSQL::PostgreSQL(),
                 dbname = '{dbname}',
                 host = '{dbhost}',
                 port = '{dbport}',
                 user = '{dbuser}',
                 password = '{dbpass}')

# Consulta para obter os dados necessários para o cálculo
query <- "
    SELECT b.id AS biome_id, td.risk_classification
    FROM Biomes b
    JOIN BiomeLandscapes bl ON bl.biome_id = b.id
    JOIN Landscapes l ON l.id = bl.landscape_id
    JOIN LandscapeLocalities ll ON ll.landscape_id = l.id
    JOIN Localities loc ON loc.id = ll.locality_id
    JOIN Occurrence o ON o.locality_id = loc.id
    JOIN ThreatDegrees td ON td.species_id = o.species_id
"

# Executando a consulta
resultados <- dbGetQuery(con, query)

# Mapeando classificações de risco para valores numéricos
risk_classification_dict <- c(
  "Baixo Risco" = 1,
  "Moderado Risco" = 2,
```

```
"Alto Risco" = 3
)

# Convertendo classificações de risco para valores numéricos
resultados$risk_score <- risk_classification_dict[
  resultados$risk_classification
]

# Calculando a média de risco por bioma
risk_scores_avg <- aggregate(
  resultados$risk_score,
  by = list(resultados$biome_id),
  FUN = mean
)
colnames(risk_scores_avg) <- c("biome_id", "average_risk_score")

# Gerando o gráfico com ggplot2
biome_names <- c(
  "Amazônia", "Cerrado", "Mata Atlântica",
  "Caatinga", "Pampa", "Pantanal"
)
risk_scores_avg$biome_name <- factor(
  risk_scores_avg$biome_id,
  levels = 1:6,
  labels = biome_names
)

ggplot(risk_scores_avg, aes(x = biome_name, y = average_risk_score)) +
  geom_bar(stat = "identity") +
  theme_minimal() +
  labs(title = "Risco Médio por Bioma",
       x = "Bioma",
       y = "Risco Médio")
```

```
# Fechando a conexão com o banco de dados
dbDisconnect(con)
''' )
```

4.1.1.2 Vantagens e Desvantagens da Integração Python-R

Vantagens:

- Permite a utilização de bibliotecas avançadas de visualização de dados em Python.
- Facilita a integração de códigos já existentes em R com novos desenvolvimentos em Python.
- Flexibilidade para utilizar a vasta gama de pacotes disponíveis tanto em R quanto em Python.

Desvantagens:

- Pode introduzir complexidade adicional na configuração e manutenção do ambiente.
- A performance pode ser um problema devido à sobrecarga de chamadas interlinguagens.
- Necessidade de conhecimento tanto em R quanto em Python para a implementação e manutenção.

4.1.2 Mudança dos Cálculos Diretamente em R e Armazenamento dos Resultados no Banco de Dados

Outra alternativa é realizar a mudança dos cálculos diretamente no R, configurando o script R para salvar os resultados de cada função no banco de dados da aplicação. Essa abordagem permite que os resultados sejam armazenados de forma estruturada e possam ser facilmente acessados por qualquer ferramenta de visualização de dados.

4.1.2.1 Execução de Cálculos em R

Os scripts R podem ser adaptados para salvar os resultados em um banco de dados após a execução dos cálculos. Isso pode ser feito utilizando pacotes R como DBI e RPostgreSQL. O pacote DBI fornece uma interface genérica para comunicação entre

R e sistemas de gerenciamento de banco de dados, facilitando a conexão e a execução de consultas SQL ((R-SIG-DB), 2021). Para a interação específica com bancos de dados PostgreSQL, o pacote RPostgreSQL é utilizado, permitindo que os dados sejam inseridos, atualizados e consultados diretamente no banco de dados a partir do ambiente R (RPostgreSQL, 2024). Esse método, aplicado ao código inicial, é mostrado a seguir:

```
library(DBI)
library(RPostgreSQL)

# Conectando ao banco de dados PostgreSQL
con <- dbConnect(RPostgreSQL::PostgreSQL(),
                 dbname = dbname,
                 host = dbhost,
                 port = dbport,
                 user = dbuser,
                 password = dbpass)

# Consulta para obter os dados necessários para o cálculo
query <- "
SELECT b.id AS biome_id, td.risk_classification
FROM Biomes b
JOIN BiomeLandscapes bl ON bl.biome_id = b.id
JOIN Landscapes l ON l.id = bl.landscape_id
JOIN LandscapeLocalities ll ON ll.landscape_id = l.id
JOIN Localities loc ON loc.id = ll.locality_id
JOIN Occurrence o ON o.locality_id = loc.id
JOIN ThreatDegrees td ON td.species_id = o.species_id
"

# Executando a consulta
resultados <- dbGetQuery(con, query)

# Mapeando classificações de risco para valores numéricos
risk_classification_dict <- c(
  "Baixo Risco" = 1,
```

```

    "Moderado Risco" = 2,
    "Alto Risco" = 3
)

# Convertendo classificações de risco para valores numéricos
resultados$risk_score <- risk_classification_dict[
  resultados$risk_classification
]

# Calculando a média de risco por bioma
risk_scores_avg <- aggregate(
  resultados$risk_score,
  by = list(resultados$biome_id),
  FUN = mean
)

colnames(risk_scores_avg) <- c("biome_id", "average_risk_score")

# Adicionando a data de cálculo
risk_scores_avg$calculation_date <- Sys.Date()

# Inserindo os resultados na tabela calculos_risco_bioma
dbWriteTable(con, "calculos_risco_bioma",
  risk_scores_avg,
  append = TRUE,
  row.names = FALSE)

# Fechando a conexão com o banco de dados
dbDisconnect(con)

```

4.1.2.2 *Vantagens e Desvantagens de Realizar os Cálculos Diretamente em R*

Vantagens:

- Permite o uso completo do ambiente R e suas bibliotecas, que são bem estabelecidas para análises estatísticas.

- Armazenamento estruturado dos resultados diretamente no banco de dados, facilitando o acesso para visualização.

Desvantagens:

- Dependência contínua do ambiente R, o que pode ser uma limitação se houver necessidade de migrar para outra linguagem.
- Necessidade de configuração e manutenção da integração entre R e o banco de dados.

4.1.3 Tradução do Código R para Python

A terceira alternativa é traduzir os códigos R para Python. Essa abordagem permite que os cálculos sejam realizados diretamente em Python, utilizando bibliotecas nativas como `numpy` e `pandas`. O `numpy` é utilizado para realizar operações numéricas de maneira eficiente e fornece suporte extensivo para arrays e funções matemáticas (NumPy Developers, 2024). O `pandas` é utilizado para manipulação de dados e oferece estruturas de dados e ferramentas para análise de dados tabulares (Pandas Developers, 2024).

4.1.3.1 Tradução de Códigos R para Python

Traduzir os scripts R para Python envolve a substituição de funções e pacotes R por suas equivalentes em Python. A seguir está um exemplo de tradução do código inicial:

```
import psycopg2
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

# Conectando ao banco de dados PostgreSQL
conn = psycopg2.connect (
    dbname=dbname,
    host=dbhost,
    port=dbport,
    user=dbuser,
    password=dbpass
```

```

)

# Consulta para obter os dados necessários para o cálculo
query = """
    SELECT b.id AS biome_id, td.risk_classification
    FROM Biomes b
    JOIN BiomeLandscapes bl ON bl.biome_id = b.id
    JOIN Landscapes l ON l.id = bl.landscape_id
    JOIN LandscapeLocalities ll ON ll.landscape_id = l.id
    JOIN Localities loc ON loc.id = ll.locality_id
    JOIN Occurrence o ON o.locality_id = loc.id
    JOIN ThreatDegrees td ON td.species_id = o.species_id
    """
resultados = pd.read_sql(query, conn)

# Mapeando classificações de risco para valores numéricos
risk_classification_dict = {
    "Baixo Risco": 1,
    "Moderado Risco": 2,
    "Alto Risco": 3
}
resultados['risk_score'] = resultados['risk_classification'].map(
    risk_classification_dict
)

# Calculando a média de risco por bioma
risk_scores_avg = resultados.groupby('biome_id')['risk_score'] \
    .mean() \
    .reset_index()

# Adicionando os nomes dos biomas
biome_names = {
    1: "Amazônia",
    2: "Cerrado",

```

```

    3: "Mata Atlântica",
    4: "Caatinga",
    5: "Pampa",
    6: "Pantanal"
}
risk_scores_avg['biome_name'] = risk_scores_avg['biome_id']
.map(biome_names)

# Plotando o gráfico
plt.figure(figsize=(10, 6))
plt.bar(risk_scores_avg['biome_name'], risk_scores_avg['risk_score'],
color='skyblue')
plt.title("Risco Médio por Bioma")
plt.xlabel("Bioma")
plt.ylabel("Risco Médio")
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()

# Fechando a conexão com o banco de dados
conn.close()

```

4.1.3.2 Vantagens e Desvantagens da Tradução para Python

Vantagens:

- Elimina a necessidade de integração interlinguagem, simplificando a arquitetura do sistema.
- Aproveita a vasta gama de bibliotecas Python para análise e visualização de dados.
- Facilita a manutenção e evolução do código, utilizando uma única linguagem de programação.

Desvantagens:

- Requer a reescrita completa dos scripts R, o que pode ser trabalhoso e sujeito a erros.

- Pode haver diferenças nos resultados devido a variações nas implementações das funções entre R e Python.
- Necessidade de garantir que todas as funcionalidades do código original em R sejam corretamente replicadas em Python.

4.2 Escolha da Atualização dos Scripts de R

Com base nas análises das alternativas levantadas, optou-se por atualizar os scripts de R para salvar os valores diretamente em novas tabelas no banco de dados. Um fator que pesou bastante nessa escolha é o fato de que a equipe destinada para as manunτεύções futuras desse script tem sua base de conhecimento de programação em R. Além disso salvar os resultados dos cálculos no banco de dados foi suficiente para a aplicação não ficar dependente da estrutura do R para fazer toda a etapa de visualização. Para isso, foi criado um novo esquema denominado *calculations*, dedicado exclusivamente à área de cálculos. Este esquema contém tabelas que refletem os dados consolidados dos cálculos realizados pelos scripts em R.

A decisão de criar novas tabelas no esquema *calculations* foi motivada pela necessidade de manter uma organização clara e uma separação lógica dos dados, facilitando a manutenção e a consulta futura. Cada tabela dentro deste esquema é projetada para armazenar resultados específicos dos cálculos, garantindo que os dados estejam prontamente disponíveis para análise e visualização.

4.3 Visualização de Dados

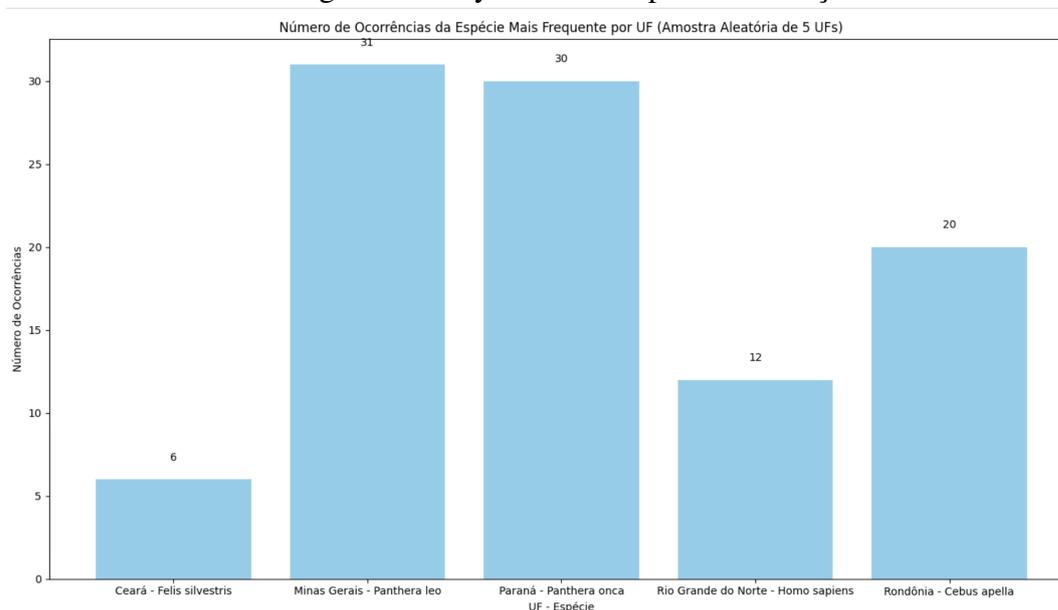
Nesta seção, serão apresentadas as abordagens para visualização de dados que foram consideradas e implementadas no sistema. Serão abordadas tanto a visualização dos dados em Python quanto a utilização de ferramentas de BI.

4.3.1 Visualização de Dados em Python

Após a execução dos cálculos, os resultados podem ser visualizados utilizando bibliotecas Python, como `matplotlib`, `seaborn` ou `plotly`. A `matplotlib` é utilizada para criar visualizações estáticas, como gráficos de linha e histogramas, ofere-

cendo controle detalhado sobre todos os aspectos da renderização dos gráficos (Matplotlib, 2024). Para gráficos mais estilizados e com maior facilidade na criação de visualizações estatísticas, a `seaborn` é uma excelente escolha, pois se baseia na `matplotlib`, mas oferece uma interface de alto nível para gerar visualizações atraentes e informativas (Seaborn, 2024). Além disso, para gráficos interativos e personalizáveis, a `plotly` se destaca, permitindo a criação de visualizações dinâmicas que podem ser exploradas diretamente no navegador (Plotly Technologies Inc., 2024). Na Figura 4.1 temos um exemplo de visualização de dados utilizando `matplotlib`.

Figura 4.1 – Python - Exemplo Visualização



Fonte: De autoria própria

4.3.1.1 Vantagens e Desvantagens da Visualização em Python

Vantagens:

- Flexibilidade para criar visualizações altamente customizadas e interativas.
- Forte integração com outros pacotes Python, facilitando a análise e manipulação dos dados.

Desvantagens:

- Requer conhecimentos avançados em programação para criar e customizar visualizações complexas.

- Menos intuitivo e acessível para usuários não técnicos em comparação com ferramentas de BI.

4.3.2 Visualização de Dados com Ferramentas de BI

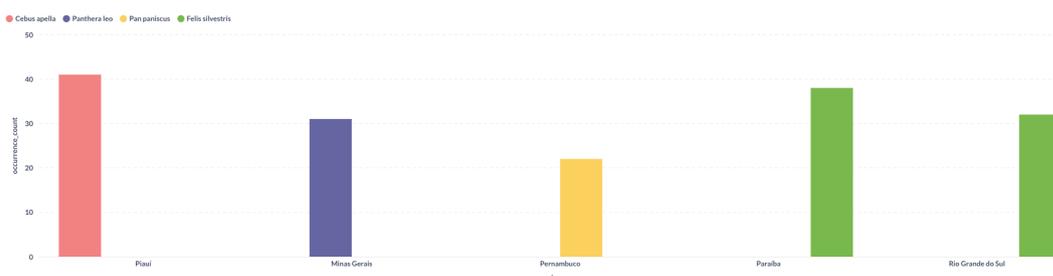
O Metabase é uma ferramenta de visualização de dados open-source que permite criar dashboards interativos e realizar consultas dinâmicas nos dados. Ele é fácil de usar, requer pouca configuração e oferece uma interface intuitiva para análise de dados (INC., 2024).

A escolha do Metabase se deu por sua facilidade de integração com o banco de dados PostgreSQL, suporte a visões materializadas (VWM), e sua capacidade de atender às necessidades do projeto com uma interface amigável.

Outras ferramentas de mercado, como o Power BI e o Apache Superset, foram consideradas, mas apresentaram limitações em relação às necessidades específicas do projeto. O Power BI, por exemplo, não oferece suporte nativo para visões materializadas e utiliza uma linguagem própria para consultas, o que poderia complicar a integração e o uso pelos desenvolvedores (Microsoft Power BI, 2024). Já o Apache Superset, apesar de ser uma ferramenta poderosa e também open-source, possui uma curva de aprendizado mais íngreme e requer configurações mais complexas para alcançar o mesmo nível de funcionalidade que o Metabase oferece de forma mais simplificada (Apache Superset, 2024).

Esses aspectos levaram à conclusão de que o Metabase seria a escolha ideal para a visualização de dados no projeto, combinando robustez, facilidade de uso, e integração eficiente com o banco de dados existente. Na Figura 4.2 temos um exemplo de visualização de dados utilizando metabase.

Figura 4.2 – Metabase - Exemplo Visualização



Fonte: De autoria própria

4.3.2.1 Vantagens e Desvantagens do Metabase

Vantagens:

- Interface amigável e intuitiva, adequada para usuários com pouca experiência técnica.
- Suporte a uma ampla gama de bancos de dados, incluindo PostgreSQL, MySQL, MongoDB, entre outros.
- Possibilidade de criar dashboards interativos e compartilhá-los com outros membros da equipe.
- Ferramentas de consulta dinâmica que permitem explorar os dados sem a necessidade de escrever SQL.

Desvantagens:

- Funcionalidades mais avançadas podem ser limitadas em comparação com outras ferramentas de BI comerciais.
- Necessidade de configurar e manter a infraestrutura do Metabase, especialmente para grandes volumes de dados.
- Menor flexibilidade para personalização em comparação com soluções totalmente customizáveis em Python.

4.3.3 Escolha do Metabase como Ferramenta de Visualização de Dados

Após considerar as alternativas para visualização de dados, optou-se pelo uso do Metabase como a principal ferramenta de visualização devido à sua fácil integração com o PostgreSQL e a capacidade de criar dashboards interativos com pouca configuração. Além disso, o Metabase oferece uma interface amigável, o que facilita a adoção por parte dos usuários, mesmo aqueles com pouca experiência técnica. Sua flexibilidade para realizar consultas dinâmicas sem necessidade de programação foi um fator decisivo na escolha.

Outro fator importante na escolha do Metabase é que ele é uma ferramenta open-source amplamente adotada. Por isso, ele conta com uma comunidade ativa que constantemente contribui com atualizações, correções e novos recursos. Isso garante que a

ferramenta continuará evoluindo e se adaptando às necessidades futuras, sem depender de soluções proprietárias que podem se tornar obsoletas ao longo do tempo. Além disso, o Metabase oferece suporte a uma ampla variedade de integrações com diferentes bancos de dados e plataformas, o que facilita possíveis expansões e adaptações do sistema conforme novas tecnologias ou requisitos surgirem.

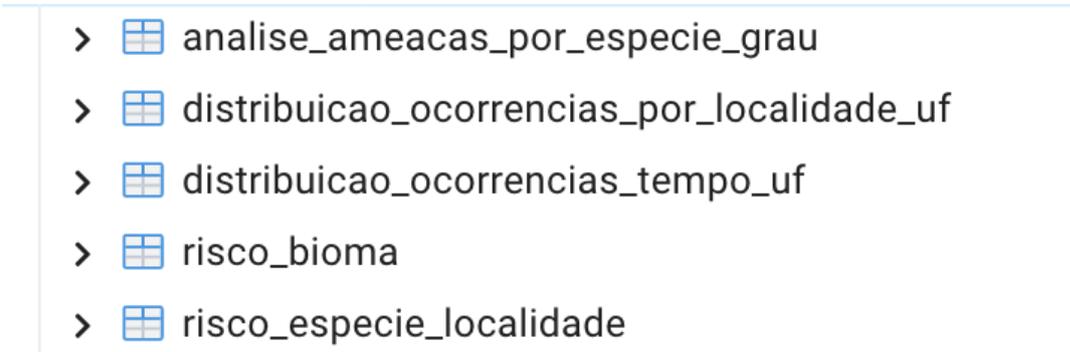
4.4 Mudanças no Banco de Dados

Para suportar os novos cálculos e a visualização dos dados, foi necessário realizar mudanças no banco de dados, criando novas tabelas e otimizando o armazenamento e a consulta dos resultados.

4.4.1 Criação de Tabelas no esquema calculations

Para organizar os dados consolidados dos cálculos, criou-se novas tabelas no esquema *calculations*. Cada tabela foi projetada para armazenar resultados específicos dos cálculos, garantindo uma separação lógica e facilitando a manutenção futura. A Figura 4.3 ilustra algumas tabelas do novo esquema.

Figura 4.3 – Banco - Tabelas Cálculos

- 
- >  `analise_ameacas_por_especie_grau`
 - >  `distribuicao_ocorrencias_por_localidade_uf`
 - >  `distribuicao_ocorrencias_tempo_uf`
 - >  `risco_bioma`
 - >  `risco_especie_localidade`

Fonte: De autoria própria

4.4.2 Escalabilidade e Uso de Visões Materializadas

Devido à necessidade de uma possível escalabilidade no projeto e ao fato de que os dados possam ser inseridos por diversos usuários e consultados por muitos outros simultaneamente, o acesso ao banco de dados e a visualização dos dados no Metabase podem ficar lentos. Isso é especialmente crítico quando há muitos acessos e inserções/cálculos acontecendo ao mesmo tempo.

Para mitigar esse problema, optou-se pela utilização de visões materializadas para o armazenamento final dos cálculos (Materialize, Inc., 2024). Cada tabela de resultado de cálculo dentro do esquema *calculations* tem sua própria visão materializada. As visões materializadas oferecem vários benefícios, como:

- **Performance Melhorada:** As visões materializadas armazenam os resultados das consultas de forma persistente, o que reduz o tempo de resposta para consultas subsequentes, pois os dados já estão pré-calculados.
- **Redução da Carga no Banco de Dados:** Ao utilizar visões materializadas, o banco de dados não precisa recalcular os resultados de consultas complexas repetidamente, diminuindo a carga de processamento.
- **Atualizações Controladas:** As visões materializadas podem ser atualizadas em intervalos definidos, garantindo que os dados estejam relativamente atualizados sem sobrecarregar o sistema com atualizações contínuas.

4.4.3 Criação de Visões Materializadas

Cada tabela de resultados dentro do esquema *calculations* tem sua própria visão materializada, que armazena os resultados das consultas de forma persistente, facilitando o acesso rápido aos dados consolidados. A Figura 4.4 ilustra algumas visões materializadas das tabelas de resultados dos cálculos.

Figura 4.4 – Banco - VWM's Cálculos

- >  vwm_analise_ameacas_por_especie_grau
- >  vwm_distribuicao_ocorrencias_por_localidade_uf
- >  vwm_distribuicao_ocorrencias_tempo_uf
- >  vwm_risco_bioma
- >  vwm_risco_especie_localidade

Fonte: De autoria própria

4.5 Execução dos Cálculos na Aplicação

Com base nas alternativas apresentadas anteriormente, foi necessário implementar uma solução que permitisse a execução dos scripts em R de maneira mais acessível e integrada ao sistema MamInAgro. Essa solução precisava garantir que os cálculos fossem realizados de forma eficiente.

4.5.1 Integração com Flask para Chamada de Cálculos R

Para integrar os scripts de R com o sistema MamInAgro, foi criado um microserviço em Python utilizando o framework Flask. Essa escolha se baseia na simplicidade e flexibilidade oferecidas pelo Flask, permitindo a criação rápida de APIs RESTful que facilitam a comunicação entre diferentes componentes do sistema (Pallets Projects, 2024).

Para rodar o script R dentro da API, foi utilizada a biblioteca `subprocess`, que permite a execução de comandos do sistema operacional a partir do Python, incluindo scripts em R (Python Documentation, 2024). Essa integração facilita a execução dos cálculos em R, armazenando os resultados nas tabelas apropriadas do esquema *calculations*.

Além disso, foi definida uma tarefa agendada que executa os cálculos e atualiza as visões materializadas diariamente, na última hora do dia. Essa configuração foi implementada para garantir que os dados não fiquem desatualizados por longos períodos, proporcionando uma base de dados mais precisa e confiável para a análise e visualização.

A facilidade de uso e a rápida implementação foram fatores determinantes para a

escolha do Flask. Sua capacidade de suportar a integração entre Python e R de maneira robusta e eficiente garante que os cálculos realizados em R sejam facilmente acessíveis e manipuláveis a partir do sistema principal.

4.6 Backoffice

Para complementar as funcionalidades do MamInAgro, foi desenvolvido um backoffice utilizando React, mantendo a coesão tecnológica com a aplicação principal (Facebook Inc., 2024). Este backoffice serve como uma interface administrativa, oferecendo um controle aprimorado sobre os dados que entram no sistema e as operações realizadas.

As principais funcionalidades do backoffice incluem:

- **Sistema de Login:** Para garantir a segurança e o controle de acesso, o backoffice possui uma tela de login onde os usuários devem autenticar-se com um nome de usuário e senha. A Figura 4.5 ilustra a interface de login do backoffice.
- **Gráficos de Monitoramento:** Uma vez autenticado, o usuário é direcionado para uma página inicial onde gráficos dinâmicos mostram informações como a quantidade de novos usuários por dia ou a inserção de dados em diferentes tabelas. A Figura 4.6 ilustra essa página inicial com seus respectivos gráficos de monitoramento. Os três gráficos presentes na figura representam as três últimas tabelas onde foram realizadas inserções. A principal finalidade desses gráficos é monitorar o uso do sistema, indicando a quantidade de inserções realizadas nos últimos dias, podendo identificar possíveis anomalias.
- **Execução de Cálculos:** Um botão dedicado permite a execução dos scripts em R por meio do microserviço da API Flask, garantindo que os cálculos sejam realizados e os resultados armazenados no banco de dados de forma eficiente.
- **Acesso ao Metabase:** Outro botão oferece um link direto para a URL do Metabase, permitindo que os administradores naveguem rapidamente para a ferramenta de visualização de dados.

4.6.1 Segurança com AWS Cognito

A segurança e o controle de acesso são fundamentais para o funcionamento do backoffice. Para garantir que apenas usuários autorizados possam acessar as funcionalidades críticas, como a execução de cálculos e a visualização de dados, foi integrado o serviço de autenticação e autorização AWS Cognito. Este serviço oferece uma série de vantagens que reforçam a segurança e a confiabilidade do sistema:

- **Segurança Gerenciada:** O AWS Cognito cuida de toda a parte de autenticação, gerenciamento de usuários e sessões de forma segura, eliminando a necessidade de desenvolver uma solução personalizada para login e autenticação. Isso reduz a chance de falhas de segurança relacionadas à implementação de autenticação própria.
- **Integração Simples com Frontend:** A integração com o React é simplificada, o que permite que a autenticação e a autorização sejam feitas diretamente pelo AWS Cognito sem a necessidade de passar pelos servidores da aplicação principal. Isso significa que a API do MamInAgro não precisa lidar com dados de login, melhorando a segurança e reduzindo o risco de ataques como o de força bruta.
- **Gerenciamento de Sessões Seguras:** O Cognito gerencia as sessões dos usuários de maneira eficiente, emitindo tokens de autenticação seguros (JWT), que são usados para acessar diferentes partes do backoffice e da API. Esse sistema de tokens permite que a aplicação valide as permissões do usuário sem expor senhas ou dados sensíveis.
- **Escalabilidade e Confiabilidade:** Sendo um serviço da AWS, o Cognito oferece alta escalabilidade e confiabilidade, garantindo que o sistema possa crescer sem comprometer a segurança. Além disso, conta com autenticação multifator (MFA) e políticas de senha forte, o que agrega camadas extras de proteção.

A implementação do AWS Cognito no backoffice do MamInAgro assegura que apenas usuários autenticados e autorizados possam interagir com as funcionalidades sensíveis da aplicação, mantendo a integridade e a privacidade dos dados, e garantindo que o sistema se mantenha seguro e eficiente à medida que cresce (Amazon Web Services, 2024).

Figura 4.5 – Backoffice - Login

MamInAgro

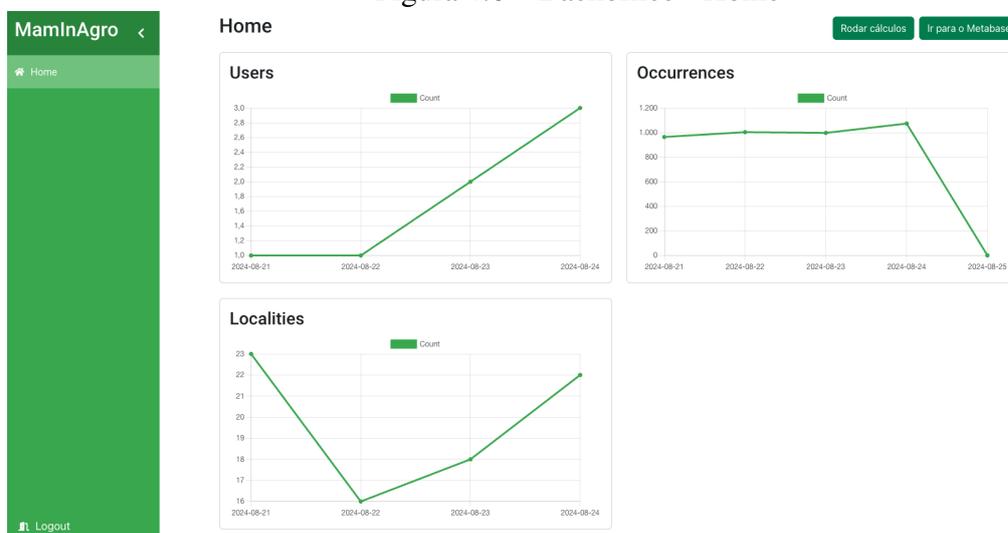
Username

Password

Acessar

Fonte: De autoria própria

Figura 4.6 – Backoffice - Home



Fonte: De autoria própria

4.7 Nova Arquitetura para o Sistema de Cálculos

Com base em todas as novas funcionalidades e as novas características do sistema, temos uma nova arquitetura para todo o processo percorrido na aplicação MamInAgro.

Essa arquitetura é composta por várias etapas interconectadas, garantindo que os dados sejam processados de forma eficiente e estejam corretamente disponíveis para análise.

A arquitetura final segue os seguintes passos:

- **Inserção de Dados:** O usuário insere dados no banco de dados, que são armazenados em suas respectivas tabelas no banco de dados principal.
- **Execução dos Cálculos:** Após uma inserção, o sistema de gerenciamento de processos no backoffice permite que um microserviço em Python, através de uma API Flask, seja acionado para executar os scripts em R. Esses scripts realizam os cálculos necessários e atualizam os resultados nas novas tabelas dentro do esquema *calculations*.
- **Armazenamento dos Resultados:** Os resultados dos cálculos são então armazenados de forma estruturada nas tabelas do esquema *calculations*. Este armazenamento organizado facilita a consulta e a utilização dos dados para análises futuras.
- **Visualização de Dados:** Finalmente, os dados consolidados e atualizados são exibidos na ferramenta de visualização de dados, permitindo uma análise visual e interativa dos resultados.

Essa arquitetura possui capacidade de integrar de maneira eficiente os diferentes componentes do sistema, desde a coleta de dados até a sua análise e visualização. O backoffice desempenha um papel importante nesta arquitetura, ao gerenciar a execução dos cálculos em R por meio do microserviço, oferecendo uma interface centralizada para o controle e monitoramento dos processos.

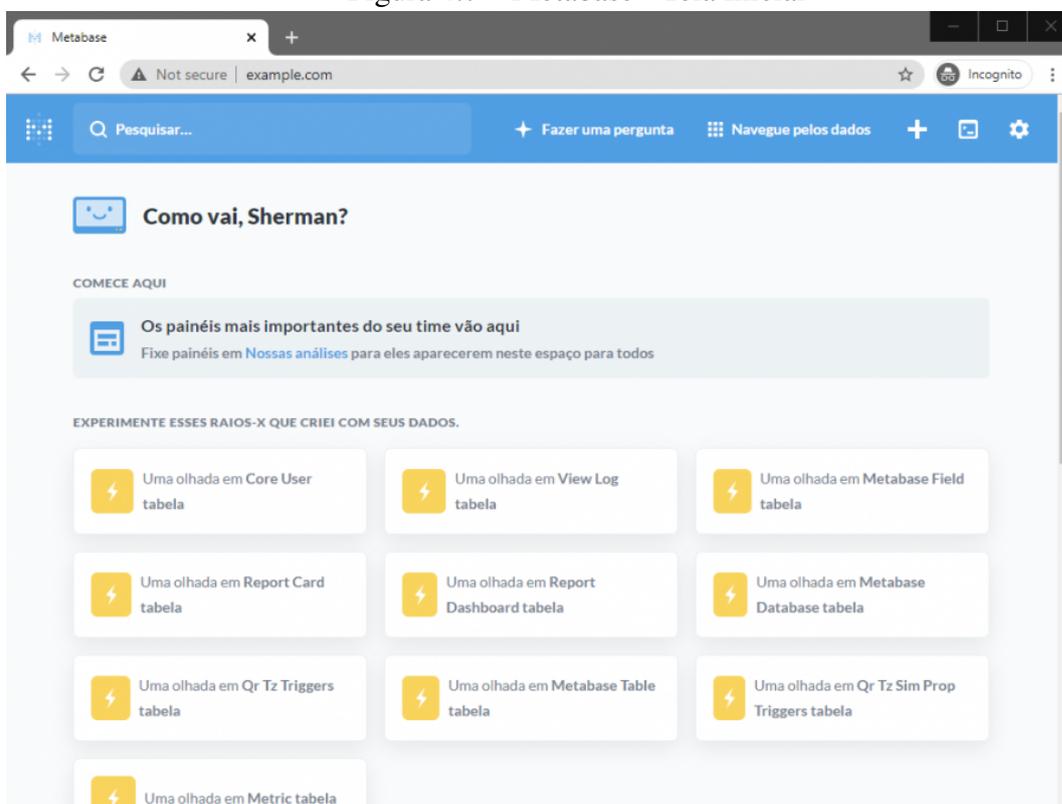
4.8 Implementação de Dashboards no Metabase

Na última etapa do processo temos a criação de dashboards no Metabase que permitem uma visualização interativa e intuitiva dos dados consolidados. Nesta seção, será detalhado o processo de criação dos dashboards e as funcionalidades oferecidas pelo Metabase.

4.8.1 Processo de Criação de Dashboards

Para criar um dashboard no Metabase, é necessário primeiro configurar a conexão com o banco de dados. O Metabase oferece uma interface amigável para essa configuração e realiza algumas análises pré-definidas ao estabelecer a conexão. A Figura 4.7 ilustra a tela inicial do Metabase no primeiro acesso, onde o usuário recebe algumas possíveis análises pré-definidas.

Figura 4.7 – Metabase - Tela Inicial



Fonte: <https://www.letscloud.io/community/como-instalar-o-metabase-no-ubuntu-20-04-com-proxy-reverso-nginx>

4.8.2 Perguntas e SQL Editor no Metabase

O Metabase permite a criação de perguntas diretamente no modelo de banco de dados, utilizando uma interface visual intuitiva. Além disso, oferece um editor SQL completo para consultas mais complexas. A Figura 4.8 mostra o editor SQL do Metabase,

onde os usuários podem escrever e executar scripts SQL diretamente, permitindo consultas avançadas e personalizadas.

- Perguntas no Modelo de Banco de Dados: Permite criar consultas e visualizações simples sem a necessidade de escrever SQL.
- SQL Editor: Para consultas avançadas, o editor SQL permite escrever e executar scripts diretamente no Metabase.

Figura 4.8 – Metabase - SQL Editor

```

1 SELECT u.name AS uf_name,
2     u.abbreviation AS uf_abbreviation,
3     avg(rsl.risk_score) AS avg_risk_score,
4     max(rsl.calculation_date) AS last_calculation_date
5 FROM calculations.risco_especie_localidade rsl
6     JOIN localities l ON rsl.locality_id = l.id
7     JOIN ufs u ON l.uf_id = u.id
8 GROUP BY u.name, u.abbreviation

```

uf_name	uf_abbreviation	avg_risk_score	last_calculation_date
Ceará	CE	1.74	agosto 18, 2024
Piauí	PI	1.74	agosto 18, 2024
Bahia	BA	1.74	agosto 18, 2024
Roraima	RR	1.74	agosto 18, 2024
Rondônia	RO	1.74	agosto 18, 2024
Paraná	PR	1.74	agosto 18, 2024
Sergipe	SE	1.74	agosto 18, 2024
Amazonas	AM	1.74	agosto 18, 2024

Fonte: De autoria própria

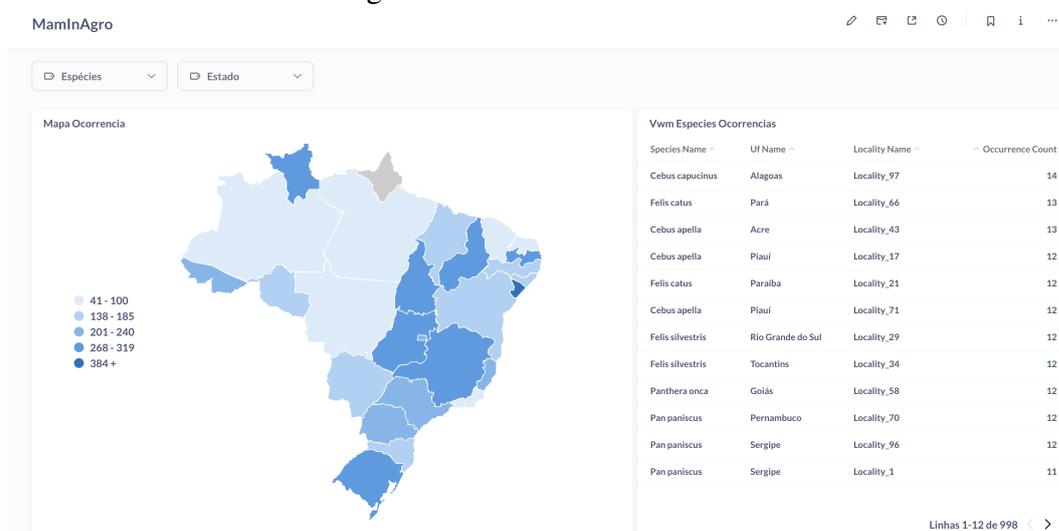
4.8.3 Dashboards para Diferentes Públicos

Optou-se por criar dois dashboards principais:

- Dashboard Interno: Projetado para o pessoal da aplicação MamInAgro manipular e criar seus próprios dashboards.
- Dashboard Público: Criado para expor os dados ao público através de um link público, permitindo acesso a informações relevantes.

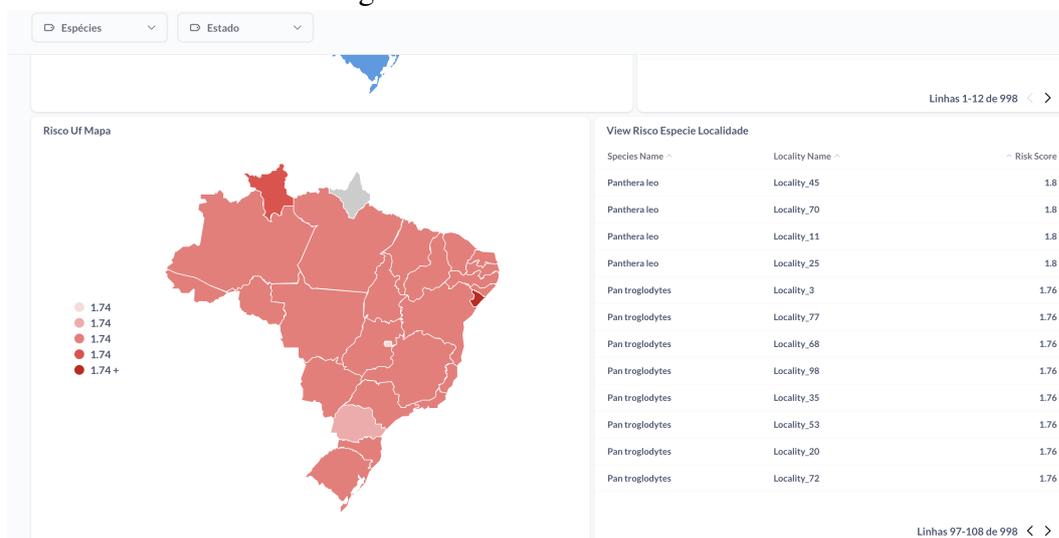
Nas figuras abaixo, podem-se ver exemplos dos dashboards dinâmicos do Metabase. A Figura 4.9 mostra um dashboard interno, enquanto a Figura 4.10 ilustra um dos dashboards públicos.

Figura 4.9 – Metabase - Dashboard 1



Fonte: De autoria própria

Figura 4.10 – Metabase - Dashboard 2



Fonte: De autoria própria

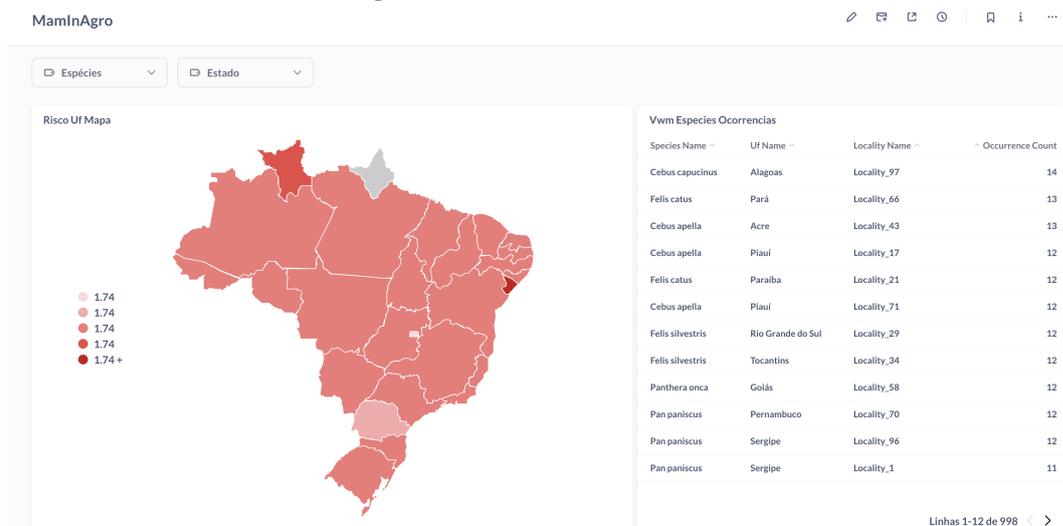
4.8.4 Funcionalidades dos Dashboards

Os dashboards no Metabase incluem várias funcionalidades que facilitam a análise de dados, como:

- **Mapas Interativos:** Utilizados para mostrar ocorrências de espécies por UF.
- **Filtros Dinâmicos:** Permitem aos usuários ajustar as visualizações de acordo com critérios específicos.

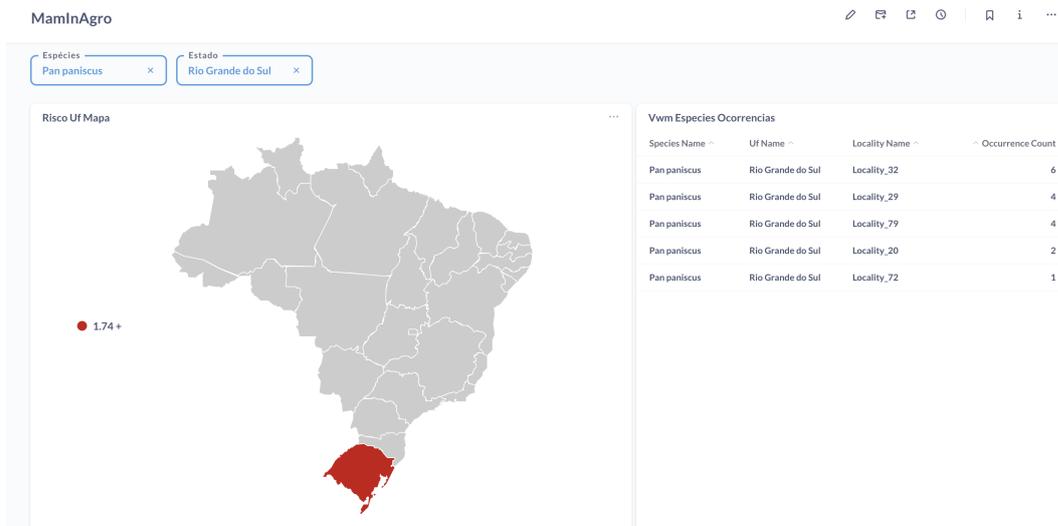
As figuras abaixo exemplificam o uso dessas funcionalidades nos dashboards do Metabase. A Figura 4.11 mostra um dashboard sem a aplicação de filtros, enquanto a Figura 4.12 ilustra o mesmo dashboard após a aplicação de filtros.

Figura 4.11 – Metabase - Sem Filtro



Fonte: De autoria própria

Figura 4.12 – Metabase - Com Filtro



Fonte: De autoria própria

Com base em outras aplicações mencionadas nos trabalhos relacionados, foram implementados dashboards que oferecem uma visão clara e detalhada dos dados, facilitando a tomada de decisões e a análise de informações críticas.

4.8.5 Fluxo Completo do Sistema de Cálculos

A nova arquitetura para o sistema de cálculos implementado no projeto MamInAgro pode ser visualizada através do fluxo completo de dados, desde a inserção na aplicação web até a visualização final dos resultados no Metabase, destacando as principais etapas e componentes envolvidos, juntamente com as tecnologias utilizadas.

O processo descrito pode ser detalhado da seguinte forma:

- **Inserção de Dados:** Os dados são inseridos na aplicação web pelos usuários, utilizando tecnologias de frontend como **React**.
- **Banco de Dados:** Esses dados são então armazenados em um banco de dados **PostgreSQL**, que é organizado para facilitar o acesso e a manipulação.
- **Backoffice em React:** O backoffice em **React** mostra os últimos dados inseridos no banco de dados **PostgreSQL** e permite que os administradores monitorem as operações. Ele também tem a capacidade de chamar manualmente o microserviço em **Flask** para executar os cálculos quando necessário.

- **Microserviço em Flask:** O microserviço desenvolvido em **Flask**, utilizando a linguagem **Python**, é responsável por orquestrar a execução dos cálculos, chamando os scripts necessários.
- **Execução dos Cálculos:** Os cálculos são realizados por scripts escritos em **R**, que processam os dados e geram os resultados necessários.
- **Salvamento dos Resultados:** Os resultados desses cálculos são salvos em tabelas específicas dentro do esquema `calculations` no banco de dados **PostgreSQL**.
- **Visualização no Metabase:** Finalmente, esses resultados são visualizados no **Metabase**, que organiza os dados em dashboards interativos, divididos em:
 - **Dashboards Internos:** Voltados para a equipe interna do projeto, permitindo a manipulação e criação de dashboards personalizados.
 - **Dashboards Públicos:** Criados para exibir dados ao público de forma acessível.

Este fluxo completo serve como uma visão geral do sistema, facilitando a compreensão das etapas envolvidas e a conexão entre os diferentes componentes do processo.

5 AVALIAÇÃO DA USABILIDADE

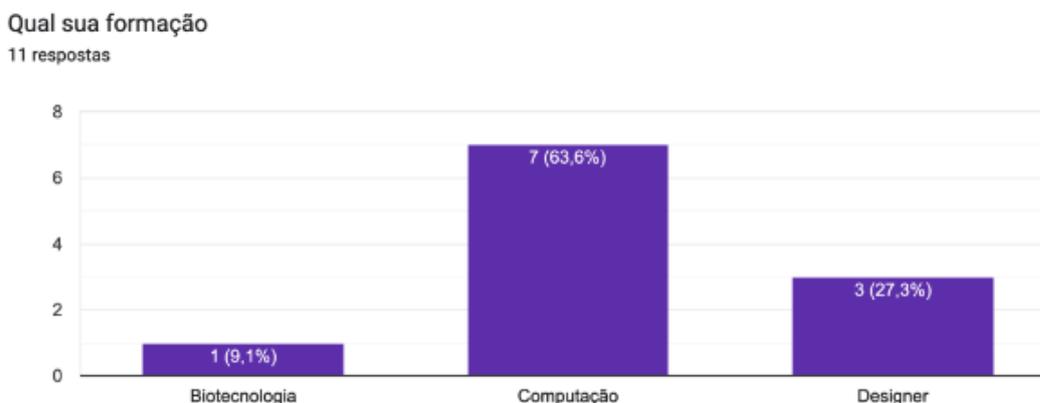
Este capítulo dedica-se a apresentar os testes realizados pelos usuários na plataforma Metabase. O público-alvo da aplicação inclui pessoas de diferentes áreas de formação, que utilizaram o Metabase para visualizar e interagir com dados em dashboards predefinidos. Os testes foram focados na usabilidade da interface, facilidade de navegação, clareza das visualizações, velocidade de resposta, e satisfação geral com a ferramenta.

Os usuários responderam a um questionário online, que foi utilizado como ferramenta para medir o nível de dificuldade e a satisfação dos participantes durante o uso do Metabase. Ao todo, 11 pessoas participaram do experimento.

5.1 Resultados

A faixa etária dos participantes variou de 18 a 44 anos. A área de formação foi diversificada, com 7 participantes da área de Computação, 3 de Design e 1 de Biotecnologia (Figura 5.1). Em relação ao nível de escolaridade, 6 participantes tinham Ensino Superior Completo, 2 estavam com Ensino Superior Incompleto, 2 possuíam Pós-graduação Incompleta, e 1 participante tinha Ensino Médio Incompleto (Figura 5.2).

Figura 5.1 – Área de Formação dos Participantes

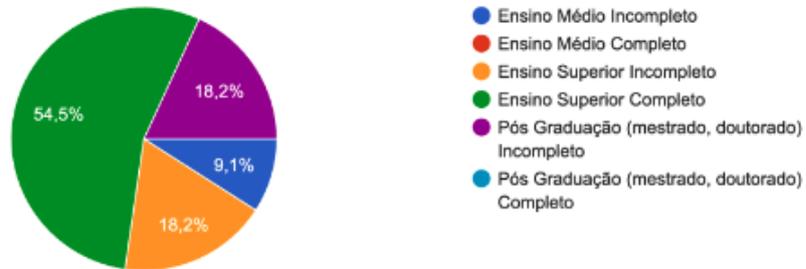


Fonte: Google Forms

Figura 5.2 – Nível de Escolaridade dos Participantes

Qual seu nível de escolaridade?

11 respostas



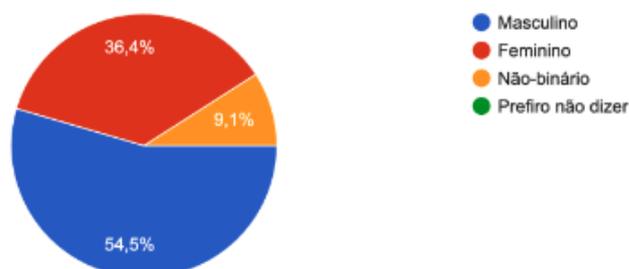
Fonte: Google Forms

Em termos de identidade de gênero, 6 participantes se identificaram como homens, 4 como mulheres, e 1 como não-binário (Figura 5.3). Quando questionados sobre o nível de experiência com internet, 10 participantes avaliaram-se com nota máxima (5), e 1 participante deu nota 4 (Figura 5.4).

Figura 5.3 – Identidade de Gênero dos Participantes

Qual sua identidade de gênero?

11 respostas

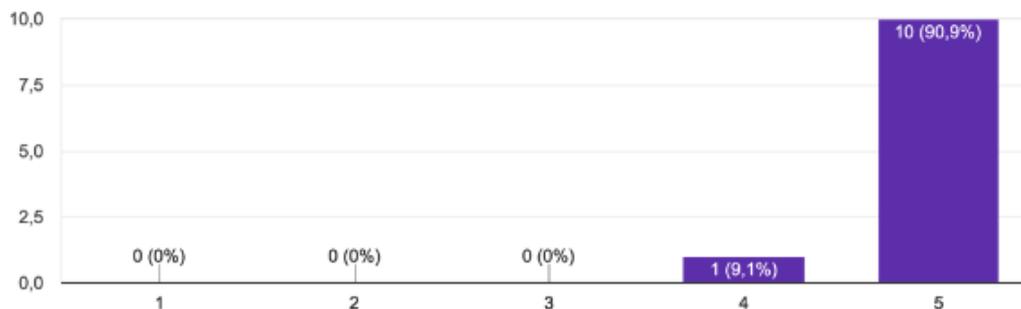


Fonte: Google Forms

Figura 5.4 – Nível de Experiência com Internet dos Participantes

Qual seu nível de experiência com internet?

11 respostas



Fonte: Google Forms

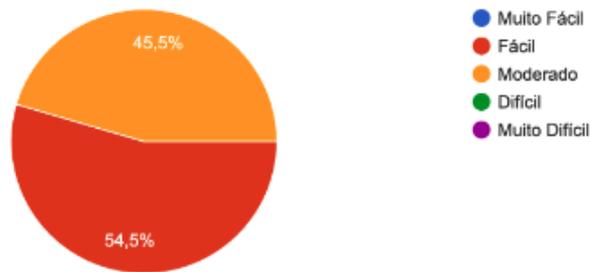
Após a coleta das informações demográficas e da avaliação inicial da usabilidade, os participantes foram convidados a realizar uma série de tarefas práticas utilizando a plataforma Metabase. As atividades propostas incluíam acessar a página do Metabase, recarregar a página, visualizar os gráficos iniciais apresentados e, em seguida, realizar duas operações de filtragem específicas: a primeira consistia em filtrar os dados por uma espécie aleatória, e a segunda em filtrar por uma espécie aleatória e seu estado de origem. Essas tarefas foram projetadas para avaliar a facilidade de navegação, a clareza das visualizações e a velocidade de resposta da plataforma durante o uso real.

Na avaliação da facilidade de entendimento das diferentes visualizações dos dashboards do Metabase, 6 participantes classificaram como "Fácil" e 5 como "Moderado" (Figura 5.5). Quanto à velocidade de resposta ao realizar consultas ou carregar dashboards, 4 participantes consideraram "Rápido" e 7 consideraram "Muito Rápido" (Figura 5.6).

Figura 5.5 – Facilidade de Entendimento das Visualizações

Como você avaliaria a facilidade de entendimento das diferentes visualizações do dashboard do Metabase?

11 respostas

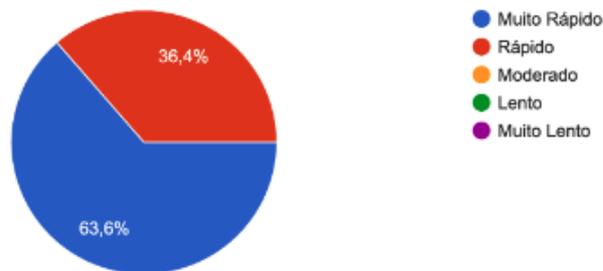


Fonte: Google Forms

Figura 5.6 – Velocidade de Resposta ao Realizar Consultas

Como você avaliaria a velocidade de resposta do Metabase ao realizar consultas ou carregar dashboards?

11 respostas



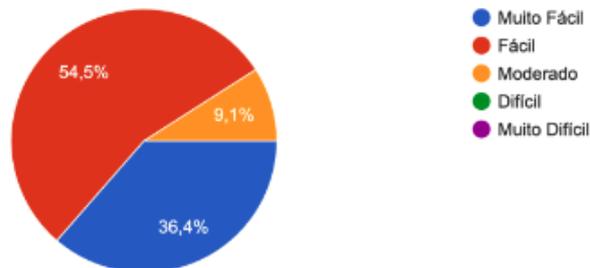
Fonte: Google Forms

Ao serem questionados sobre a facilidade de criar e customizar filtros no Metabase, 1 participante considerou "Moderado", 6 consideraram "Fácil" e 4 avaliaram como "Muito Fácil" (Figura 5.7). Sobre a intuitividade da interface para encontrar e entender as informações nos dashboards, 8 participantes consideraram "Intuitiva", 2 avaliaram como "Moderadamente Intuitiva" e 1 como "Muito Intuitiva" (Figura 5.8).

Figura 5.7 – Facilidade de Criação e Customização de Filtros

Como você avaliaria a facilidade de criar e customizar filtros no Metabase?

11 respostas

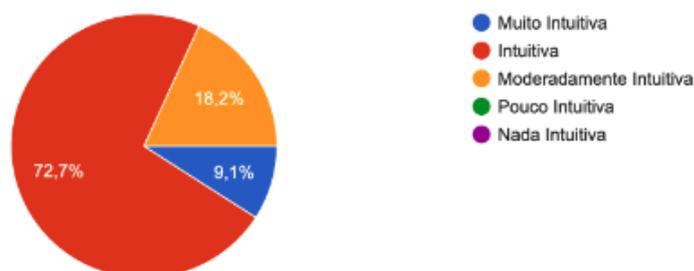


Fonte: Google Forms

Figura 5.8 – Intuitividade da Interface

Quão intuitiva você considera a interface do Metabase para encontrar e entender as informações apresentadas nos dashboards?

11 respostas



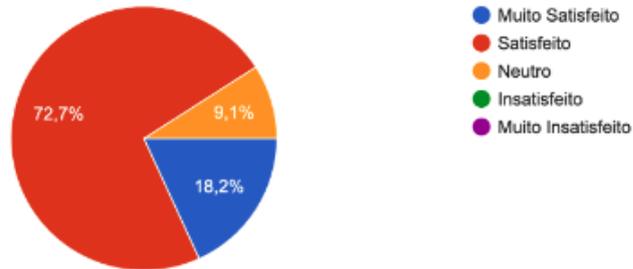
Fonte: Google Forms

Por fim, o nível de satisfação geral com o Metabase como ferramenta de visualização de dados foi positivo: 8 participantes estavam "Satisfeitos", 2 estavam "Muito Satisfeitos", e 1 participante se manteve "Neutro" em relação à ferramenta (Figura 5.9).

Figura 5.9 – Satisfação Geral com o Metabase

Qual é o seu nível de satisfação geral com o Metabase como ferramenta de visualização de dados?

11 respostas



Fonte: Google Forms

Os resultados indicam que, em geral, os usuários consideraram o Metabase uma ferramenta intuitiva, com respostas rápidas às consultas e uma interface fácil de usar. No entanto, a avaliação "Moderado" em algumas áreas sugere que ainda há espaço para melhorias na experiência do usuário, especialmente em termos de personalização de filtros e clareza das visualizações.

6 CONCLUSÃO

Este projeto teve como objetivo principal o desenvolvimento e a implementação de uma solução eficiente para a execução e visualização de cálculos, utilizando uma arquitetura moderna e integrada. Ao longo do trabalho, diferentes abordagens foram exploradas para a realização desses cálculos, sendo decidido manter os scripts de cálculos na linguagem atual, ajustando-os para salvar os valores no banco de dados da aplicação. Em conjunto, foi utilizado um microserviço Flask como a solução mais adequada para orquestrar a execução dos scripts em R.

A escolha do Metabase como ferramenta de Business Intelligence se destacou por sua facilidade de integração com o banco de dados PostgreSQL, seu suporte a views materializadas e sua interface intuitiva, permitindo uma análise de dados mais acessível e eficaz. Esta escolha foi fundamentada na comparação com outras ferramentas de mercado, como Power BI e Apache Superset, que apresentaram limitações em relação às necessidades específicas do projeto.

O projeto também se beneficiou da utilização de um cron job para automatizar a atualização das views materializadas e a execução dos cálculos diariamente, garantindo que os dados permanecessem atualizados para análises subsequentes. Esse componente ajuda a manter a integridade e a relevância dos dados no sistema.

Apesar dos avanços, o projeto enfrenta algumas limitações, especialmente em termos de escalabilidade em sistemas de grande porte. No entanto, essas questões abrem caminhos para futuras melhorias, como a adoção de novas tecnologias que possam oferecer ainda maior suporte a big data, principalmente na forma de entrada de dados no banco de dados, e a possível implementação de técnicas de machine learning para análises preditivas.

O projeto demonstrou a viabilidade de integrar diferentes tecnologias para a execução e visualização de cálculos complexos em um ambiente de dados. As escolhas tecnológicas realizadas possibilitaram a criação de um sistema eficaz e adaptável às necessidades do usuário, oferecendo uma base sólida para futuras expansões e implementações.

REFERÊNCIAS

Amazon Web Services. **AWS Cognito Documentation**. 2024. Disponível em: <<https://docs.aws.amazon.com/cognito/latest/developerguide/what-is-amazon-cognito.html>>.

Apache Superset. **Apache Superset**. 2024. Disponível em: <<https://superset.apache.org/>>.

Atlas of Living Australia. **Atlas of Living Australia**. 2024. Disponível em: <<https://biocache.ala.org.au/>>.

ECODEBATE. Agrotóxicos levam à perda de biodiversidade no mundo. **EcoDebate**, 2023. Acesso em: 22 ago. 2024. Disponível em: <<https://www.ecodebate.com.br/2024/01/12/agrotoxicos-levam-a-perda-de-biodiversidade-no-mundo/>>.

Facebook Inc. **React - A JavaScript library for building user interfaces**. 2024. Disponível em: <<https://reactjs.org/>>.

Fundação Heinrich Böll. Sul global paga caro pelo uso de pesticidas, revela o novo atlas dos agrotóxicos. **Mongabay Brasil**, 2023. Acesso em: 22 ago. 2024. Disponível em: <<https://brasil.mongabay.com/2024/01/sul-global-paga-carro-pelo-uso-de-pesticidas/>>.

Global Biodiversity Information Facility. **GBIF**. 2024. Disponível em: <<https://www.gbif.org/>>.

INC., M. **Metabase: Open Source Business Intelligence Tool**. 2024. Disponível em: <<https://www.metabase.com/>>.

Materialize, Inc. **Why, How, and When To Use Materialized Views**. 2024. Accessed: 2024-08-21. Disponível em: <<https://materialize.com/blog/why-how-and-when-to-use-materialized-views>>.

Matplotlib. **Matplotlib: Visualization with Python**. 2024. Disponível em: <<https://matplotlib.org/>>.

Microsoft Power BI. **Power BI Documentation**. 2024. Disponível em: <<https://powerbi.microsoft.com/>>.

NatureServe Explorer. **NatureServe Explorer**. 2024. Disponível em: <<https://explorer.natureserve.org/>>.

NumPy Developers. **NumPy: The fundamental package for array computing with Python**. 2024. Disponível em: <<https://numpy.org/>>.

Pallets Projects. **Flask Documentation**. 2024. Disponível em: <<https://flask.palletsprojects.com/>>.

Pandas Developers. **Pandas: Python Data Analysis Library**. 2024. Disponível em: <<https://pandas.pydata.org/>>.

Plotly Technologies Inc. **Plotly: Collaborative Data Science**. 2024. Disponível em: <<https://plotly.com/>>.

Python Documentation. **subprocess — Subprocess management**. 2024. Disponível em: <<https://docs.python.org/3/library/subprocess.html>>.

(R-SIG-DB), R. S. I. G. on D. **DBI: R Database Interface**. 2021. Disponível em: <<https://cran.r-project.org/web/packages/DBI/index.html>>.

RPostgreSQL. **RPostgreSQL: R Interface to PostgreSQL**. 2024. Disponível em: <<https://cran.r-project.org/web/packages/RPostgreSQL/index.html>>.

Rpy2 Documentation. **Rpy2 Documentation**. 2024. Disponível em: <<https://rpy2.github.io/>>.

SANTIN, L. **MamInAgro - Um banco de dados para registro de observações de espécies da fauna brasileira**. Monografia (Bacharelado em Engenharia da Computação) — Universidade Federal do Rio Grande do Sul, Instituto de Informática, Porto Alegre, Brasil, 2024. Orientador: Prof. Dr. Renata de Matos Galante.

Seaborn. **Seaborn: Statistical Data Visualization**. 2024. Disponível em: <<https://seaborn.pydata.org/>>.

Sistema de Informação sobre a Biodiversidade Brasileira. **SiBBr**. 2024. Disponível em: <<https://www.sibbr.gov.br/>>.

WILLIAMS, D. R. e. a. Agroecology: protecting, restoring, and promoting biodiversity. **BMC Ecology and Evolution**, 2023. Acesso em: 22 ago. 2024. Disponível em: <<https://bmcecolvol.biomedcentral.com/articles/10.1186/s12862-023-02140-y>>.