

UNIVERSIDADE DE PASSO FUNDO  
INSTITUTO DE CIÊNCIAS EXATAS E GEOCIÊNCIAS  
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO APLICADA

PLATAFORMA DE  
AQUISIÇÃO E  
DISPONIBILIZAÇÃO DE  
DADOS METEOROLÓGICOS

Ronaldo Serpa da Rosa

Dissertação apresentada como requisito parcial  
à obtenção do grau de Mestre em Computação  
Aplicada na Universidade de Passo Fundo.

Orientador: Prof. Dr. Clyde Willian Fraise

Coorientador: Prof. Dr. Willingthon Pavan

Passo Fundo

2017



CIP – Catalogação na Publicação

---

R788p Rosa, Ronaldo Serpa da  
Plataforma de aquisição e disponibilização de dados  
meteorológicos / Ronaldo Serpa da Luz. – 2017.  
71 f. : il. color. ; 30 cm.

Orientador: Prof. Dr. Clyde Willian Fraisse.

Coorientador: Willigthon Pavan.

Dissertação (Mestrado em Computação Aplicada) –  
Universidade de Passo Fundo, 2017.

1. Software. 2. Meteorologia. 3. Estudos climáticos. I. Fraisse,  
Clyde Willian, orientador. II. Pavan, Willingthon, coorientador.  
III. Título.

CDU: 551.50

---

Catalogação: Bibliotecária Marciéli de Oliveira - CRB 10/2113



ATESTADO

Atestamos que **RONALDO SERPA DA ROSA**, matrícula 63974, defendeu, aos trinta dias do mês de março do presente ano, seu Trabalho de Conclusão, com o Título: **“Plataforma de Aquisição e disponibilização de dados Meteorológicos”**. Informamos que o mesmo obteve aprovação e está aguardando os encaminhamentos internos para a confecção do diploma.

Sendo o que tínhamos para o momento, nos colocamos a disposição para esclarecimentos.

Coordenação do Programa de Pós Graduação em Computação Aplicada, do Instituto de Ciências Exatas e Geociências, da Universidade de Passo Fundo, aos trinta dias do mês de março do ano de dois mil e dezessete.



**Prof. Dr. Rafael Rieder**  
Coordenador do Programa de Pós-Graduação  
em Computação Aplicada



À minha filha Carolina que me inspirou e me fez ter forças para concluir este trabalho. A minha esposa Marcele por toda compreensão, carinho e força. A minha mãe Cenira por ter me dado todo o suporte na vida. Ao meu amigo Marciano (*in memoriam*) que infelizmente não conseguiu concluir esta etapa aqui na terra.





## AGRADECIMENTOS

Primeiramente gostaria de agradecer a Deus por ter me dado a vida e saúde para conseguir completar essa etapa.

Gostaria de agradecer o melhor presente que ganhei na minha vida, a minha família. Agradeço a minha esposa Marcele e a minha filha Carolina pela compreensão nas minhas ausências e pelo carinho dedicado principalmente nos momentos mais delicados. Vocês são a razão do meu viver e o motivo pelo qual saio da cama todos os dias.

Agradeço especialmente a minha mãe, Cenira Serpa, minha companheira para todas as horas, inspiradora, batalhadora que tanto lutou e me ensinou. Se hoje consegui chegar até aqui, foi pelo esforço dela.

Agradeço aos professores Clyde Fraise e Willingthon Pavan por acreditar em mim e no meu trabalho, pelo tempo dedicado, pelos ensinamentos, pelas oportunidades de conhecimento e pelo exemplo de professores e pesquisadores.

Agradeço ao meu ex-aluno e amigo Alexandre Caetano da Rocha pelo companheirismo e por trabalhar neste projeto comigo.

Agradeço a UPF por estar presente em toda minha formação acadêmica, pela oportunidade de realizar o sonho de atuar como docente no ensino superior e de descobrir a minha verdadeira vocação.

Agradeço aos colegas do IFRS pelo apoio e pela motivação no desenvolvimento deste trabalho.

Agradeço aos professores e colegas do PPGCA e do grupo Mosaico por todo apoio, conhecimento e coleguismo.

Meus sinceros agradecimentos a todos!



# PLATAFORMA DE AQUISIÇÃO E DISPONIBILIZAÇÃO DE DADOS METEOROLÓGICOS

## RESUMO

A utilização de dados meteorológicos tem sido de fundamental importância para a realização de trabalhos nas mais diversas áreas de estudos. O relacionamento dos elementos meteorológicos com dados específicos da área em estudo, permite gerar informações que facilitam a interpretação de resultados e auxiliam na tomada de decisão. Apesar destes dados estarem, na grande maioria, disponibilizados de forma aberta e gratuita, existem alguns fatores que dificultam a utilização. Neste trabalho é descrito o desenvolvimento de uma plataforma genérica, flexível e escalável, que possibilita o acesso a repositórios de dados meteorológicos, bem como, a aquisição e a disponibilização dos dados. A validação da plataforma foi realizada por meio de dois estudos de caso, onde foram utilizados os repositórios de dados do modelo de previsão do tempo ETA 15Km e da Redemet. Por fim, concluiu-se que a plataforma possui a estrutura necessária para atender os objetivos propostos.

Palavras-Chave: meteorologia, scripts.



# PLATFORM FOR THE ACQUISITION AND AVAILABILITY OF METEOROLOGICAL DATA

## ABSTRACT

The use of meteorological data is of critical importance for the accomplishment of projects in the several areas of study. The relation among meteorological elements with a specific area of study allows the generation of information that facilitates the interpretation of results and helps in decision-making. Even though these data are for the most part available in an open and free format, there are some issues that hinder its use. This study presents the development of a generic, flexible and scalable platform that allows the access to data repositories as well as the acquisition and availability of data. The platform validation was carried out by two case studies, where the data repositories of the ETA 15Km and Redemet. Finally, it was concluded that the platform has the necessary structure to meet the proposed objectives.

Keywords: meteorological, scripts.



## LISTA DE FIGURAS

Figura 1.	Fluxograma das observações meteorológicas realizadas no Brasil[11]. . . . .	26
Figura 2.	Processo utilizado nas requisições de dados implementado no Backend . .	36
Figura 3.	Acoplamento dos scripts ao Backend . . . . .	36
Figura 4.	Diagrama de sequências das interações entre Backend - <i>script</i> - repositórios de dados -Service Layer . . . . .	37
Figura 5.	Arquitetura da plataforma . . . . .	39
Figura 6.	Tela de cadastro dos repositórios de dados . . . . .	40
Figura 7.	Tela de consulta dos repositórios de dados . . . . .	40
Figura 8.	Diagrama de classes do modelo de dados da plataforma - parte 1 . . . . .	41
Figura 9.	Diagrama de classes do modelo de dados da plataforma - parte 2 . . . . .	42
Figura 10.	Estrutura de diretório criada pelo Backend no processamento de cada requisição . . . . .	42
Figura 11.	Fluxo de processamento das requisições . . . . .	43
Figura 12.	Diagrama de máquina de estados das requisições . . . . .	43
Figura 13.	Arquivo disponibilizado em um dos repositórios do modelo ETA 15KM . .	47
Figura 14.	Organização e nomenclatura das pastas do repositório ETA 15KM . . . . .	48
Figura 15.	Nomenclatura e exemplo de arquivos do repositório ETA 15KM . . . . .	48
Figura 16.	Tela de cadastro dos dados básicos do repositório Modelo ETA 15KM . .	49
Figura 17.	Tela de cadastro dos <i>scripts</i> de <i>download</i> e processamento do repositório Modelo ETA 15KM . . . . .	50
Figura 18.	Tela de cadastro das variáveis disponibilizadas pelo repositório Modelo ETA 15KM . . . . .	50
Figura 19.	Tela de cadastro das cidades de cobertura do repositório Modelo ETA 15KM . . . . .	50
Figura 20.	Tela de cadastro dos formatos de saída do repositório Modelo ETA 15KM	51
Figura 21.	Tela de pesquisa de repositório de dados disponíveis para a cidade de Passo Fundo . . . . .	51
Figura 22.	Tela de informação dos parâmetros da requisição para o repositório do Modelo ETA 15KM . . . . .	52
Figura 23.	Tela de acompanhamento das requisições . . . . .	52
Figura 24.	Arquivo de saída da requisição . . . . .	52
Figura 25.	Exemplo de mensagem METAR . . . . .	56
Figura 26.	Arquivo contendo as mensagens METAR da requisição do usuário . . . . .	57

Figura 27.	Tela de cadastro dos dados básicos do repositório Redemet . . . . .	58
Figura 28.	Tela de cadastro dos <i>scripts</i> de <i>download</i> e processamento . . . . .	58
Figura 29.	Tela de cadastro das variáveis disponibilizadas pelo repositório . . . . .	58
Figura 30.	Tela de cadastro das cidades de disponibilização do repositório . . . . .	59
Figura 31.	Tela de cadastro dos formatos de saída . . . . .	59
Figura 32.	Tela de pesquisa de repositório de dados disponíveis para a cidade de Passo Fundo . . . . .	59
Figura 33.	Tela de informação dos parâmetros da requisição . . . . .	60
Figura 34.	Tela de acompanhamento das requisições . . . . .	60
Figura 35.	Arquivo de saída da requisição . . . . .	63



## LISTA DE SIGLAS

NOAA – National Oceanic and Atmospheric Administration  
CPC – Climate Prediction Center  
NASA – National Aeronautics and Space Administration  
RTMA – Real-Time Mesoscale Analysis  
INMET – Instituto Nacional de Meteorologia  
CPTEC – Centro de Previsão de Tempo e Estudos Climáticos  
ANA – Agência Nacional de Águas  
IAG/USP – Instituto Astronômico e Geofísico da Universidade de São Paulo  
OMM – Organização Meteorológica Mundial  
FTP – File Transfer Protocol  
ABNT – Associação Brasileira de Normas Técnicas  
UPF – Universidade de Passo Fundo  
BAD – Biblioteca de Acesso a Dados  
CS – Camada de Serviço  
JSF – Java Server Faces  
MVC – Model View Controller  
API – Application Programming Interface  
ORM – *Object Relational Mapping*  
JPA – *Java Persistence API*  
REDEMET – Rede de Meteorologia da Aeronáutica  
METAR – Meteorological Aerodrome Report



# SUMÁRIO

1	INTRODUÇÃO .....	21
2	REVISÃO DE LITERATURA .....	23
2.1	UTILIZAÇÃO DOS DADOS METEOROLÓGICOS .....	23
2.2	DADOS METEOROLÓGICOS: CONCEITOS BÁSICOS E DISPONIBILIZAÇÃO ..	24
2.3	SOLUÇÕES DE ACESSO AOS DADOS METEOROLÓGICOS .....	28
3	PLATAFORMA DE AQUISIÇÃO E DISPONIBILIZAÇÃO DE DADOS METEROLÓGICOS.....	33
3.1	RESUMO .....	33
3.2	INTRODUÇÃO .....	33
3.3	DESENVOLVIMENTO.....	34
3.3.1	Frontend .....	34
3.3.2	Backend .....	35
3.3.3	Database Access Library .....	38
3.3.4	Service Layer .....	38
3.4	RESULTADOS E DISCUSSÃO .....	39
3.5	CONSIDERAÇÕES FINAIS .....	44
4	ESTUDO DE CASO: REPOSITÓRIO DE DADOS DE PREVISÃO MO- DELO ETA 15KM .....	47
5	ESTUDO DE CASO: REPOSITÓRIO DE DADOS OBSERVADOS RE- DEMET .....	55
6	CONSIDERAÇÕES FINAIS .....	65
7	TRABALHOS FUTUROS.....	67
	REFERÊNCIAS .....	69



# 1. INTRODUÇÃO

Há milhares de anos o homem percebeu que de alguma maneira o tempo e o clima exercem forte influência sobre a vida humana. Desta forma, passou a observar, estudar e registrar a influência do tempo e do clima para entender os fenômenos da natureza e orientar ações no seu habitat. Com a evolução e a criação de instrumentos de medição, as observações e os registros foram sendo aprimorados dando um caráter mais quantitativo e regular[1].

Nas últimas décadas houve um interesse acentuado pela climatologia no mundo, principalmente devido aos fenômenos meteorológicos que vem acontecendo e a preocupação com a sustentabilidade na geração de alimentos. Isto pode ser comprovado pela criação de diversos órgãos governamentais especializados em registrar, monitorar e analisar as condições de tempo e clima. Estes órgãos prestam diversos serviços relevantes para a comunidade, destacando-se, principalmente no meio acadêmico e em centros de pesquisa, a disponibilização das observações históricas e de previsões do tempo e clima.

O acesso aos dados meteorológicos têm sido de fundamental importância para a realização de muitos trabalhos que avaliam o impacto das variações climáticas nas mais diversas áreas, tais como agrícola, saúde pública, energia elétrica, desastres naturais, transportes aéreos, entre outras. A partir do relacionamento dos elementos meteorológicos com dados específicos da área em estudo é possível gerar informações que facilitam a interpretação de resultados e auxiliam na tomada de decisão.

Atualmente existe uma grande quantidade de repositórios de dados meteorológicos disponíveis na rede mundial de computadores, principalmente disponibilizados por órgãos governamentais de meteorologia, porém acesso aos mesmos não é uma tarefa trivial. Um dos fatores dificultantes é a localização dos repositórios de dados. Apesar destes estarem, na grande maioria, disponibilizados de forma aberta e gratuita, por vezes localizar o repositório que contenha os dados de interesse do usuário é uma dificuldade.

Outro fator é a falta de padronização dos repositórios, pois cada instituição organiza e disponibiliza os dados conforme lhe convém. Além disso, não existe uma padronização no formato dos arquivos disponibilizados, sendo necessário algumas vezes a utilização de ferramentas específicas para a manipulação dos mesmos, com é o caso dos arquivos em formato de grade.

De acordo com Almeida et al.[2] sistema integrado de dados meteorológicos para aquisição, armazenamento e visualização de dados é de interesse dos centros operacionais de previsão de tempo e também pode beneficiar universidades, em apoio as atividades de ensino e pesquisa.

Tendo em vista as dificuldades expostas e a importância da utilização dos dados meteorológicos, este trabalho visa apresentar o desenvolvimento de uma plataforma genérica, flexível e escalável que possibilite o acesso de forma aos repositórios de dados meteorológicos.

Além de localizar os repositórios de dados, a plataforma também possibilita o acoplamento de *scripts* de *download* e processamento de dados. Isto permite que o sistema acesse os repositó-

rios, efetue o download e o processamento automático dos dados, disponibilizando-os em diversos formatos para o usuário final.

Este trabalho está organizado da seguinte forma: No capítulo 2, é apresentada a revisão de literatura, onde são abordados alguns trabalhos sobre a utilização dos dados meteorológicos, bem como a alguns aspectos básicos sobre os dados meteorológicos e algumas soluções de acesso aos dados meteorológicos existentes. No Capítulo 3, é apresenta em formato de artigo científico a plataforma de aquisição e disponibilização de dados meteorológicos. No Capítulo 4 é apresentado um estudo de caso utilizando o modelo de previsão numérica ETA 15KM. Em seguida, no Capítulo 5 é apresentado um segundo estudo de caso utilizando o repositório de dados da Redemet. As considerações são apresentadas no Capítulo 6 e as conclusões no Capítulo 8.

## 2. REVISÃO DE LITERATURA

Neste capítulo serão abordados alguns trabalhos que foram relevantes tanto para a fundamentação teórica, quanto para o entendimento de alguns conceitos básicos e para a definição da arquitetura e funcionamento da plataforma. Neste sentido, inicialmente é apresentada uma contextualização sobre a utilização dos dados meteorológicos na mais diversas áreas da sociedade. Posteriormente são apresentados alguns aspectos sobre os dados meteorológicos, especialmente, a disponibilização, bem como são abordadas algumas das principais instituições de monitoramento e pesquisa de tempo e clima e os respectivos serviços oferecidos. Por fim, são apresentadas algumas soluções de acesso aos dados meteorológicos, bem como a descrição do funcionamento de cada uma das soluções.

### 2.1 UTILIZAÇÃO DOS DADOS METEOROLÓGICOS

Um dos grandes desafios da humanidade no século 21 são os efeitos das variações climáticas causados pelo aumento dos níveis de dióxido de carbono e outros gases de efeito estufa na atmosfera [3]. Cada vez mais, surge a necessidade de realizar estudos para avaliar o impacto destas variações nas mais diversas áreas, tais como saúde pública, construção civil, energia elétrica, meio ambiente, desastres naturais, indústria, comércio entre outras.

A agricultura é uma das áreas da economia mais influenciadas pelas variações de tempo e clima, pois os elementos meteorológicos além de influenciar nos processos metabólicos das plantas, afetam também as mais diversas atividades no campo, bem como a relação das plantas com microorganismos, insetos, fungos e bactérias, que podem favorecer a ocorrência de pragas e doenças. Além disso, deve-se considerar também que grande parte das práticas agrícolas de campo dependem das condições de tempo e da umidade no solo para que possam ser realizadas de forma eficiente[4].

De acordo com Bergamaschi e Matzenauer[5], os elementos meteorológicos são os principais responsáveis pelas oscilações e frustrações das safras agrícolas em todo o Brasil. Diversos estudos de produção agrícola indicam alta relação entre as variações das safras das principais culturas com as condições meteorológicas e climáticas.

Tendo em vista a importância no clima na produção agrícola, o uso de informações meteorológicas e climáticas são fundamentais para a realização de estudos e pesquisas que visam tornar a agricultura uma atividade sustentável[4]. Estas informações são essenciais para auxiliar em diversas atividades, tais como planejamento de cultivos, escolha da melhor condição para a execução de diferentes práticas agrícolas, auxílio a tomada de decisão e principalmente na resiliência do sistema agrícola quanto as variações meteorológicas [6].

Neste contexto, muitos trabalhos têm sido realizados, como por exemplo o trabalho realizado por Bergamaschi e Matzenauer[5], os quais avaliaram a relação entre o rendimento de milho e as variáveis hídricas. A avaliação foi realizada utilizando a precipitação pluvial, radiação solar global,

temperatura média do ar, umidade relativa do ar, velocidade do vento e a partir destes foi estimada a evapotranspiração de referência. Por fim, os autores identificaram que o rendimento da cultura do milho pode ser estimado a partir das variáveis hídricas e que o consumo relativo de água e a deficiência hídrica são as variáveis mais eficientes para indicar as variações de rendimento de grãos de milho no estado do Rio Grande do Sul.

Outro trabalho relevante foi realizado por Da Cunha et al.[7], no qual o autor analisou o risco climático para a cultura de trigo no Rio Grande do Sul. No desenvolvimento do trabalho o Da Cunha et al. utilizou técnicas de modelagem e simulação de culturas e ferramentas de geoprocessamento alimentados pelos dados meteorológicos de 36 localidades do estado do Rio Grande Sul, com séries históricas de observações meteorológicas diárias, entre 20 e 30 anos. Com isto o Da Cunha et al. conseguiu identificar áreas com menor chance de riscos à cultura de trigo, tendo como referência a época de semeadura[7].

Além das diversas aplicações na área agrícola, os dados meteorológicos têm se mostrado úteis em outras áreas, como se pode observar no trabalho desenvolvido por Westphal et al.[8]. O autor desenvolveu uma metodologia para estimativa de consumo de energia elétrica de edificações não residenciais climatizadas utilizando dados mensais de temperatura, pressão, nebulosidade e umidade relativa do período de 1961 a 1990[8].

Na área da saúde pública Limeira[9] analisou a influência dos elementos meteorológicos na incidência da dengue, meningite e pneumonia. Foram utilizadas as temperaturas médias, máximas e mínimas, umidade relativa do ar e precipitação do ano de 1992 até 2000 de João Pessoa/PB [9]. Com o estudo, os autores conseguiram identificar a relação da temperatura máxima e da precipitação com a predição da incidência das doenças em João Pessoa.

Relacionando dados diários de umidade relativa do ar, precipitação, número consecutivo de dias sem chuva com as ocorrências de incêndios florestais, Soares[10] conseguiu determinar um índice de risco de incêndio diário denominado Índice de Monte Alegre ou também encontrado como Fórmula de Monte Alegre. O índice tem sido usado por diversas empresas e instituições florestais para estimar o grau de perigo de incêndio e ajudar non planejamento das atividades de prevenção e combate[10].

## 2.2 DADOS METEOROLÓGICOS: CONCEITOS BÁSICOS E DISPONIBILIZAÇÃO

Os dados meteorológicos são um conjunto de observações relacionadas aos elementos meteorológicos de uma localidade em um determinado momento, originados de observações por instrumentos meteorológicos e modelos numéricos. Entre os diversos elementos meteorológicos observados ou de previsão, os mais comuns são: temperatura máxima e mínima, umidade relativa do ar, direção e velocidade do vento, precipitação, pressão atmosférica, cobertura de nuvens, radiação entre outros.

De acordo com Torres[11], *“uma observação meteorológica de superfície consiste na medição ou determinação de todos os elementos que, em seu conjunto, representam as condições*



*meteorológicas em um dado momento e em um determinado lugar*”. Estas observações são fundamentais para o conhecimento do clima e para a investigação das leis gerais que regem fenômenos. O registro das observações meteorológicas vem sendo realizado há milhares de anos, no entanto, no Brasil as primeiras séries temporais conhecidas foram registradas nos séculos XVIII e XIX. A partir do século XIX, o Brasil passou a utilizar instrumentos para medir os eventos atmosféricos mais comuns[1].

Ainda no século XIX, um importante marco na meteorologia brasileira foi a criação do Imperial Observatório do Rio de Janeiro, cujo objetivo inicial era realizar observações meteorológicas para orientar os alunos das escolas militares de terra e mar. Nesta mesma época navios hidrográficos da Marinha brasileira também realizavam observações meteorológicas regulares, o que deu origem a Repartição Central Meteorológica da Marinha[1].

Após algumas mudanças ocorridas nas instituições no decorrer dos anos, em 1909 foi realizada a unificação das atividades do Observatório com as redes de observação da Marinha e também do Telégrafo Nacional, os quais deram origem ao Serviço Meteorológico Nacional, atualmente conhecido como INMET (Instituto Nacional de Meteorologia), vinculado ao Ministério da Agricultura, Pecuária e Abastecimento[1].

O INMET é responsável pela coordenação e desenvolvimento das atividades meteorológicas no país, sendo o representante do Brasil na OMM (Organização Meteorológica Mundial), sendo esta a entidade responsável pela coordenação das atividades meteorológicas no mundo.

Atualmente o objetivo principal do INMET é prover informações meteorológicas à sociedade brasileira para auxiliar no processo de tomada de decisão, contribuindo para o desenvolvimento sustentável do País[12], já as observações dos Ministérios da Aeronáutica e da Marinha são direcionadas à navegação área e marítima[11].

Entre as principais atribuições do INMET destaca-se a elaboração e divulgação diária da previsão do tempo, avisos e boletins meteorológicos especiais; promover a execução de estudos e levantamentos meteorológicos e climatológicos aplicados a agricultura e outras atividades correlatas; estabelecer, coordenar e operar as redes de observações meteorológicas e de transmissão de dados, inclusive aquelas integradas à rede internacional[12].

A estrutura do INMET é composta por mais de 400 estações terminais, onde as observações são coletadas e enviadas para os centros e subcentros coletores. No caso dos subcentros, estes compilam as mensagens em boletins ou coletivos parciais e transmitem para os centros coletores. Então, os centros coletores reúnem os dados coletivos parciais em forma de coletivos territoriais e transmitem para o Centro Regional de Telecomunicações de Brasília, conforme pode se visualizar na Figura 1.

O Centro Regional de Telecomunicações de Brasília também recebe e organiza os dados de outros países da América do Sul e transmite ao Centro Meteorológico Mundial de Washington, que por sua vez, troca informações outros centros meteorológicos mundiais[13].

As estações meteorológicas ou estações terminais são responsáveis por registrar grande parte das observações meteorológicas no mundo, sendo classificadas de acordo com a respectiva

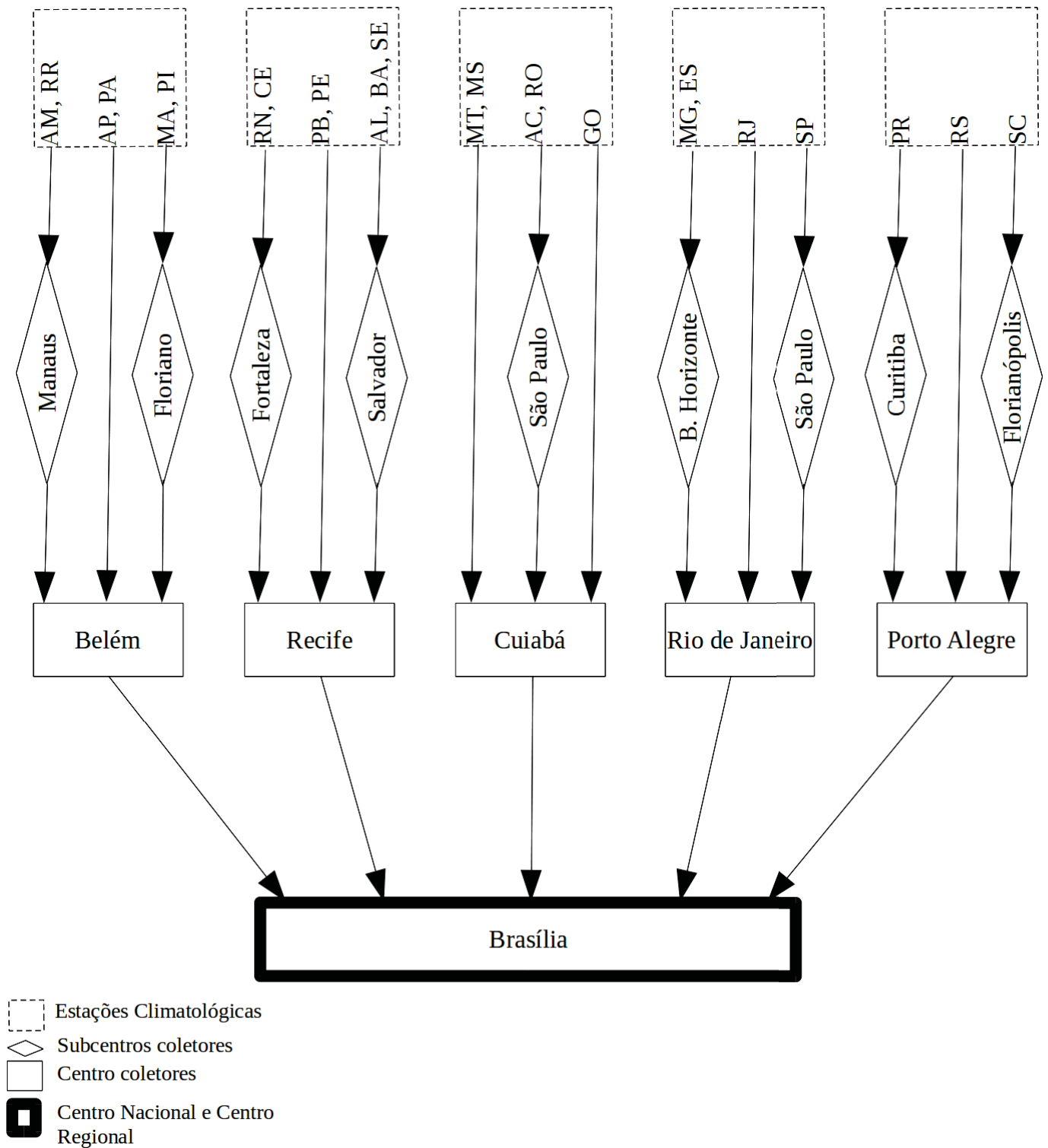


Figura 1. Fluxograma das observações meteorológicas realizadas no Brasil[11].

finalidade. As estações sinóticas são utilizadas para auxiliar na previsão do tempo e coletam dados de direção e velocidade do vento, temperatura do ar, umidade relativa do ar, chuva, pressão atmosférica, nuvens, geadas. As estações climatológicas são utilizadas para determinar o clima de uma região, após um histórico de no mínimo 30 anos de observação, e coletam dados de direção e velocidade do vento, temperatura do ar, umidade relativa do ar, chuva, pressão atmosférica, nuvens, geadas,

temperatura do solo, evapotranspiração, orvalho, evaporação e radiação solar. Em ambas as estações as leituras são realizadas as 9 horas, 15 horas e as 21 horas[11].

Para uso específico na agricultura são utilizadas as estações agroclimatológicas, este tipo de estação fornece informações para estudos sobre a influência dos elementos meteorológicos sobre as culturas e também realiza observações que estimam o desenvolvimento das culturas[11].

Além das estações meteorológicas, aviões comerciais, bóias meteorológicas, navios, balões meteorológicos, imagens de satélites e radares também registram dados das observações meteorológicas no mundo, muitas vezes cobrindo áreas que as estações não conseguem alcançar.

As observações meteorológicas também são essenciais para a confecção de cartas de previsão do tempo, geradas com a ajuda de modelos físico-matemáticos de alta resolução que simulam o comportamento futuro da atmosfera e permitem que sejam realizadas previsões do tempo com dias de antecedência[12].

No Brasil o Centro de Previsão de Tempo e Estudos Climáticos (CPTEC) é órgão responsável pelos assuntos relacionados a previsão numérica de tempo e clima. O CPTEC fornece previsões do tempo de curto e médio prazo, além de previsões climáticas de alta precisão, por meio de técnicas de modelagem numérica altamente complexas da atmosfera e oceanos. As previsões são realizadas com o uso de supercomputadores e do conhecimento de profissionais especializados[14].

A Agência Nacional de Águas (ANA) é o órgão brasileiro responsável pela regulação, gestão e monitoramento de recursos hídricos de rios e reservatórios. A agência realiza o monitoramento hidrometeorológico operando cerca de 4.543 estações, com o objetivo de estimar o volume de chuvas, a evaporação da água, o nível e a vazão dos rios, a quantidade de sedimentos e a qualidade das águas em estações respectivamente relacionadas: pluviométricas, evaporimétricas, fluviométricas, sedimentométricas e de qualidade da água. Os dados coletados pelas estações são utilizados para realização de estudos, definição de políticas públicas, avaliação da disponibilidade hídrica e para monitoramento de eventos críticos, tais como cheias e estiagens, entre outros assuntos de interesse da sociedade[15].

A nível mundial também existem diversos órgãos que tratam sobre assuntos de meteorologia e recursos naturais, como por exemplo o NOAA (National Oceanic and Atmospheric Administration) que é um órgão do governo americano, ligado ao Departamento de Comércio, responsável por realizar pesquisas com o intuito de melhorar a compreensão do meio ambiente e também por monitorar assuntos de meteorologia, oceanos, atmosfera e clima. A NASA (National Aeronautics and Space Administration), outro órgão do governo americano, estuda e monitora a distribuição e variabilidade da precipitação dentro dos trópicos por meio do satélite TRMM (Tropical Rainfall Measuring Mission)[3].

As instituições além de realizar estudos, pesquisas e monitorar o clima e o tempo também prestam um importante serviço, principalmente para a comunidade acadêmica e para centros de pesquisas, que é a disponibilização dos dados coletados nas observações, o que resulta em uma importante e grande massa de dados meteorológicos.

O INMET, por exemplo, disponibiliza os dados da rede de estações meteorológicas por meio de uma ferramenta disponível no web site, onde é possível ter acesso a dados como: temperatura, umidade, ponto de orvalho, pressão, vento, radiação e chuva de diversas regiões do Brasil dos últimos 365 dias. Os dados históricos podem ser acessados por meio Banco de Dados Meteorológicos para Ensino e Pesquisa (BDMEP)[12].

O CPTEC também disponibiliza um grande volume de dados sobre previsões de tempo numérica a nível regional, que varia entre períodos de 5, 7 e 11 dias e entre resoluções espaciais de 15x15 km, 5x5 km, 40x40km e global que varia entre períodos de 7 e 15 dias e entre resoluções espaciais de 20x20 km e 100x100km[14].

O NOAA disponibiliza no web site a instituição uma ferramenta para os usuários consultar dados meteorológicos observados de temperatura e precipitação desde 1971 a 2000 de mais de 9.800 estações meteorológicas dos Estados Unidos. Os dados são disponibilizados em séries temporais anuais, mensais, diárias e horárias, considerando a média dos últimos 30 anos[16].

A disponibilização dos dados das observações e de previsão numérica, na grande maioria, são realizadas diariamente em endereços FTP (File Transfer Protocol) e/ou em sistemas específicos das instituições de forma aberta e gratuita para utilização da sociedade em geral. Em alguns casos, a utilização dos dados deve obedecer às políticas de uso estabelecidas pela instituição fornecedora.

### 2.3 SOLUÇÕES DE ACESSO AOS DADOS METEOROLÓGICOS

A utilização dos dados meteorológicos em estudos e pesquisas gerou a necessidade de soluções que facilitem o acesso aos mesmos de forma simples e eficiente. Neste sentido, rOpenSci[17] motivado pela frequente utilização da linguagem R[18] em trabalhos utilizando dados meteorológicos desenvolveu um pacote denominado “RNOAA”. O pacote possibilita o acesso aos dados meteorológicos provenientes do NOAA, tais como temperatura, precipitação, cobertura de gelo.

Para alimentar o sistema de previsão de doenças na web, Kang et al.[19] desenvolveu uma solução de acesso aos dados meteorológicos. O objetivo do sistema desenvolvido pelo autor é gerar notificações em determinados períodos de tempo, na resolução espacial 240m x 240m com base em dados meteorológicos [19].

Para isto, inicialmente o sistema de aquisição de dados meteorológicos acessa os dados das redes de estações meteorológicas normais e de desastres e armazena estes no sistema de armazenamento de dados. Posteriormente o sistema de processamento de trabalhos busca os dados meteorológicos no sistema de armazenamento e insere na entrada dos modelos de previsão de doenças. Os dados gerados como saída dos modelos são armazenados no sistema de armazenamento e disponibilizados para os sistemas de processamento de trabalhos e ao serviço web[19].

Diversos serviços também permitem o acesso aos dados meteorológicos de vários locais do mundo, como por exemplo o OpenWeatherMap. O serviço provê dados meteorológicos para mais de 200.000 cidades por meio de uma interface de geolocalização disponibilizada no website ou por meio de Web Service, onde é possível acessar dados observados e de previsão, como por exemplo,

de velocidade do vento, de cobertura de nuvens e outros. A utilização do serviço possui um custo financeiro ou com limitações de uso [20].

A Embrapa (Empresa Brasileira de Pesquisas Agropecuárias) desenvolveu um sistema para fornecer dados meteorológicos e agrometeorológicos de diversos municípios e estados brasileiros. O Agritempo é um sistema de Monitoramento Agrometeorológico alimentado por várias fontes de dados meteorológicos, principalmente pelos dados das estações do INMET, que passam por um processo de migração antes de serem disponibilizados[21].

Após o processamento dos dados o sistema disponibiliza produtos, como por exemplo, boletins agrometeorológicos, mapas de monitoramento climático e de previsão, mapas de índice de seca (SPI), sumários e mapas de séries históricas, zoneamento agrícola, mapas de probabilidade de precipitação, mapas de solos e outros. Além de disponibilizar estes produtos o sistema também possibilita a opção de exportação dos dados meteorológicos em formatos de fácil manipulação.

Devido à heterogeneidade dos formatos de dados das fontes que alimentam o sistema Agritempo, [22] desenvolveu um processo denominado migração de dados, o qual é composto pelas seguintes atividades: (i) coleta do dado; (ii) conversão do dado para um formato padrão; (iii) simulação dos dados faltantes e correção dos dados com problemas; (iv) envio dos dados; (v) inclusão do dado na base de dados. Utilizando este processo o autor automatizou grande parte das atividades de importação de dados, o que resultou na redução de erros nos dados, aumentando a confiabilidade e a qualidade dos produtos gerados pelo sistema Agritempo.

Os centros de pesquisas meteorológicas também são grande consumidores de dados meteorológicos, sendo estes necessários tanto na entrada dos modelos numéricos de previsão do tempo quanto para a análise do clima e do tempo. Neste sentido diversas soluções de acesso, tratamento e análise dos dados meteorológicos foram desenvolvidas com o intuito de suprir estas necessidades.

O Metview é um sistema de acesso, processamento e visualização de dados meteorológicos, voltado para meteorologista e para pesquisa operacional. O sistema foi projetado para suportar diversos formatos, tais como BUFR, GRID, matrizes em ASCII e geopoints. Os dados são oriundos de observações, imagens de satélites meteorológicos e prognósticos de previsão do tempo e são acessados de forma transparente para os usuários [23].

Uma das fontes de dados que alimenta o sistema MetView é o Banco de Dados Meteorológico (BDM). O projeto BDM é uma parceria do CPTEC e do INPE com o objetivo de centralizar os dados meteorológicos para utilização em trabalhos operacionais de previsão do tempo e estudos climáticos [24].

O BDM é alimentado por sistemas que recuperam os dados de observações meteorológicas que enviadas regularmente ao CPTEC pela rede Global Telecommunication System (GTS) da Organização Meteorológica Mundial (OMM), por intermédio do INMET. Os dados são processados por outro sistema, denominado pré-processamento, no qual é efetuado o controle de qualidade dos dados por meio de um índice de confiabilidade associado a cada uma das variáveis climáticas [24].

O banco de dados armazena três tipos de dados: primários, associativos e descritivos. Os dados primários correspondem aos dados de observações, sendo estes os de maior volume no

banco de dados. Os dados associativos são utilizados como índices dos dados primários, os quais descrevem as coberturas temporais, geográfica, local de armazenamento e outras características dos dados primários para facilitar a pesquisa[24].

Os descritivos são dados relativos as estações meteorológicas, metadados, e outros dados estáticos necessários para o funcionamento do sistema. Outros sistemas também estão incorporados ao ambiente do BDM, como por exemplo, o sistema de interface interativa. Por meio deste sistema é possível manter os dados de apoio e os metadados do sistema, assim como, a recuperação e exportação de dados observacionais[24].

Ainda no que se refere a soluções de acesso aos dados em centros de pesquisa meteorológicas, outra solução encontrada é o Local Data Manage (LDM), projeto coordenado pelo Unidata Program Center (UPC/UCAR). O objetivo do projeto é disponibilizar um sistema descentralizado (rede ponto a ponto), onde cada nó da rede recebe e repassa dos dados meteorológicos[25].

Por meio do sistema é possível acessar os dados convencionais de estações meteorológicas, boias, navios, aviões, radiossondas, resultados de modelos numéricos, imagens de satélites, dados de radar e produtos experimentais, tais como saídas em alta resolução de modelos e outros dados experimentais preparados e inseridos no Internet Data Distribution (IDD) pelo NOAA Forecast Systems Laboratory (FSL) e pelo NCEP (National Centers for Environmental Prediction)[25].

No Brasil os centros operacionais possuem acesso aos dados através da rede oficial da Organização Meteorológica Mundial, o GTS (Global Telecommunication System), enquanto universidades e empresas normalmente os obtêm através da internet (FTP ou HTTP) a partir dos centros produtores de dados (INMET, INPE, DECEA, etc) ou com que disponibilizam os dados do GTS em endereços de FTP[25].

Para facilitar o acesso aos dados do IDD, Pessoa et al.[26] criou um sistema de aquisição e armazenamento de dados oceânicos, com o objetivo de captar, decodificar, armazenar e extrair os dados provenientes de boias oceânicas, distribuídas pelo IDD via LDM. O acesso aos dados do sistema podem ser acessados via interface web e/ou linha de comando utilizando scripts de extração. Segundo o autor o acesso por script apesar de ser menos intuitivo e amigável é mais eficiente que o método anterior.

Com o objetivo de auxiliar produtores e pesquisadores na análise das variações climáticas e no auxílio da tomada de decisão, Sousa[27] desenvolveu uma metodologia de automatização do processo de coleta e disponibilização de dados meteorológicos, baseada na utilização de scripts. No referido trabalho o autor utilizou o programa software Autolt e a tecnologia de scripts e arquivos em lotes para automatizar a coleta de dados da estação meteorológica, que anteriormente necessitava da intervenção humana para ser realizada[27].

Os scripts são utilizados para realizar o *download* dos dados meteorológicos coletados pela estação meteorológica e posterior para processar os dados. No processamento dos dados os arquivos são lidos e são extraídos somente os dados definidos como interessantes para a disponibilização dos usuários finais. Após o processamento dos dados é executado outro *script* que envia o arquivo com

os dados de interesse do usuário para um diretório na *web*. Então os usuários finais conseguem acessar e consultar os dados da estação por meio de um site web[27].

Silva et al.[28] desenvolveu um sistema web com o objetivo de disponibilizar dados meteorológicos para pesquisadores, agricultores, instituições de fomento e crédito agrícola, veículos de comunicação e o público em geral. Os dados disponibilizados pelo sistema são originados de estações meteorológicas gerenciadas pela Embrapa Meio-Norte, coletados em um intervalo de 30 e 30 minutos por um computador com software de comunicação específicos das estações instaladas. Posteriormente os dados são importados, organizados e armazenados no banco de dados do servidor da instituição[28].

Para acessar os dados de seu interesse os usuários utilizam um sistema web, no qual é possível inserir parâmetros para a pesquisa, tais como período de pesquisa e local, e então obtêm o resultado na tela, tendo a possibilidade de exportar em formato de planilha eletrônica. Conforme Silva et al.[28], "... o sistema dinamizou a forma de manipulação dos dados climáticos e ao mesmo tempo mostrou efetivo na recuperação de informações de variáveis climáticas, mostrando-se uma ferramenta útil o manejo de irrigação das culturas no Estado do Piauí"[28].





### 3. PLATAFORMA DE AQUISIÇÃO E DISPONIBILIZAÇÃO DE DADOS METEOROLÓGICOS

#### 3.1 RESUMO

As variações de tempo e clima tem sido constante alvo de estudos nas mais diversas áreas da economia, tendo um destaque elevado na área agrícola. Cada vez mais as informações meteorológicas tornam-se importantes para a realizam de estudos e pesquisas que visam garantir a sustentabilidade da sociedade em geral. Diversas instituições de estudos de meteorologia disponibilizam dados de tempo e clima de forma aberta e gratuita. Contudo o acesso a estes dados nem sempre é uma tarefa trivial devido a diversos fatores dificultantes, principalmente no que diz respeito a heterogeneidade dos repositórios de dados. Neste contexto, considerando a importância da utilização dos dados meteorológicos, este trabalho apresenta uma plataforma que tem por objetivo principal auxiliar o acesso aos dados meteorológicos. A plataforma foi desenvolvida na plataforma Java, sendo utilizada diversas tecnologias e frameworks, tais como serviços web, JSF e PostgreSQL.

#### 3.2 INTRODUÇÃO

Nas últimas décadas houve um interesse acentuado pela climatologia no mundo, principalmente devido aos fenômenos meteorológicos que vem acontecendo e a preocupação com a geração de alimentos. Isto pode ser comprovado pela criação de diversos órgãos governamentais especializados em registrar, monitorar e analisar as condições de tempo e clima. Estes órgãos prestam diversos serviços relevantes para a comunidade, destacando-se principalmente no meio acadêmico e em centros de pesquisa a disponibilização das observações históricas e de previsões do tempo e clima.

O acesso aos dados meteorológicos têm sido de fundamental importância para a realização de muitos trabalhos que avaliam o impacto das variações climáticas nas mais diversas áreas, como por exemplo os trabalhos desenvolvidos por [7][9][10]. A partir do relacionamento dos elementos meteorológicos com dados específicos da área em estudo é possível gerar informações que facilitam a interpretação de resultados e auxiliam na tomada de decisão.

Atualmente existe uma grande quantidade de repositórios de dados meteorológicos disponíveis na rede mundial de computadores, principalmente disponibilizados por órgãos governamentais de meteorologia. Porém acesso aos mesmos não é uma tarefa trivial, devido a diversos fatores dificultantes, entre eles se destacam a dispersão dos repositórios de dados, a falta de padronização dos repositórios, a falta de padronização no formato dos arquivos disponibilizados.

Tendo em vista as dificuldades expostas e a importância da utilização dos dados meteorológicos, este trabalho objetivou o desenvolvimento de uma plataforma flexível e adaptativa que possibilita o acesso de forma transparente aos repositórios de dados. Além de facilitar o acesso, a

plataforma também possibilita o acoplamento de scripts de download e processamento de dados, de forma genérica, flexível e escalável. Isto permite que o sistema acesse os repositórios, efetue o download e o processamento dos dados de forma automática e disponibilize estes em formatos de fácil manipulação para o usuário final.

Com o objetivo de melhor descrever a solução desenvolvida, o restante deste trabalho está organizado como segue. A seção 3.3 introduz os aspectos técnicos envolvidos na concepção e desenvolvimento. Na seção 3.4 é apresentada a arquitetura da plataforma e os respectivos componentes funcionais. Por fim, as seções 3.5 e 3.6 apresentam as considerações finais e conclusões do trabalho.

### 3.3 DESENVOLVIMENTO

Quando se trata de desenvolvimento de software a utilização de processos, técnicas e boas práticas são fundamentais para alcançar os objetivos esperados e obter um produto final de qualidade. Neste sentido, a plataforma desenvolvida neste trabalho foi projetada com base em boas práticas de engenharia de software, tendo a arquitetura estrutura de componentes funcionais interoperantes.

De acordo com Sommerville[29] "*A arquitetura baseada em componentes é uma das seis melhores práticas de engenharia de software recomendadas pelo RUP (Rational Unified Process), justamente por envolver uma abordagem orientada a reuso*".

Os componentes da plataforma são distribuídos de acordo com o papel que exercem para o funcionamento da mesma, conforme descrito a seguir. O Frontend é o componente responsável pelas interações com os usuários da plataforma. O Backend constitui o acesso aos repositórios de dados, bem como a aquisição e processamento dos dados. O Service Layer (SL) é o componente que realiza as interações entre os componentes da plataforma. O Database Access Library (DAL) é o componente que possibilita o acesso ao banco de dados da plataforma. Nas seções seguintes são descritos detalhes de cada componente da plataforma.

#### 3.3.1 Frontend

As interfaces desenvolvem um papel importante na interação homem-máquina, em especial as interfaces gráficas quando o assunto em questão é software. Neste sentido, o Frontend é o componente responsável por realizar a interface de comunicação dos usuários com a plataforma, abstraindo toda a complexidade e os aspectos técnicos necessários para o acesso aos repositórios de dados.

Para o desenvolvimento do componente foi utilizado o *pattern Model View Controller* – Modelo Visão Controle (MVC), por meio do *framework* Java Server Faces (JSF). Segundo Jacobi e Fallows[30], "*O JSF é um framework de interface de usuário (UI) para aplicações Java, criado para facilitar o desenvolvimento e a manutenção de aplicações que rodam em um Java application Server e apresentam suas UIs de volta para um cliente específico*".

O MVC é um padrão arquitetural de desenvolvimento de aplicações que separa a aplicação em três camadas funcionais. A camada *model* constitui na lógica e domínio relacionados aos objetos que implementam as funcionalidades principais da aplicação. A camada *view* é a camada de apresentação dos dados, na qual encontram-se os objetos que representam a interface gráfica com o usuário. A camada *controller* atua como um intermediário entre as camadas *view* e *model*, além de controlar os fluxos da aplicação[31].

Neste contexto, na camada de visão foi utilizado o *framework* de interface PrimeFaces, pela usabilidade e a simplicidade, sendo um dos *frameworks* de interface rica mais utilizado por desenvolvedores na plataforma Java. O PrimeFaces consiste em uma suíte de componentes JSF de código aberto que possui diversas extensões[32].

Ainda na camada de visão, foi utilizada a API (*Application Programming Interface*) do Google Maps[33] para auxiliar na localização geográfica na busca por repositórios de dados. A API fornece informações das coordenadas geográficas de um local específico entre outras informações de localização, como por exemplo, rua, cidade, estado e país. Na camada de controle encontram-se as classes responsáveis pelos fluxos de requisições das aplicações. Na camada de modelo foi utilizado o componente da própria plataforma DAL para efetuar a conexão e as transações com o banco de dados da plataforma.

### 3.3.2 Backend

Tendo em vista que diversas atividades realizadas pela plataforma não são possíveis de serem gerenciadas e visualizadas pelos usuários por meio das interfaces gráficas, é necessário um componente que execute e gerencie tais atividades e informando aos usuários sobre o andamento das mesmas, como por exemplo, o processamento das requisições de dados dos usuários.

Neste sentido, o Backend constitui um componente projetado para ser executado em segundo plano com o objetivo principal de gerenciar o processamento das requisições de dados dos usuários. Para isto o componente implementa um processo específico, conforme é possível visualizar na Figura 2.

Inicialmente é realizada a etapa de aquisição, onde os repositórios de dados são acessados e extraídos os dados de interesse do usuário para um diretório local da plataforma. Na segunda etapa, o processamento, os dados extraídos são abertos e executadas as rotinas de tratamento sob os mesmos. As rotinas executam diversas funções, tais como a extração de variáveis e períodos de interesse, extração de áreas de interesse (no caso de arquivos de grade) e outras. Após realizar o tratamento dos dados os dados são organizados em um arquivo de saída e disponibilizados para o usuário.

O componente atua como uma camada que encapsula o acesso aos repositórios de dados, bem como a aquisição e o processamento dos dados meteorológicos solicitados pelos usuários. Entende-se como repositório de dados meteorológicos, qualquer local que contenha dados meteorológicos disponíveis na *web*, tais como serviços *web*, *web site* e endereços FTP.

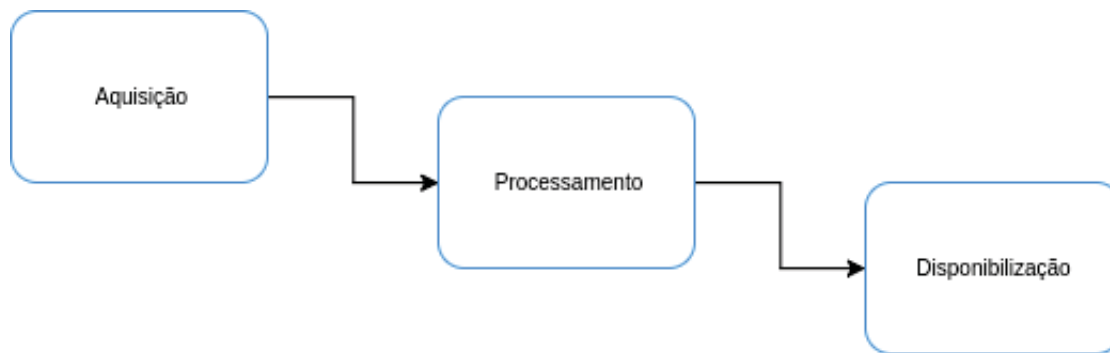


Figura 2. Processo utilizado nas requisições de dados implementado no Backend

Entre as características desejáveis para a plataforma se destacam ser uma ferramenta genérica, flexível e escalável, ou seja, extensiva a vários repositórios de dados e que permita a inclusão de novos repositórios sem a necessidade de alterar da estrutura da mesma. Porém um dos fatores dificultantes para atender estas características é a heterogeneidade dos repositórios de dados quanto a forma de acesso, estruturas de diretórios, formatos de disponibilização dos arquivos e outros.

Para tratar a heterogeneidade, visa-se a utilização de *scripts* para tratamento das características de cada repositório de dados. Os *scripts* são rotinas de códigos desenvolvidas em linguagens de programação e vinculadas ao cadastro do repositório de dados na plataforma, que tratam as especificidades de cada repositório de dados. Desta forma, cada repositório possui um *script* de *download* e um de processamento de dados.

O Backend ao processar uma requisição de dados busca no banco de dados da plataforma a informação de quais são os *scripts* vinculados ao repositório que se deseja obter os dados e realiza a execução dos mesmos, conforme Figura 3. No caso do *script* de *download*, o Backend invoca a execução do *script*, que por sua vez acessa o repositório de dados e efetua o download dos arquivos. Ao finalizar o *script* acessa a Service Layer para informar que a rotina foi finalizada, conforme é possível visualizar no diagrama de seqüências da Figura 4. O *script* de processamento segue o mesmo modelo de funcionamento.

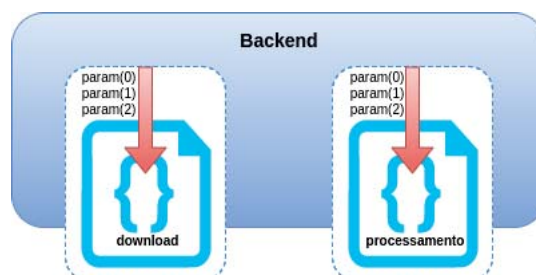


Figura 3. Comunicação do Backend com os scripts

Os *scripts* podem ser desenvolvidos em diversas linguagens de programação, tais como Java, PHP, Python, Shell Script. Nesta versão da plataforma somente *scripts* desenvolvidos na linguagem Java são suportados. As informações necessárias para a execução de cada *script* são

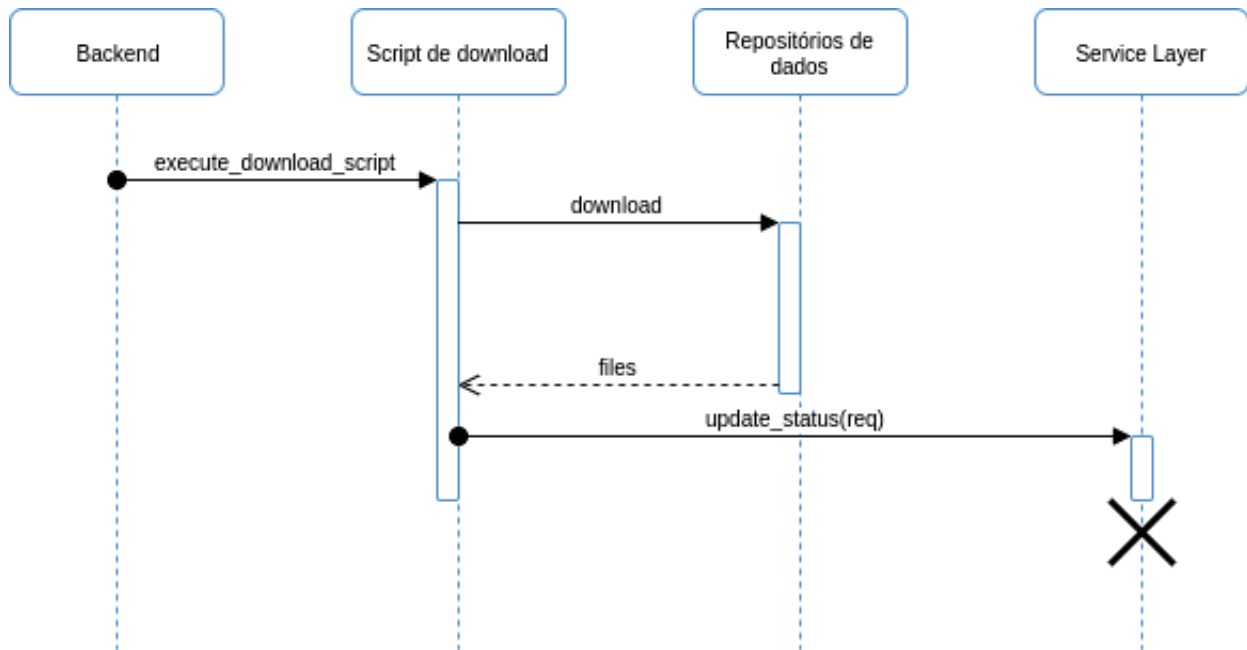


Figura 4. Diagrama de seqüências das interações entre Backend - *script* - repositórios de dados -Service Layer

repassadas pelo Backend por meio de parâmetros, sendo necessário para cada *script* estar apto a receber estes dados. Na Tabela 1, é apresentado o conjunto de parâmetros com os respectivos formatos e descrição.

Tabela 1. Relação de parâmetros informados aos *scripts*

Parâmetro	Formato	Exemplo	Descrição
nrrequisicao	#####	00001	Número da requisição processada
datainicial	DDMMAAAA	12122016	Data inicial que se deseja obter os dados
datafinal	DDMMAAAA	15122016	Data final que se deseja obter os dados
regiao	_PAIS_ UF_ CIDADE ou _X:##.##### _Y:##.#####	_BR_RS_PassoFundo ou _X:-28.232724 _Y:-52.380950	Região para a qual deseja se obter os dados. No caso de arquivos de grade são informados as coordenadas geográficas do local
vars	-var1-var2-var3	tp2m-vento-umrl-prec	Variáveis de interesse
saida	Texto	txt	Formato de Saída

O Backend foi desenvolvido utilizando o modelo de aplicação Java console e configurado para ser executado em segundo plano, conforme tempo pré determinado no agendador de tarefas do sistema operacional. No desenvolvimento do componente foi utilizado o consumo de serviço web padrão REST do componente Service Layer (SL), por meio do qual o Backend estabelece a comunicação com os demais componentes da plataforma. Também foi implementado o conceito de *threads* para gerenciar a execução das atividades de *download* e processamento de dados. Para monitorar a execução das tarefas do Backend, foi utilizada a biblioteca Log4J para gerar *logs* da execução das atividades.

### 3.3.3 Database Access Library

No intuito de maximizar o reuso na plataforma foi constituído o componente Database Access Library (DAL). Além disso o componente também otimiza e padroniza o acesso das aplicações ao banco de dados da plataforma, por meio do encapsulamento das rotinas de acesso ao banco de dados e da implementação do modelo de dados da plataforma.

O sistema gerenciador de banco de dados (SGBD) utilizado para gerenciar os dados da plataforma foi o PostgreSQL. Tendo em vista que o PostgreSQL é SGBD objeto-relacional e as aplicações que acessam o banco de dados utilizam o modelo de programação orientada a objetos, foi utilizado o ORM (*Object Relational Mapping* – Mapeamento Objeto Relacional) Hibernate com a especificação JPA (*Java Persistence API*) para realizar a transição entre objeto relacional e vice-versa.

### 3.3.4 Service Layer

Para prover a comunicação entre os componentes da plataforma foi desenvolvido o componente SL (Service Layer). O componente implementa serviços no padrão Rest e disponibiliza métodos que permitem recuperar e atualizar os dados. Conforme Lecheta[34] "*Representational State Transfer (REST) é um estilo de arquitetura de software cada vez mais utilizado no mundo inteiro, principalmente para criar serviços web e auxiliar na integração de sistemas*".

Entre as principais características da arquitetura destacam se a utilização do protocolo HTTP para a criação de serviços que retornam dados nos formatos XML ou JSON, sendo uma alternativa para o padrão SOAP e o WSDL. Outra característica relevante é a implementação de um conjunto de operações padronizadas para consulta, inserção, alteração e deleção de dados, correspondentes aos métodos GET, POST, PUT e DELETE respectivamente[34].

No desenvolvimento deste componente foram implementados métodos GET e POST, conforme é possível visualizar na tabela 2. Para a troca de mensagens foi adotado o formato JSON. Para acesso ao banco de dados da plataforma foi utilizado o componente DAL, descrito anteriormente.

Tabela 2. Métodos disponibilizados pela SL

Descrição	Tipo	URL
Recupera uma requisição por meio do ID	GET	dominio.com.br/resources/requisiton/{id}
Verifica se existem requisições no estado determinado	GET	dominio.com.br/resources/requisiton/status/{Status}
Recupera a lista de requisições de um status	POST	dominio.com.br/resources/requisiton/status/
Atualiza os dados de um requisição	POST	resources/requisiton

### 3.4 RESULTADOS E DISCUSSÃO

O objetivo geral deste trabalho foi desenvolver uma estrutura genérica, flexível e escalável, onde seja possível registrar os repositórios de dados meteorológicos e por meio desta acessar os dados disponíveis nos repositórios. Para atender esta especificação, a arquitetura da plataforma foi projetada baseada em componentes interoperantes, conforme Figura 5.

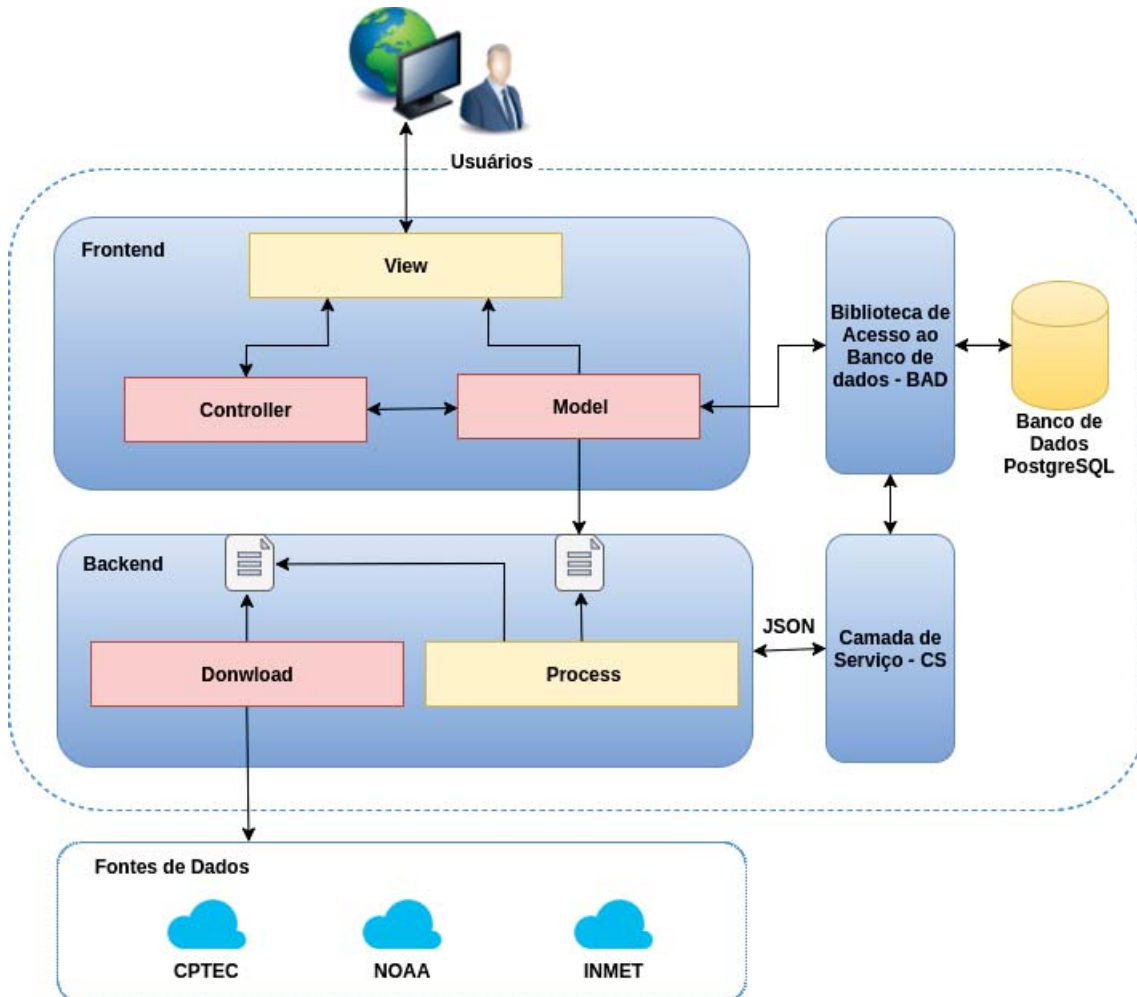


Figura 5. Visão geral da arquitetura da plataforma

Os componentes são sistemas independentes que desempenham funções específicas e interagem entre si para prover as funcionalidades da plataforma. A seguir serão descritos os principais componentes da plataforma, bem como o funcionamento e as interações.

O Frontend foi dividido em duas aplicações *web* que fazem a interface de comunicação da plataforma com os usuários. A aplicação administrativa tem suas funcionalidades voltadas para a manutenção e gerenciamento dos dados plataforma, dentre as quais se destaca o cadastro de repositório de dados (Figura 6).

A aplicação pública é destinada para usuários finais (pesquisadores, estudantes), onde os mesmos podem consultar os repositórios existentes e gerar as requisições de dados para a plataforma, conforme Figura 7.

Edição de Registro		Scripts	Variáveis	Cidades	Formato de saída
ID	<input type="text"/>				
Nome *	<input type="text"/>				
Descrição	<input type="text"/>				
Url *	<input type="text"/>				
Tipo	<input type="button" value="Observado"/> <input type="button" value="Previsão"/>				
Intervalo de dias	<input type="text"/>				
Atualização	Daily <input type="button" value="v"/>				
Fonte	CPTEC <input type="button" value="v"/>				
Resolução Temporal	Hourly <input type="button" value="v"/>				
<input type="button" value="Cancelar"/> <input type="button" value="Gravar"/>					

Figura 6. Tela de cadastro dos repositórios de dados

Os dados utilizados pelas aplicações são armazenados no banco de dados da plataforma e são acessados por meio do componente DAL. Este componente implementa o modelo, Figura 8 e 9 de dados utilizado pela plataforma. As principais entidades do modelo são "DataSource" e "Requisition", as quais representam os repositórios de dados e as requisições de dados dos usuários.

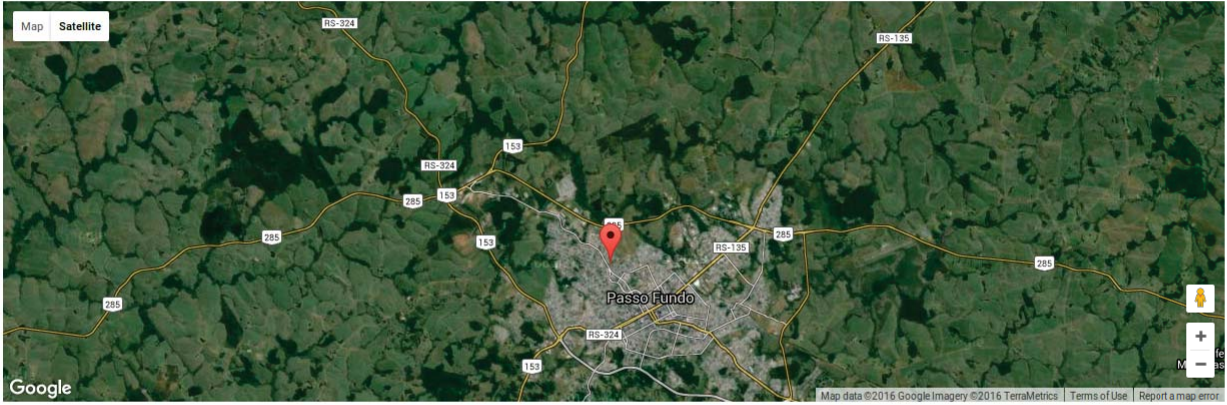
MetData	
Home Login Sign Up About	Language <input type="button" value="v"/>
Address <input type="text"/> <input type="button" value="search"/> <input type="button" value="Clear"/>	
	
Type Of Data	
<input checked="" type="radio"/> Observed	
Period	
Data inicio *	<input type="text" value="1900-01-01"/>
Data Fim *	<input type="text" value="2016-02-29"/>
<input type="button" value="Next"/>	

Figura 7. Tela de pesquisa dos repositórios de dados



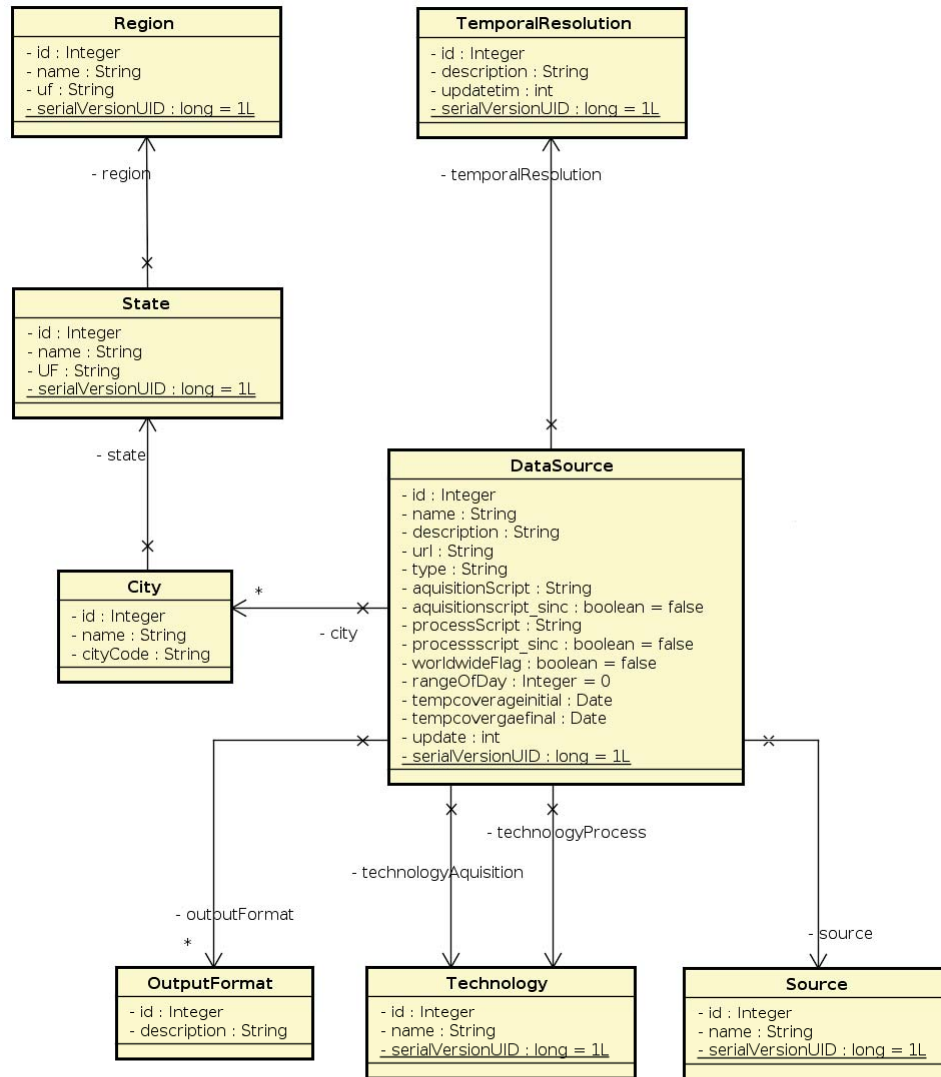


Figura 8. Diagrama de classes do modelo de dados<sup>1</sup> da plataforma - parte 1

O Backend funciona como um serviço no sistema operacional, tendo as rotinas executadas em segundo plano em períodos de tempo pré determinados no agendador de tarefas do sistema operacional. Em cada execução o componente verifica o *status* das requisições e executa o *script* de *download* e/ou processamento das requisições pendentes, levando em consideração o limite de processamento estipulado pela plataforma.

Além de executar os *scripts*, o Backend também é responsável por criar a estrutura de diretórios necessária para o processamento das requisições, conforme Figura 10. Cada requisição necessita de uma pasta raiz, nomeada com o número da requisição e dentro desta outras três pastas denominadas *download*, *unzip* e *output*. As pastas são utilizadas para organizar os arquivos de dados baixados dos repositórios, os dados processados e demais arquivos necessários no processamento das requisições.

O processamento das requisições consiste na aquisição e disponibilização dos dados solicitados pelo usuário. Para isto, a plataforma segue um processo com as seguintes etapas: (i) a

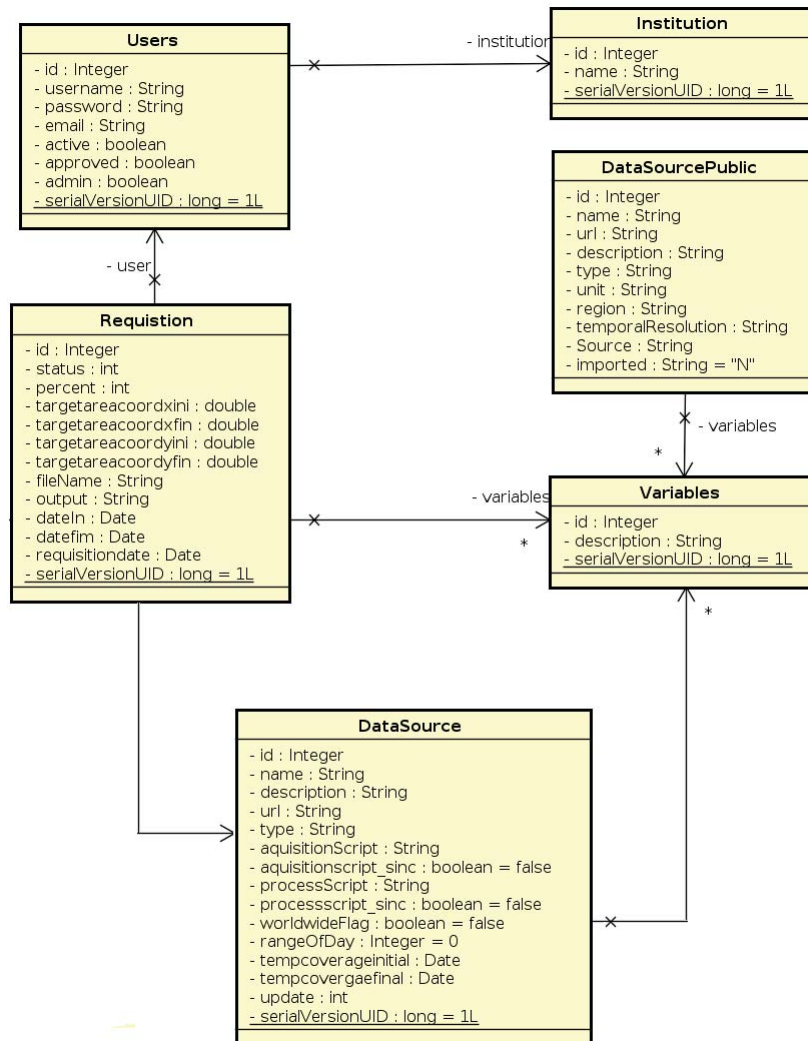


Figura 9. Diagrama de classes do modelo de dados da plataforma - parte 2

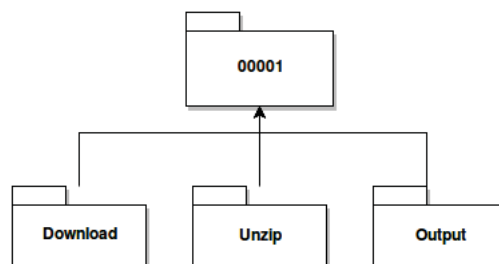


Figura 10. Estrutura de diretório criada pelo Backend no processamento de cada requisição

requisição é gerada pelo usuário por meio da aplicação pública do Frontend, a qual armazena os dados da requisição no banco de dados da aplicação; (ii) o Backend consulta as requisições pendentes para *download*, por meio da camada de serviço, e executa o *script* de *download* do repositório vinculado a requisição; (iii) o Backend informa plataforma que o *download* dos arquivos da requisição iniciou; (iv) o *script* de *download*, acessa o repositório de dados e efetua o *download* do(s) arquivo(s) com os dados de interesse do usuário para o diretório *Output* da requisição; (v) o *script* de *download* informa a plataforma que o *download* dos arquivos foi finalizado. (vi) o Backend

consulta as requisições pendentes de processamento de dados, por meio da camada de serviço, e executa o *script* de *processamento de dados* do repositório vinculado a requisição; (vii) o Backend informa a plataforma que o processamento dos dados da requisição iniciou; (viii) o *script* de processamento realiza o tratamento dos dados conforme os parâmetros definidos pelo usuário; (ix) o *script* de processamento de dados informa plataforma a finalização do processo. Na Figura 11 é possível visualizar o fluxo de processamento das requisições.

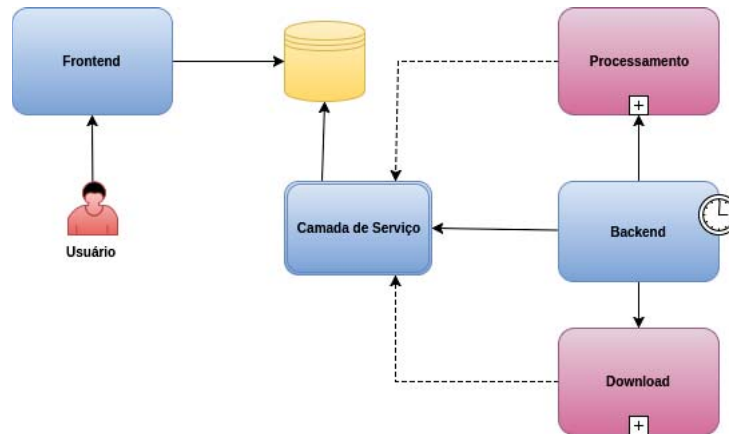


Figura 11. Fluxo de processamento das requisições

Para controlar o andamento da requisição no fluxo de processamento, a requisição assume vários estados, conforme pode se visualizar no diagrama de máquina de estados da Figura 12. O estado é controlado pelo atributo *status* da entidade *Requisition* Figura 9.

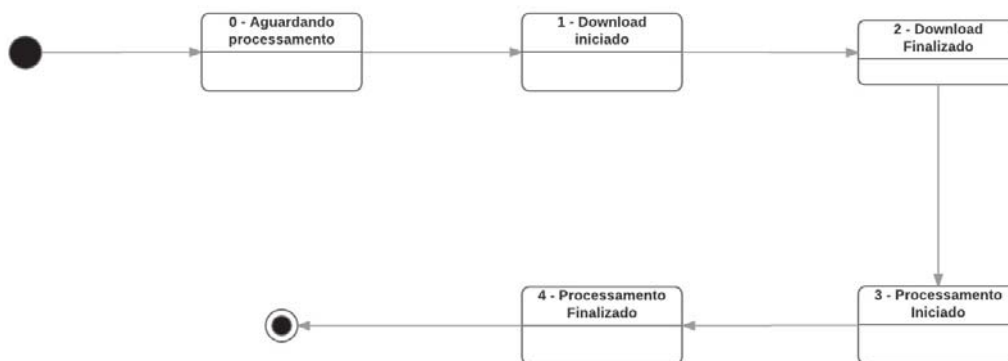


Figura 12. Diagrama de máquina de estados das requisições

Quando a solicitação é gerada o estado inicial é 0 (zero), o que indica para o sistema que a requisição está na fila aguardando para iniciar o download dos dados. Ao iniciar a atividade de download, o *status* desta solicitação é alterado para o valor 1 (um), indicando que a requisição está em atividade de download. Quando a atividade é finalizada, o *status* da requisição é alterado para o valor 2 (dois), para indicar que a tarefa foi finalizada.

A próxima atividade a ser executada é o processamento dos dados, ao iniciar o processamento o estado é alterado para o valor 3 (três), indicando que o processamento dos dados está em andamento. Após finalizada a atividade, o *status* é alterado para o valor 4 (quatro), indicando ao sistema que os dados estão disponíveis para o usuário realizar o download.

A plataforma resultante deste trabalho se apresenta como uma ferramenta interessante para estudantes, pesquisadores e demais interessados em dados meteorológicos, pois além de facilitar a localização dos repositórios de dados permite o acesso aos mesmos de forma transparente.

Diferente das soluções existentes apresentadas neste trabalho, a plataforma foi desenvolvida de forma genérica, flexível e escalável permitindo o acoplamento de novos repositórios de dados, bem como dos mecanismos de *download* e processamento de dados. No trabalho desenvolvido por Silva et al[28], também foi utilizada a tecnologia de aplicações *web* para disponibilizar dados meteorológicos para pesquisadores, agricultores, instituições de fomento e outros, porém a solução se restringe aos dados fornecidos pelas estações meteorológicas da Embrapa Meio-Norte.

Da mesma forma, Sousa et al[27] também utilizou as tecnologias de *script* e aplicações *web* para automatizar e disponibilizar os dados oriundos de estações meteorológicas. O trabalho também se diferencia por ser uma solução específica para uma necessidade, no entanto a tecnologia de *scripts* empregada para automatizar o processo de coleta, apresentam se como uma solução interessante.

Quanto a aquisição e disponibilização de dados a plataforma se mostrou estável nos testes realizados, porém alguns fatores como link de internet e recursos computacionais do servidor onde as aplicações estão instaladas podem interferir diretamente no desempenho das atividades.

### 3.5 CONSIDERAÇÕES FINAIS

A utilização de dados meteorológicos tem sido fundamental para a realização de importantes estudos, principalmente no que diz respeito as variações de tempo e clima nas mais diversas áreas da sociedade.

Neste contexto, este trabalho apresentou uma plataforma para aquisição e disponibilização de dados meteorológicos, que permite a localização de repositórios de dados meteorológicos, bem como a aquisição e disponibilização dos dados. Desta forma, os usuários da ferramenta podem focar os seus esforços na elaboração dos estudos e pesquisas envolvendo dados meteorológicos e não na obtenção destes.

Com o intuito de ser uma ferramenta genérica, flexível e escalável, a estrutura da plataforma foi projetada para permitir o acoplamento de novos repositórios de dados de maneira dinâmica. Isto se deve a utilização da tecnologia de *scripts*, que permitem tratar a heterogeneidade dos repositórios de dados.

Desta forma, entende-se que os resultados demonstrados foram satisfatórios e os objetivos do trabalho foram alcançados, ainda que algumas melhorias e correções sejam necessárias, como por exemplo a compatibilização de *scritps* desenvolvidos em linguagens de programação tais como PHP, Shell Script, Python, C++.

No que diz respeito a estrutura e arquitetura adotada para a solução, a separação em componentes possibilitou uma melhor organização da plataforma, facilitando a expansão e a ma-

nutenção da solução. A escolha da plataforma Java e das tecnologias adotadas apresentou se de forma satisfatório, tendo em vista que supriu todas as necessidades da solução.



#### 4. ESTUDO DE CASO: REPOSITÓRIO DE DADOS DE PREVISÃO MODELO ETA 15KM

Para validação da plataforma foi implementado um estudo de caso utilizando um dos repositórios de dados prognósticos do modelo ETA 15KM[35], disponibilizados pelo CPTEC no endereço FTP "<ftp://ftp1.cptec.inpe.br/modelos/io/tempo/regional/Eta15km/sisalert/>".

O ETA é um modelo de mesoescala, em ponto de grade, de equações primitivas utilizado internacionalmente para previsão meteorológica operacional e em pesquisa. As variáveis prognósticas mais comuns do modelo são: temperatura do ar, componentes zonal e meridional do vento, umidade específica, água líquida da nuvem, pressão à superfície e energia cinética turbulenta [35].

O repositório disponibiliza dados de previsão para várias cidades do Brasil e de outros países da América Latina em dois horários de referência, sendo um a meia-noite (0h) e o outro ao meio dia (12h). Cada arquivo contém dados de projeção horária de 7 dias, totalizando 169 horas de projeção, conforme Figura 13;

hora	tp2m	umr1	lwnv	mdnv	hinv	prec	ocis	vento	dirv	pslc	tgsc
1	26.12	90.22	0.00	0.00	0.02	0.00	0.00	3.90	352.11	994.12	24.26
2	24.47	88.81	0.74	0.74	0.75	2.44	0.00	3.41	353.16	993.86	24.19
3	24.43	87.03	0.71	0.71	0.71	1.87	0.00	3.94	358.31	993.81	24.15
4	24.12	86.15	0.62	0.62	1.00	0.89	0.00	2.97	349.59	993.60	24.06
5	23.76	85.79	0.62	0.62	1.00	0.86	0.00	2.35	348.76	993.36	23.95
6	23.50	86.16	0.58	0.63	1.00	0.84	0.00	2.29	348.29	993.54	23.84
7	23.27	86.26	0.58	0.61	1.00	0.74	0.00	1.93	357.21	993.47	23.72
8	23.10	85.96	0.47	0.77	1.00	0.37	0.00	1.94	2.64	993.73	23.60
9	22.95	85.20	0.48	0.75	1.00	0.76	0.00	2.21	1.26	994.13	23.48
10	22.78	83.96	0.34	0.75	1.00	0.19	0.00	2.23	11.38	994.55	23.36
11	22.59	83.16	0.00	0.88	1.00	0.00	0.00	2.16	13.75	994.99	23.22
12	23.49	83.15	0.00	0.76	1.00	0.00	14.58	1.91	9.86	995.19	23.11
13	24.55	82.93	0.01	0.12	1.00	0.00	84.55	1.39	349.52	995.48	23.32
14	25.84	86.17	0.63	0.63	1.00	0.93	460.42	2.98	13.72	996.13	23.72
15	25.79	87.37	0.71	0.71	1.00	1.81	409.81	4.21	13.82	995.55	24.00
16	25.80	87.31	0.60	0.60	0.69	0.75	399.35	3.94	12.11	994.48	24.26
17	26.28	87.03	0.63	0.63	0.63	0.97	526.59	3.42	14.06	994.18	24.60
18	26.53	86.11	0.60	0.60	0.60	0.71	522.11	3.33	14.24	993.53	24.91
19	26.75	84.97	0.63	0.63	0.63	0.95	526.11	3.03	12.63	993.03	25.22
20	26.76	83.95	0.59	0.59	0.59	0.67	456.90	2.77	5.65	992.30	25.46
21	26.77	83.61	0.56	0.56	0.56	0.54	428.06	2.71	10.92	992.42	25.65
22	26.59	83.15	0.59	0.59	1.00	0.66	385.99	2.77	9.23	992.68	25.76
23	26.14	82.98	0.62	0.62	1.00	0.90	167.86	2.68	7.79	992.55	25.76
24	24.78	82.08	0.58	0.58	1.00	0.63	99.33	2.57	12.65	992.86	25.62
25	24.48	81.82	0.41	0.41	1.00	0.35	66.22	2.16	16.95	993.79	25.39
26	24.21	82.29	0.56	0.56	1.00	0.61	0.00	1.46	21.31	994.72	25.15
27	23.91	83.03	0.51	0.65	1.00	0.73	0.00	1.40	23.11	995.25	24.94
28	23.62	83.75	0.16	0.64	1.00	0.27	0.00	1.38	24.84	995.56	24.73
29	23.26	84.21	0.00	0.80	1.00	0.00	0.00	1.27	28.95	996.21	24.50
30	22.94	84.60	0.00	0.89	1.00	0.00	0.00	1.27	35.00	996.47	24.27
31	22.66	85.12	0.00	0.85	1.00	0.00	0.00	1.08	40.29	996.10	24.06
32	22.40	85.53	0.00	1.00	1.00	0.00	0.00	0.85	54.56	995.87	23.85
33	22.24	85.52	0.00	1.00	1.00	0.00	0.00	0.48	37.51	995.96	23.66
34	22.10	84.98	0.00	1.00	1.00	0.00	0.00	0.29	1.63	996.17	23.48
35	21.97	84.24	0.00	1.00	1.00	0.00	0.00	0.13	318.20	996.27	23.31
36	22.78	83.43	0.00	1.00	1.00	0.00	0.93	0.23	126.50	996.44	23.20
37	23.91	83.09	0.00	1.00	1.00	0.00	52.10	0.51	129.55	997.26	23.34
38	25.33	82.72	0.00	1.00	0.66	0.00	268.00	0.70	117.84	998.24	23.69
39	26.60	82.78	0.00	0.04	0.00	0.00	353.61	0.68	124.27	997.46	24.18

Figura 13. Arquivo disponibilizado em um dos repositórios do modelo ETA 15KM

A cada disponibilização de dados é gerada uma pasta com o padrão de nomenclatura AAAAMMDDHH, conforme pode ser visualizado no Figura 14, onde AAAA corresponde ao ano de disponibilização do arquivo; MM ao mês de disponibilização do arquivo; DD ao dia de disponibilização do arquivo; HH ao horário de disponibilização do arquivo, que pode ser "00" ou "12", tendo em vista que os arquivos são disponibilizados a meia-noite e ao meio dia.

Em cada pasta são disponibilizados em torno de 559 arquivos com dados prognósticos, sendo que cada arquivo contém os dados de previsão de uma cidade. Os arquivos são disponibilizados em formato texto (\*.txt) e possuem um layout de dados fixo, conforme é possível visualizar na Figura 13.

Name	Size	Last Modified
2017021512		15-02-2017 17:07:00
2017021600		16-02-2017 05:19:00
2017021612		16-02-2017 18:02:00
2017021700		17-02-2017 05:20:00
2017021712		17-02-2017 18:01:00
2017021800		18-02-2017 04:39:00
2017021812		18-02-2017 17:27:00
2017021900		19-02-2017 03:42:00
2017021912		19-02-2017 15:58:00
2017022000		20-02-2017 03:50:00
2017022012		20-02-2017 16:24:00
2017022100		21-02-2017 03:40:00
2017022112		21-02-2017 16:10:00

Figura 14. Organização e nomenclatura das pastas do repositório ETA 15KM

Os arquivos seguem o padrão de nomenclatura conforme é possível visualizar na Figura 15, onde "eta\_15km" corresponde ao prefixo utilizado; AAAA, MM, DD e HH seguem o mesmo padrão utilizado nas pastas; SIGLAPAIS corresponde a sigla do país da previsão, por exemplo, BR, PA, PY; UF corresponde a sigla da unidade federativa do estado da previsão, por exemplo, RS, SC, PR; CIDADE corresponde ao nome da cidade da previsão, sem espaços, acentos e caracteres especiais, como por exemplo "PassoFundo", "PortoAlegre". Na Figura 15 é possível visualizar dois exemplos de arquivos de dados prognósticos para as cidades de Passo Fundo e Porto Alegre.

**Nomenclatura:**

**eta\_15km\_AAAAMMDDHH\_SIGLAPAIS\_UF\_CIDADE.txt**

**Exemplo:**

 eta\_15km\_2017022112\_BR\_RS\_PassoFundo.txt

 eta\_15km\_2017022112\_BR\_RS\_PortoAlegre.txt

Figura 15. Nomenclatura e exemplo de arquivos do repositório ETA 15KM

Uma vez analisadas as características e o funcionamento do repositório de dados, foram desenvolvidos os *scripts* de *download* e processamento de dados para o repositório, conforme códigos da Tabela 7 e 8. Os *scripts* foram desenvolvidos utilizando a linguagem de programação Java e as bibliotecas nativas da linguagem.

No *script* de *download*, inicialmente foram setados os valores dos parâmetros de entrada para variáveis locais e realizada a adequação dos valores da variáveis de acordo com as necessidades do *script*, conforme é possível visualizar no código da Tabela 7. Também foram configurados os diretórios utilizados pela requisição e alterado o *status* da requisição para indicar que a atividade de *download* foi iniciada.

Posteriormente foi realizada a conexão com o endereço de FTP do repositório de dados e efetuado o *download* do arquivo com os dados de interesse do usuário para o servidor da aplicação.



Ao finalizar o *download* dos arquivos foi alterado novamente o *status* da requisição para informar a plataforma que a atividade de *download* foi finalizada.

Para o *script* de processamento, inicialmente foi seguida a mesma rotina de leitura dos parâmetros de entrada, configuração do diretório da requisição e alteração do *status* da requisição utilizada no *script* de *download*. Em seguida, o arquivo baixado foi carregado e realizado o tratamento dos dados. Neste caso, os dados foram organizados conforme as variáveis e período selecionados pelo usuário na requisição. Após realizar o tratamento, um novo arquivo com os dados solicitados pelo usuário foi gerado e gravado na pasta *Output* do diretório da requisição. Por fim, foi alterado o status da requisição, para informar a plataforma a finalização da atividade de processamento dos dados e também do processamento da requisição.

Uma vez analisadas as características do repositório e desenvolvidos os *scripts* de *download* e processamento de dados para o repositório, foi efetuado o cadastramento do repositório na plataforma, por meio da aplicação administrativa do Frontend.

Edição de Registro		Scripts	Variáveis	Cidades	Formato de saída
ID	<input type="text" value="7"/>				
Nome *	<input type="text" value="Modelo ETA 15KM"/>				
Descrição	<input type="text" value="Modelo de Previsão Numérica ETA 15KM"/>				
Url *	<input type="text" value="ftp://ftp1.cptec.inpe.br/modelos/io/tempo/regional/Eta15km/sisalert/"/>				
Tipo	<input type="radio"/> Observado <input checked="" type="radio"/> Previsão				
Intervalo de dias	<input type="text" value="7"/>				
Atualização	Daily ▾				
Fonte	CPTEC ▾				
Resolução Temporal	Hourly ▾				
		<input type="button" value="Cancelar"/> <input type="button" value="Gravar"/>			

Figura 16. Tela de cadastro dos dados básicos do repositório Modelo ETA 15KM

Inicialmente foram informados os dados básicos de caracterização do repositório, conforme se pode observar na Figura 16. Em seguida foram vinculados os arquivos dos *scripts* de *download* e processamento, conforme Figura 17.

Na tela da Figura 18 foram selecionadas as variáveis meteorológicas que serão disponibilizadas pelo repositório. Na sequência foram selecionadas as cidades para as quais o repositório está disponível, conforme Figura 19. Por fim, foi selecionado o formato de saída dos dados, conforme Figura 20.

Edição de Registro | Scripts | Variáveis | Cidades | Formato de saída

Tecnologia Aquisition	JAVA
Aquisition Script	<p>+ Aquisition Script Upload Cancel</p> <p>Download_ETA.jar 48.2 KB</p>
Tecnologia Process	JAVA
Process Script	<p>+ Process Script Upload Cancel</p> <p>Process_ETA.jar 39.7 KB</p>

Figura 17. Tela de cadastro dos *scripts* de *download* e processamento do repositório Modelo ETA 15KM

Edição de Registro | Scripts | Variáveis | Cidades | Formato de saída

Selecione um registro		
Novo registro		+ Adicionar
Variáveis	Opções	
Pressão Atmosférica	🗑	
Radiação	🗑	
Temperatura Máxima 2M Altura	🗑	
Umidade Relativa	🗑	
Precipitação	🗑	
Velocidade do Vento	🗑	

Figura 18. Tela de cadastro das variáveis disponibilizadas pelo repositório Modelo ETA 15KM

Edição de Registro | Scripts | Variáveis | Cidades | Formato de saída

Global \*  
 Sim  Não

Selecione um registro		Passo Fundo RS
Novo registro		+ Adicionar
Cidades	Opções	
Porto Alegre	🗑	
Passo Fundo	🗑	

Figura 19. Tela de cadastro das cidades de cobertura do repositório Modelo ETA 15KM

Após concluir o cadastro do repositório foi gerada uma requisição de dados para o mesmo, por meio da aplicação pública do Frontend. Inicialmente foi realizada uma pesquisa dos repositórios de dados disponíveis para a cidade de Passo Fundo. Como resultado foi apresentado o repositório

Edição de Registro   Scripts   Variáveis   Cidades   Formato de saída

Texto (\*.txt)   + Adicionar

Formato de saída	Opções
Texto (*.txt)	

Figura 20. Tela de cadastro dos formatos de saída do repositório Modelo ETA 15KM

“Modelo de Previsão Numérica ETA 15KM”, conforme Figura 21. Em seguida foram selecionadas as seguintes variáveis meteorológicas: temperatura máxima a 2 metros, umidade relativa, precipitação, velocidade do vento; também foi informado o período de interesse dos dados e o formato de saída dos dados, conforme é possível visualizar na Figura 22. Por fim a requisição de dados foi gerada e o seu número de identificação informado na tela.

Mapa

Listagem de Fonte de Dados				
ID	Tipo	Nome	Url	Opções
11	P	Modelo ETA 15KM	ftp://ftp1.cptec.inpe.br/modelos/10/tempo/reg	Selecionar

Figura 21. Tela de pesquisa de repositório de dados disponíveis para a cidade de Passo Fundo

Para acompanhar o andamento da requisição foi acessada funcionalidade “Minhas Requisições”. Ao concluir o processamento da requisição, o *status* foi alterado para concluído e habilitado o *download* do arquivo de saída, conforme se pode visualizar na Figura 23. Na Figura 24 é possível visualizar o conteúdo do arquivo gerado com os dados solicitados.

Listagem de Fonte de Dados				
ID	Tipo	Nome	Uri	Opções
11	P	Modelo ETA 15KM	ftp://ftp1.cptec.inpe.br/modelos/io/tempo/reg	^ Selecionar

Descrição	
Descrição	
<input type="checkbox"/>	Pressão Atmosférica
<input type="checkbox"/>	Radiação
<input checked="" type="checkbox"/>	Temperatura Máxima 2M Altura
<input checked="" type="checkbox"/>	Umidade Relativa
<input checked="" type="checkbox"/>	Precipitação
<input checked="" type="checkbox"/>	Velocidade do Vento

Data inicial da requisição:

Data final da requisição:

Formato de saída:

Cancelar Gravar

Figura 22. Tela de informação dos parâmetros da requisição para o repositório Modelo ETA 15KM

Filtro Ordenação e Paginação		
(1 of 1) <b>1</b> 10 ▾		
ID	Status	Opções
2	Concluída	^ Detalhes ^ Download
contains contains		
(1 of 1) <b>1</b> 10 ▾		

Figura 23. Tela de acompanhamento das requisições

```

REQ_00001_OUT.csv (-/DAP/Requisitions/00001/Proc
Open Save Undo
REQ_00001_OUT.csv x
hora;tp2m;umrl;prec;vento;
1;22.99;85.70;0.0;7.89;
2;20.75;86.21;0.0;7.48;
3;20.03;85.80;0.0;7.15;
4;19.54;85.34;0.0;7.13;
5;19.36;85.50;0.0;7.38;
6;19.16;85.04;0.0;7.41;
7;18.86;83.07;0.0;7.57;
8;18.25;80.49;0.0;7.26;
9;17.62;78.60;0.0;6.79;
10;17.16;77.21;0.0;6.80;
11;18.86;75.97;0.0;6.03;
12;20.37;75.99;0.0;5.31;
13;22.10;76.44;0.0;5.40;
14;23.77;76.80;0.0;5.30;
15;25.41;77.17;0.0;5.31;
16;26.58;76.80;0.0;4.83;
17;27.67;74.49;0.0;4.67;
18;28.46;71.13;0.0;4.57;
19;28.88;69.07;0.0;4.24;
20;28.94;68.91;0.0;4.35;
21;28.41;70.90;0.0;4.58;
22;27.17;76.58;0.0;5.07;
23;24.57;85.15;0.9;8.52;
24;23.41;80.59;0.8;8.10;
Plain Text Tab Width: 8 Ln 1, Col 1 INS

```

Figura 24. Arquivo de saída da requisição

Tabela 3. *Script de download* do repositório ETA 15KM

```

1 public static void main(String[] args) throws IOException {
2     String nrrequisicao = args[0];
3     String periodoinicial = args[1];
4     String periodofinal = args[2];
5     String cidade = args[3];
6     String dir = "//home//ronaldo//DAP//Requisitions//" +
7     nrrequisicao + "//";
8     String dir_baixados = "Baixados//";
9     try {
10    updateStatusRequisitions(Integer.parseInt(nrrequisicao), 1);
11    logger.info("Iniciando conexao FTP");
12    FTPClient ftp = new FTPClient();
13    ftp.connect("ftp1.cptec.inpe.br", 21);
14    ftp.enterLocalPassiveMode();
15    ftp.login("anonymous", "");
16    checkReply(ftp, "FTP server refused connection.");
17    checkReply(ftp, "FTP server refused login.");
18    logger.info("FTP Conectado");
19    logger.info("Acessando diretorios");
20    ftp.changeWorkingDirectory("modelos");
21    ftp.changeWorkingDirectory("io");
22    ftp.changeWorkingDirectory("tempo");
23    ftp.changeWorkingDirectory("regional");
24    ftp.changeWorkingDirectory("Eta15km");
25    ftp.changeWorkingDirectory("sisalert");
26    ftp.changeWorkingDirectory(periodoinicial + "00");
27    Date data;
28    String d, arquivo = "eta_15km_"+periodoinicial+
29    "00" + cidade + ".txt";
30    ftp.setFileTransferMode(FTPClient.BLOCK_TRANSFER_MODE);
31    ftp.setType(FTPClient.BINARY_FILE_TYPE);
32    FileOutputStream fos;
33    logger.info("Diretorio de trabalho:" + dir + dir_baixados);
34    fos = new FileOutputStream(dir + dir_baixados+arquivo);
35    try {
36    logger.info("Iniciando a transferencia do(s) arquivo(s): " + arquivo);
37    if (ftp.retrieveFile(arquivo, fos)) {
38    logger.info("Arquivo baixado com sucesso!");
39    } else {
40    logger.info("Erro ao efetuar download do arquivo " + arquivo);
41    }
42    } catch (Exception ex) {
43    logger.info("Ocorreu a seguinte execucao ao efetuar download do arquivo:
44    " + ex.getMessage());
45    }
46    logger.info("Atualizando o status da requisicao");
47    updateStatusRequisitions(Integer.parseInt(nrrequisicao), 2);
48    } catch (Exception ex) {
49    System.out.println("Erro: " + ex.getMessage());
50    updateStatusRequisitions(1, -2000);
51    }

```

Tabela 4. *Script* de processamento de dados do repositório ETA 15KM

```

1 public static void main(String[] args) {
2   String nrrequisicao = args[0];
3   String periodoinicial = args[1];
4   String periodofinal = args[2];
5   String cidade = args[3];
6   String variaveis = args[4];
7   String arquivo = "eta_15km_" + periodoinicial + "00" + cidade + ".txt";
8   logger.info("Iniciando o processamento da requisicao: " + nrrequisicao);
9   logger.info("Setando os diretorios de trabalho");
10  String dir = "//home//ronaldo//DAP//Requisitions//" + nrrequisicao + "//";
11  String dir_baixados = "Baixados//";
12  String dir_descopactados = "Descompactados//";
13  String dir_processados = "Processados//";
14  boolean tp2m=false,umrl=false,lwnv=false,mdnv=false,hinv=false,prec=
      false,ocis=false,vento=false,dirv=false,pslc=false,tgsc=false;
15  String[] vars = variaveis.split("-");
16  logger.info("Verificando as variaveis selecionadas");
17  verificaVariaves(vars);
18  String arquivo_novo = "hora;";
19  logger.info("Montando Cabecalho do arquivo de saida");
20  montaCabecalho();
21  arquivo_novo += "\n";
22  logger.info("Iniciando a geracao do arquivo de saida");
23  try {
24    FileReader arq = new FileReader(dir + dir_baixados + arquivo);
25    geraArquivo(arq);
26    logger.info("Arquivo gerado com sucesso!");
27    logger.info("Atualizando o status da requisicao");
28    updateStatusRequisitions(Integer.parseInt(nrrequisicao), 4);
  }

```

## 5. ESTUDO DE CASO: REPOSITÓRIO DE DADOS OBSERVADOS REDEMET

O segundo estudo de caso foi implementado utilizando o repositório de dados observados da Redemet. A Redemet é uma rede de meteorologia do comando da Aeronáutica que tem por objetivo integrar produtos meteorológicos voltados à aviação civil e militar[36]. O repositório permite acesso aos dados observados de diversos aeródromos do país, por meio de uma API disponível no endereço eletrônico "[http://www.redemet.aer.mil.br/api/consulta\\_automatica/index.php](http://www.redemet.aer.mil.br/api/consulta_automatica/index.php)". A consulta aos dados é realizada de acordo com os valores informados aos parâmetros da API. Na tabela 5 é possível visualizar os principais parâmetros, bem como a sua descrição.

Tabela 5. Parâmetros da API da Redemet - Fonte: Redemet[37]

Argumento	Obrigatório	Default	Descrição
local	SIM	vazio	Array de Localidades, FIR ou números sinóticos separados por vírgula (São aceitas no máximo 100 valores por solicitação).
msg	SIM	vazio	Array de Tipo Mensagem que estão disponíveis no banco de dados da REDEMET.
data_ini	NÃO	vazio	Intervalo Inicial no formato YYYYMMDDHH. Caso não seja informado a data inicial será considerado data e hora atual.
data_fim	NÃO	vazio	Intervalo Final no formato YYYYMMDDHH (O intervalo entre a data inicial e final não pode ser superior a 24 horas). Caso não seja informado a data final será considerado data e hora atual.
data_hora	NÃO	sim	Caso NÃO deseje que a referência de data e hora do dia, venha anteriormente a mensagem, informe seu valor como 'nao' (sem acento)
cabecalho	NÃO	não	Caso deseje que o cabeçalho da mensagem venha acima da mensagem informe seu valor como 'sim'
saida_html	NÃO	não	Caso deseje que o resultado possa ser visto diretamente no browser, como uma mensagem em cada linha, informe seu valor como 'sim'
quebra_synop	NÃO	não	Caso deseje que o início da mensagem SYNOP tenha uma quebra de linha após o campo datahora na mensagem, informe seu valor como 'sim'. Exemplo: Mensagem armazenada na base de dados da REDEMET: AAXX 16064 83378 32970 30000 10189 20125 38965 48540 83030 333 59007=Mensagem após a quebra de linha: AAXX 1606483378 32970 30000 10189 20125 38965 48540 83030 333 59007=

Para cada requisição a API pode retorna diversos tipos de mensagens meteorológicas utilizadas para navegação aérea, tais como METAR, TAF, SPECI, SIGMET. Neste estudo de caso foram utilizadas as mensagens do tipo METAR, as quais fornecem informações sobre as condições

do tempo de um aeródromo específico. Um exemplo de mensagem METAR pode ser visualizado na Figura 25.

**METAR SBRJ 231200Z 31015G27KT 280V350 4000 -RA SCT020 BKN120 25/20 Q1012 =**

Figura 25. Exemplo de mensagem METAR

As mensagens do tipo METAR são decodificadas, conforme a tabela 6.

Tabela 6. Decodificação de uma mensagem METAR - Fonte: Redemet[37]

CAMPO	DESCRIÇÃO
METAR	Designador do código para informação meteorológica aeronáutica regular
SBRJ	Indicador de localidade. Exemplo Passo Fundo (SBPF), Porto Alegre (SBPA)
231200Z	Informação do dia e o horário da observação, em horas e minutos UTC (Universal Time Coordinated), seguido da letra indicadora Z
31015G27KT 280V350	Informações da direção e a velocidade do vento à superfície em relação ao norte verdadeiro
4000	Informações referente à visibilidade horizontal predominante, expressa em metros
-RA	Informações de fenômenos de tempo. Exemplo: chuva, chuvisco, neve e etc
SCT020 BKN120	Informações sobre a quantidade e altura da base das nuvens
25/20	Informações sobre a temperatura do ar e do ponto de orvalho
Q1012	Informações sobre a pressão QNH
=	Designador de término da mensagem

Tendo em vista as características e o funcionamento do repositório de dados, foram desenvolvidos os *scripts* de *download* e processamento de dados, conforme códigos da Tabela 7 e 8. Os *scripts* foram desenvolvidos utilizando a linguagem de programação Java e as bibliotecas nativas da linguagem.

No *script* de *download*, inicialmente foram atribuídos os valores dos parâmetros de entrada para variáveis locais e realizado o tratamento dos valores de acordo com o *script*, conforme é possível visualizar no código da Tabela 7. Devido a codificação utilizada pela API para identificação das cidades, foi criado um *array* para fazer a transformação da cidade no código necessário. Também foram configurados os diretórios utilizados pela requisição e alterado o *status* da requisição para indicar que a atividade de *download* foi iniciada.

Posteriormente foi realizada a conexão HTTP com o endereço da API do repositório de dados e realizada a geração de um arquivo de texto com as mensagens METAR contendo os dados de interesse do usuário, conforme Figura . Ao finalizar a tarefa novamente foi alterado o *status* da requisição para informar a plataforma que a atividade de *download* foi finalizada.

Para o *script* de processamento, inicialmente adotou-se a mesma rotina de leitura dos parâmetros de entrada, configuração do diretório da requisição e alteração do *status* da requisição utilizada no *script* de *download*. Em seguida, o arquivo gerado na etapa de *download* foi carregado e então realizado o tratamento dos dados.



```

redemet_00008.txt.txt x
2017032300 - METAR SBPA 230000Z 10009KT CAVOK 23/19 Q1018=
2017032301 - METAR SBPA 230100Z 09011KT CAVOK 22/19 Q1018=
2017032302 - METAR SBPA 230200Z 11007KT CAVOK 22/19 Q1018=
2017032303 - METAR SBPA 230300Z 12008KT CAVOK 22/19 Q1018=
2017032304 - METAR SBPA 230400Z 10005KT CAVOK 22/19 Q1017=
2017032305 - METAR SBPA 230500Z 09004KT CAVOK 21/19 Q1017=
2017032306 - METAR SBPA 230600Z 10005KT CAVOK 21/19 Q1016=
2017032307 - METAR SBPA 230700Z 12003KT CAVOK 21/20 Q1016=
2017032308 - METAR SBPA 230800Z 07002KT CAVOK 21/19 Q1016=
2017032309 - METAR SBPA 230900Z 11002KT CAVOK 21/19 Q1016=
2017032310 - METAR SBPA 231000Z VRB02KT CAVOK 21/20 Q1017=
2017032311 - METAR SBPA 231100Z 08002KT CAVOK 22/21 Q1017=
2017032312 - METAR SBPA 231200Z 07004KT 030V110 CAVOK 24/20 Q1017=
2017032313 - METAR SBPA 231300Z 07003KT 040V110 CAVOK 25/20 Q1017=
2017032314 - METAR SBPA 231400Z 34004KT 280V070 9999 FEW040 26/18 Q1017=
2017032315 - METAR SBPA 231500Z 34005KT 310V040 9999 FEW040 26/17 Q1017=
2017032316 - METAR SBPA 231600Z 33005KT 280V020 9999 FEW040 27/19 Q1016=
2017032317 - METAR SBPA 231700Z 31004KT 240V040 9999 SCT045 BKN070 27/18 Q1015=
2017032318 - METAR SBPA 231800Z 32005KT 220V360 9999 FEW030 BKN100 27/18 Q1015=
2017032319 - METAR SBPA 231900Z VRB04KT 9999 SCT035 FEW040TCU BKN100 28/18 Q1014=
2017032320 - METAR SBPA 232000Z 09004KT 050V130 9999 FEW035 FEW040TCU BKN100 27/18 Q1014=
2017032321 - METAR SBPA 232100Z 13017KT 100V160 9999 BKN040 FEW045TCU 25/19 Q1014=
2017032322 - METAR SBPA 232200Z 10010KT 9999 FEW040 FEW045TCU BKN100 23/19 Q1014=
2017032323 - METAR SBPA 232300Z 11017KT 9999 BKN011 BKN020 FEW030TCU BKN100 22/19 Q1015=
Plain Text Tab Width: 8 Ln 5, Col 59 INS

```

Figura 26. Arquivo contendo as mensagens METAR da requisição do usuário

Para este caso, o tratamento de dados constituiu na interpretação das mensagens METAR contidas no arquivo e extraído os dados conforme as variáveis e período selecionados pelo usuário na requisição. Após o tratamento, um novo arquivo com os dados solicitados pelo usuário foi gerado e gravado na pasta *Output* do diretório da requisição. Por fim, foi alterado o status da requisição, para informar a plataforma a finalização da atividade de processamento dos dados e também do processamento da requisição.

Após analisar as características do repositório e desenvolver os *scripts* de *download* e processamento de dados para o repositório, foi efetuado o cadastramento do repositório na plataforma, por meio da aplicação administrativa do Frontend.

Inicialmente foram informados os dados básicos de caracterização do repositório, conforme se pode observar na Figura 27. Em seguida foram vinculados os arquivos dos *scripts* de *download* e processamento desenvolvidos para o repositório, conforme Figura 28.

Na tela da Figura 29 foram selecionadas as variáveis meteorológicas que serão disponibilizadas pelo repositório. Por tratar-se de dados específicos de algumas cidades, na sequência foram selecionadas as cidades para as quais o repositório está disponível, conforme Figura 30. Por fim, foi selecionado o formato de saída dos dados, conforme Figura 31.

Após concluir o cadastro do repositório foi gerada uma requisição de dados para o mesmo, por meio da aplicação pública do Frontend. Inicialmente foi realizada uma pesquisa dos repositórios de dados disponíveis para a cidade de Passo Fundo. Como resultado foi apresentado o repositório “Redemet”, conforme Figura 32. Em seguida foram selecionadas as seguintes variáveis meteoroló-

Edição de Registro		Scripts	Variáveis	Cidades	Formato de saída
ID	<input type="text" value="8"/>				
Nome *	<input type="text" value="Redemet"/>				
Descrição	<input type="text" value="Rede de Meteorologia do Comando da Aeronáutica"/>				
Url *	<input type="text" value="http://www.redemet.aer.mil.br/?i=facilidades&amp;p=api-redemet"/>				
Tipo	<input type="button" value="Observado"/> <input type="button" value="Previsão"/>				
Tempcoverageinitial	<input type="text" value="01/01/2012"/>	<input type="button" value="🗑"/>			
Tempcoveragefinal	<input type="text"/>	<input type="button" value="🗑"/>			
Atualização	<input type="text" value="Hourly"/> ▾				
Fonte	<input type="text" value="Redemet"/> ▾				
Resolução Temporal	<input type="text" value="Hourly"/> ▾				
		<input type="button" value="Cancelar"/> <input type="button" value="Gravar"/>			

Figura 27. Tela de cadastro dos dados básicos do repositório Redemet

Edição de Registro		Scripts	Variáveis	Cidades	Formato de saída
Tecnologia Aquisition	<input type="text" value="Java"/> ▾				
Aquisition Script	<input type="button" value="➕ Aquisition Script"/> <input type="button" value="↗ Upload"/> <input type="button" value="🗑 Cancel"/>				
	<input type="text" value="Download_Redemet.jar"/> 1.4 MB <input type="button" value="✕"/>				
Tecnologia Process	<input type="text" value="Java"/> ▾				
Process Script	<input type="button" value="➕ Process Script"/> <input type="button" value="↗ Upload"/> <input type="button" value="🗑 Cancel"/>				
	<input type="text" value="Process_Redemet.jar"/> 1.4 MB <input type="button" value="✕"/>				

Figura 28. Tela de cadastro dos *scripts* de *download* e processamento

Edição de Registro		Scripts	Variáveis	Cidades	Formato de saída
<input type="button" value="Selecionar registro"/>		<input type="text" value="Ponto de Orvalho"/> ▾			
<input type="button" value="Novo registro"/>		<input type="button" value="+ Adicionar"/>			
Variáveis		Opções			
Temperatura		<input type="button" value="🗑"/>			
Ponto de Orvalho		<input type="button" value="🗑"/>			

Figura 29. Tela de cadastro das variáveis disponibilizadas pelo repositório

gicas: temperatura e ponto de orvalho; também foi informado o período de interesse dos dados e o formato de saída dos dados, conforme é possível visualizar na Figura 33. Por fim a requisição de dados foi gerada e o seu número de identificação informado na tela.

Edição de Registro   Scripts   Variáveis   Cidades   Formato de saída

Global \*  
 Sim  Não

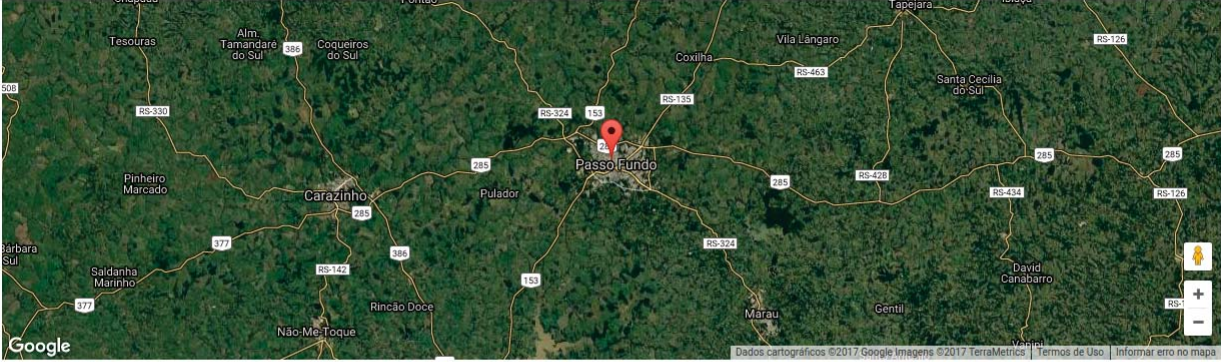
Selecione um registro	Porto Alegre RS
<input type="button" value="Novo registro"/>	<input type="button" value="+ Adicionar"/>
<b>Cidades</b>	<b>Opções</b>
Passo Fundo	<input type="button" value="🗑"/>
Porto Alegre	<input type="button" value="🗑"/>

Figura 30. Tela de cadastro das cidades de disponibilização do repositório

Edição de Registro   Scripts   Variáveis   Cidades   Formato de saída

txt	<input type="button" value="+ Adicionar"/>
<b>Formato de saída</b>	<b>Opções</b>
txt	<input type="button" value="🗑"/>

Figura 31. Tela de cadastro dos formatos de saída



Dados cartográficos ©2017 Google, Imagens ©2017 TerraMetrics, Termos de Uso, Informar erro no mapa

- Mapa

Listagem de Fonte de Dados				
ID	Tipo	Nome	Uri	Opções
7	Previsão	Modelo de Previsão Numérica ETA 15Km	ftp://.....	<input type="button" value="Selecionar"/> <input type="button" value="🗑"/>
8	Observado	Redemet	http://www.redemet.aer.mil.br/?i=facilidades&p=api-redemet	<input type="button" value="Selecionar"/> <input type="button" value="🗑"/>

Figura 32. Tela de pesquisa de repositório de dados disponíveis para a cidade de Passo Fundo

Para acompanhar o andamento da requisição foi acessada funcionalidade “Minhas Requisições”. Ao concluir o processamento da requisição, o *status* foi alterado para concluído e habilitada a opção de *download* do arquivo de saída, conforme se pode visualizar na Figura 34. Na Figura 35 é possível visualizar o conteúdo do arquivo gerado com os dados solicitados.

Listagem de Fonte de Dados				
ID	Tipo	Nome	Url	Opções
7	Previsão	Modelo de Previsão Numérica ETA 15Km	ftp://.....	^ Selecionar ↻
8	Observado	Redemet	http://www.redemet.aer.mil.br/?i=facilidades&p=api-redemet	^ Selecionar ↻

Descrição	
<input checked="" type="checkbox"/>	Descrição
<input checked="" type="checkbox"/>	Temperatura
<input checked="" type="checkbox"/>	Ponto de Orvalho
Data inicial da requisição	2017-04-20 <input type="text"/>
Data final da requisição	2017-04-21 <input type="text"/>
Formato de saída	txt <input type="text"/>

Figura 33. Tela de informação dos parâmetros da requisição

**Filtro Ordenação e Paginação**

(1 of 1)      10 ▾

ID	Status	Opções
<input type="text"/>	<input type="text"/>	
2	Concluída	^ Detalhes ^ Download
contains	contains	

(1 of 1)      10 ▾

Figura 34. Tela de acompanhamento das requisições

Tabela 7. *Script de download do repositório Redemet*

```

1 public static void main(String[] args) throws IOException {
2     String codcidades [][] = {
3         {"PassoFundo", "sbpf"}, {"PortoAlegre", "sbpa"}, {"Pelotas", "sbpk"}, {"
4             SantaMaria", "sbsm"}, {"CaxiasdoSul", "sbcx"}, {"Canoas", "sbco"}, {"
5                 Uruguaina", "sbug"}, {"Bage", "sbbg"};
6     try {
7         logger.debug("*** Recebendo parametros processamento da requisicao *");
8         String nrrequisicao = args[0];
9         String periodoinicial = args[1];
10        String periodofinal = args[2];
11        String cidade = args[3];
12        String endereco [];
13        String codcidade = "";
14        endereco = cidade.split("_");
15        cidade = endereco[2].replace("+", "");
16        for (int i = 0; i < codcidades.length; i++) {
17            if (codcidades[i][0].contentEquals(cidade)) {
18                codcidade = codcidades[i][1];
19                break;}}
20        logger.debug("Setando Diretorios de trabalho");
21        String dir = "//home//ronaldo//DAP//Requisitions//" + nrrequisicao + "
22            //";
23        String dir_baixados = "Baixados//";
24        String dir_descopactados = "Descompactados//";
25        String dir_processados = "Processados//";
26        String arquivo = "redemet_" + nrrequisicao + ".txt";
27        StringBuilder conteudo = new StringBuilder();
28        logger.debug("Iniciando processamento da requisicao: " + nrrequisicao);
29        periodoinicial += "00";
30        periodofinal += "23";
31        String strurl = "http://www.redemet.aer.mil.br/api/consulta_automatica/
32            index.php?local="+ codcidade + "&msg=metar&data_ini=" +
33                periodoinicial + "&data_fim=" + periodofinal + "&data_hora=sim&
34                    cabecalho=nao";
35        URL url = new URL(strurl);
36        URLConnection conn = url.openConnection();
37        conn.addRequestProperty("User-Agent", "Mozilla/4.0 (compatible; MSIE
38            6.0; Windows NT 5.1)");
39        BufferedReader buffer = new BufferedReader(new InputStreamReader(conn.
40            getInputStream()));
41        String linha;
42        String vet [];
43        while ((linha = buffer.readLine()) != null) {
44            vet = linha.split(" ");
45            System.out.println(vet[2]);
46            if(vet[2].contentEquals("METAR")){
47                conteudo.append(linha + "\n");}}
48        FileWriter arq_out = new FileWriter(dir + dir_baixados + arquivo + ".
49            txt");
50        PrintWriter gravarArq = new PrintWriter(arq_out);
51        gravarArq.print(conteudo);
52        gravarArq.close();
53        logger.debug("Arquivo baixado com sucesso!");
54        JSONObject reqJson = new JSONObject().put("status", "2").put("id", id);
55        reqRest.UpdateRequisitionId(reqJson, id);
56        logger.debug("Status da requisicao atualizado para 2");

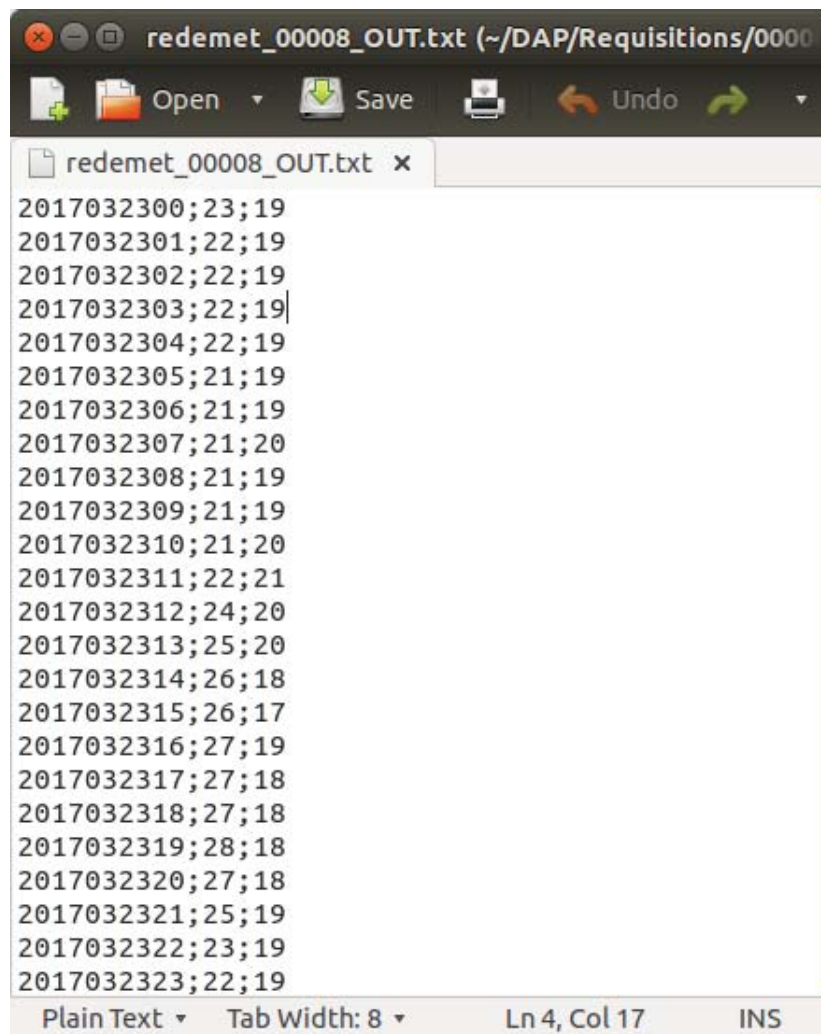
```

Tabela 8. *Script* de processamento do repositório Redemet

```

1 public static void main(String[] args) throws IOException, Exception {
2 try {
3 logger.debug("*** Recebendo parametros para processar a requisicao*");
4 String nrrequisicao = args[0];
5 String periodoinicial = args[1];
6 String periodofinal = args[2];
7 String cidade = args[3];
8 String variaveis = args[4];
9 id = Integer.parseInt(nrrequisicao);
10 logger.debug("*** Iniciando processamento da requisicao: " +
    nrrequisicao);
11 cidade = cidade.replace("+", "");
12 String arquivo = "redemet_" + nrrequisicao + ".txt";
13 boolean temp = false, porv = false;
14 String[] vars = variaveis.split("-");
15 String arquivo_novo = "hora;";
16 logger.debug("*** Verificando as variaves selecionadas na requisicao: "
    + nrrequisicao);
17 for (int i = 0; i < vars.length; i++) {
18 if (vars[i].equals("temp")) {
19 temp = true;} else if (vars[i].equals("porv")) {
20 porv = true;} }
21 arquivo_novo += "\n";
22 String dir = "//home//ronaldo//DAP//Requisitions//" + nrrequisicao + "
    //";
23 String dir_baixados = "Baixados//";
24 String dir_descopactados = "Descompactados//";
25 String dir_processados = "Processados//";
26 logger.debug("Gerando arquivo da requisicao: " + nrrequisicao);
27 FileReader arq = new FileReader(dir + dir_baixados + arquivo);
28 BufferedReader lerArq = new BufferedReader(arq);
29 String linha = lerArq.readLine();
30 linha = lerArq.readLine();
31 String vet[];
32 while (linha != null) {
33 vet = linha.split(" ");
34 arquivo_novo += vet[vet.length-2] + ";";
35 arquivo_novo += "\n";
36 linha = lerArq.readLine();}
37 arq.close();
38 String arquivo_saida = "REQ_" + nrrequisicao + "_OUT";
39 FileWriter arq_out = new FileWriter(dir + dir_processados +
    arquivo_saida + ".txt");    PrintWriter gravarArq = new PrintWriter(
    arq_out);
40 gravarArq.print(arquivo_novo);
41 gravarArq.close();
42 logger.debug("Arquivo gerado: " + dir + dir_processados + arquivo_saida
    + ".txt");    JSONObject reqJson=null;
43 reqJson = new JSONObject().put("status", 4).put("fileName",
    arquivo_saida).put("id", id);    reqRest.UpdateRequisitionId(
    reqJson, id);    logger.debug("Status da requisicao " + id + "
    alterado para 4");

```



The image shows a text editor window titled "redemet\_00008\_OUT.txt (~~/DAP/Requisitions/0000)". The window contains a list of 23 lines of text, each representing a requisition record. The records are separated by semicolons and follow a consistent format: a 10-digit ID, a two-digit number, and a two-digit number. The status of the requisition is indicated by "INS" at the bottom right of the window.

```
redemet_00008_OUT.txt x
2017032300;23;19
2017032301;22;19
2017032302;22;19
2017032303;22;19|
2017032304;22;19
2017032305;21;19
2017032306;21;19
2017032307;21;20
2017032308;21;19
2017032309;21;19
2017032310;21;20
2017032311;22;21
2017032312;24;20
2017032313;25;20
2017032314;26;18
2017032315;26;17
2017032316;27;19
2017032317;27;18
2017032318;27;18
2017032319;28;18
2017032320;27;18
2017032321;25;19
2017032322;23;19
2017032323;22;19
Plain Text Tab Width: 8 Ln 4, Col 17 INS
```

Figura 35. Arquivo de saída da requisição realizada ao repositório da Redemet





## 6. CONSIDERAÇÕES FINAIS

A utilização de dados meteorológicos em estudos aplicados as mais diversas áreas de estudos tem apresentado resultados benéficos para a sociedade. Neste sentido, o desenvolvimento de soluções computacionais que visam auxiliar o trabalho de pesquisadores e estudantes apresentam-se como ferramentas importantes para a obtenção de melhores resultados.

De acordo com Almeida et. al[2] um sistema integrado de dados meteorológicos para aquisição, armazenamento e visualização de dados é de grande interesse dos centros operacionais de previsão de tempo e também pode beneficiar universidades, em apoio as atividades de ensino e pesquisa, especialmente nos campos de análise sinótica e modelagem numérica.

Entre os principais fatores dificultantes atualmente para o acesso aos repositórios de dados meteorológicos é possível destacar a dispersão dos repositórios de dados, a falta de padronização dos repositórios de dados e dos formatos de disponibilização dos arquivos de dados.

Neste sentido, este trabalho teve como objetivo principal desenvolver uma plataforma genérica, flexível e adaptativa, que atue como um facilitador para o acesso aos dados meteorológicos disponíveis na rede mundial de computadores.

A plataforma teve a estrutura projetada para armazenar informações dos diversos repositórios de dados meteorológicos disponíveis, facilitando a localização dos mesmos quando necessário. Por meio da utilização de *scripts*, a plataforma automatiza as tarefas de aquisição e tratamento dos dados, disponibilizando-os em formatos de fácil manipulação para os usuários.

A validação do funcionamento da estrutura pode ser evidenciando no estudo de caso apresentado no Capítulo 4. Mesmo sendo um exemplo simples, foi possível perceber a aderência da solução as características esperadas para a mesma.

No que diz respeito a metodologia e arquitetura adotada para a solução, a separação em componentes possibilitou uma melhor organização da plataforma, facilitando a expansão e a manutenção da solução. A escolha da plataforma Java e das tecnologias utilizadas foram consideradas satisfatórias, tendo em vista que supriu todas as necessidades da solução.



## 7. TRABALHOS FUTUROS

Tendo em vista que a plataforma teve a estrutura projetada para suportar o registro e o acoplamento de novos repositórios de dados, para que a mesma se consolide como uma ferramenta eficaz faz se necessário a adição de mais repositórios de dados, bem como o desenvolvimento e acoplamentos dos mecanismos de aquisição e processamentos dos dados. Desta forma, pretende se criar modelos de *scripts*, nas mais diversas linguagens, para facilitar o desenvolvimento de mecanismos de *download* e processamento dos repositórios de dados.

Outro fator importante para a expansão da mesma é a ampliação do suporte aos *scripts* executados. Nesta versão inicial, a plataforma suporta somente *scripts* desenvolvidos na linguagem Java. Porém a extensão do suporte para linguagens tais como R, PHP, Shell Script, Python ampliam as possibilidades de inclusão de repositórios de dados alcançados.



## REFERÊNCIAS BIBLIOGRÁFICAS

- [1] OLIVEIRA, F. d. *100 anos de meteorologia no Brasil*. first. [S.l.]: INMET, 2009. ISBN CDU 551.1:94(81).
- [2] ALMEIDA A. L. T. FERREIRA, S. H. F. S. R. F. G. e. T. C. S. G. W. G. d. *Perspectivas de um sistema integrado de dados meteorológicos*. 2010.
- [3] NASA. *NASA*. 2016. Disponível em: <<https://pmm.nasa.gov/TRMM/mission-overview>>. Acesso em: Novembro. 11, 2017.
- [4] SENTELHAS, P. C.; MONTEIRO, J. E. B. de A. *Agrometeorologia dos cultivos*. Brasília, DF: INMET, 2009.
- [5] BERGAMASCHI, H.; MATZENAUER, R. *O milho e o clima*. first. [S.l.]: Emater/RS-Ascar, 2014. ISBN 978-85-98842-11-0.
- [6] SIVAKUMAR, M.; GOMMES, R.; BAIER, W. *Agrometeorology and sustainable agriculture. Agricultural and Forest Meteorology*, Elsevier, v. 103, n. 1, p. 11–26, 2000.
- [7] CUNHA, G. R. D.; HAAS, J. C.; ASSAD, E. D. *Zoneamento de riscos climáticos para a cultura de trigo no rio grande do sul*. 1999.
- [8] WESTPHAL, F. S. et al. *Desenvolvimento de um algoritmo para estimativa do consumo de energia elétrica de edificações não residenciais a partir de dados climáticos simplificados*. Florianópolis, SC, 2002.
- [9] LIMEIRA, R. C. *Influência de variáveis meteorológicas sobre a incidência do dengue, meningite e pneumonia em João Pessoa-PB*. *Revista Brasileira de Meteorologia*, SciELO Brasil, v. 22, n. 2, p. 183–192, 2007.
- [10] SOARES, R. V. *Desempenho da “fórmula de Monte Alegre”: índice brasileiro de perigo de incêndios florestais*. *Cerne*, v. 4, n. 1, p. 87–99, 1998.
- [11] TORRES, F. T. P. *Introdução a climatologia*. first. [S.l.]: Cengage Learning, 2016. ISBN 978-85-221-1147-3.
- [12] INMET. *INMET*. 2016. Disponível em: <<http://www.inmet.gov.br>>. Acesso em: Junho. 8, 2016.
- [13] MONTEIRO, J. E. B. *Agrometeorologia dos cultivos: o fator meteorológico na produção agrícola*. [S.l.]: Ministério da agricultura, pecuária e abastecimento (MAPA). Instituto nacional de meteorologia (INMET), 2009.

- [14] CPTEC. *CPTEC*. 2016. Disponível em: <<http://www.cptec.inpe.br>>. Acesso em: Junho. 8, 2016.
- [15] ANA. *ANA*. 2016. Disponível em: <<http://www.ana.gov.br>>. Acesso em: Junho. 8, 2016.
- [16] NOAA. *NOAA*. 2016. Disponível em: <<http://www.noaa.gov/>>. Acesso em: Junho. 8, 2016.
- [17] ROPENSCI. *ropensci*. 2016. Disponível em: <<http://ropensci.org/>>. Acesso em: Dezembro. 18, 2016.
- [18] PROJECT, R. *Language R*. 2016. Disponível em: <<https://www.r-project.org/>>. Acesso em: Dezembro. 18, 2016.
- [19] KANG, W. S.; HAN, Y. K.; KIM, S. G. A web-based information system for plant disease forecast based on weather data at high spatial resolution. *The Plant Pathology Journal*, v. 26, n. 1, p. 37–48, 2010.
- [20] OPENWEATHERMAP. *OpenWeatherMap*. 2014. Disponível em: <<http://openweathermap.org/>>. Acesso em: Novembro. 11, 2017.
- [21] COLTRI, P. P. et al. Sistema de monitoramento e previsão agrometeorológica: Agritempo. In: *CONGRESSO BRASILEIRO DE AGROMETEOROLOGIA*. [S.l.: s.n.], 2007. v. 15, p. 1.
- [22] ROMANI, L. A. S. et al. Organização do banco de dados meteorológicos do sistema agritempo. In: *IV CONGRESSO BRASILEIRO DA SOCIEDADE BRASILEIRA DE INFORMÁTICA APLICADA À AGROPECUÁRIA e À AGROINDÚSTRIA, Porto Seguro*. [S.l.: s.n.], 2003.
- [23] ALMEIDA, E.; MELO, F. Metview 2.0-um desktop meteorológico para tratamento e visualização de dados meteorológicos. *Congr. Brasil. de Meteorologia*, v. 10, n. 2000, p. 3256–3262, 2000.
- [24] FERREIRA, S. H. S.; CARVALHO, L. S. M.; FILHO, E. Ó. Banco de dados meteorológicos para previsão de tempo e estudos climáticos. In: *XI Congresso Brasileiro de Meteorologia, Rio de Janeiro*. [S.l.: s.n.], 2000.
- [25] ALMEIDA, W. G. de et al. Testes no brasil com o sistema de distribuição de dados meteorológicos pela internet (idd). 2004.
- [26] PESSOA ALEX SANDRO AGUIAR, d. A. W. G. e. C. L. Sistema de aquisição e armazenamento de dados oceânicos.
- [27] SOUSA, M. M. D.; DRUMOND, L. C.; NALDI, M. C. Artigo técnico sistema computacional para aquisição automática e disponibilização de dados meteorológicos.
- [28] SILVA, A. J. S. et al. Um sistema web para a consulta de dados meteorológicos como ferramenta de apoio ao manejo de irrigação no estado do piauí. *Rev. Tecnol. Fortaleza*, v. 29, n. 2, p. 141–147, 2008.

- [29] PRESSMAN, R.; MAXIM, B. *Engenharia de Software-8ª Edição*. [S.l.]: McGraw Hill Brasil, 2016.
- [30] JACOBI JONAS E FALLOWS, J. R. *Prof JSF e Ajax: Construindo Componentes Ricos Para a Internet*. [S.l.]: Editora Ciência Moderna Ltda., 2007.
- [31] DURELLI, V. H.; VIANA, M. C.; PENTEADO, R. A. Uma proposta de reuso de interface gráfica com o usuário baseada no padrão arquitetural mvc. *Simpósio Brasileiro de Sistemas de Informação, Rio de Janeiro–RJ, anais SBSI*, p. 48–59, 2008.
- [32] PRIMEFACES. *PrimeFaces*. 2017. Disponível em: <<https://www.primefaces.org>>. Acesso em: Novembro. 10, 2016.
- [33] GOOGLE. *Google Developers*. 2017. Disponível em: <<https://developers.google.com/maps>>. Acesso em: Novembro. 10, 2016.
- [34] LECHETA, R. R. *Web Services RESTful: Aprenda a criar web services RESTful em Java na nuvem do Google*. [S.l.]: Novatec Editora, 2015.
- [35] GUARNIERI, R. A. Emprego de redes neurais artificiais e regressão linear múltipla no refinamento das previsões de radiação solar do modelo eta. *Instituto Nacional de Pesquisas Espaciais, 171pp*, 2006.
- [36] HENRIQUE, R.; MATSCHINSKE, M. R. Meteorologia aeronáutica do sistema de controle do espaço aéreo brasileiro. *Boletim da Sociedade Brasileira de Meteorologia*, v. 29, p. 13–18, 2005.
- [37] REDEMETS. *Redemet*. 2017. Disponível em: <<http://www.redemet.aer.mil.br>>. Acesso em: Maio. 10, 2017.