

Ronaldo Aparecido da Silva

**Sistema Web de Informação Geográfica para disponibilização do Balanço
Hídrico Climatológico em Minas Gerais**

Monografia de Graduação apresentada ao
Departamento de Ciência da Computação da
Universidade Federal de Lavras como parte das
exigências do curso de Ciência da Computação para
a obtenção do título de Bacharel em Ciência da
Computação

Orientador
Prof. Dr. Pedro Castro Netto

Lavras
Minas Gerais - Brasil
2009

Ronaldo Aparecido da Silva

**Sistema Web de Informação Geográfica para disponibilização do Balanço
Hídrico Climatológico em Minas Gerais**

Monografia de Graduação apresentada ao Departamento de Ciência da Computação da Universidade Federal de Lavras como parte das exigências do curso de Ciência da Computação para a obtenção do título de Bacharel em Ciência da Computação

Aprovada em 27 de Novembro de 2009

Prof. Dr. André Vital Saúde

Prof. Dr. Antônio Carlos Fraga

Msc. Samuel Rodrigues de Sales Campos

Prof. Dr. Pedro Castro Netto
(Orientador)

Lavras
Minas Gerais - Brasil
2009

Agradecimentos

Agradeços aos meus pais Gonçalo Fransisco da Silva e Maria Neide da Silva, sem a ajuda deles jamais conseguiria completar essa jornada. Agradeço também aos amigos de antes e durante o desenvolvimento desse trabalho, amigos que me ajudaram com idéias e sugestões para a realização do projeto proposto.

Sumário

1	Introdução	1
1.1	Balanço Hídrico	2
1.2	Objetivos	3
1.2.1	Cálculo do Balanço Hídrico	3
1.2.2	Desenvolvimento da plataforma <i>WebGIS</i>	3
1.2.3	Desenvolvimento do <i>WebService</i> para o cálculo do Balanço Hídrico	3
2	Balanço Hídrico	5
2.1	Método de THORNTHWAITE e MATHER (1955)	6
2.1.1	Planilha de cálculo do balanço hídrico	7
3	Geoinformação na Internet	9
3.1	<i>Web 2.0</i>	9
3.2	<i>WebGIS</i>	10
3.2.1	<i>GIS - Geographic Information System</i>	10
3.2.2	<i>WebGIS</i>	11

4	<i>WebService REST</i>	17
4.1	<i>SOA - Service Oriented Architecture</i>	17
4.2	<i>WebServices</i>	20
4.3	<i>WebService SOAP</i>	20
4.3.1	Os protocolos WSDL e UDDI	23
4.3.1.1	WSDL - <i>Web Service Definition Language</i>	23
4.3.1.2	UDDI - <i>Universal, Description, Discovery and Integration</i>	23
4.4	<i>WebService REST</i>	23
4.4.0.3	HTTP - <i>HiperText Transfer Protocol</i>	24
4.5	<i>WebService REST - Elementos Arquiteturais</i>	26
4.5.1	<i>Resources</i>	27
4.5.2	Representação	28
4.6	Comparação de <i>WebServices REST</i> com <i>WebServices SOAP</i>	28
5	<i>Metodologia</i>	31
5.1	Balanço Hídrico Climatológico	31
5.1.1	Temperatura	31
5.1.2	Precipitação	32
5.1.2.1	Construção das matrizes	32
5.1.2.2	Importação das Matrizes para o Banco de Dados	34
5.2	<i>WebService</i>	36
5.3	<i>WebGIS</i>	38
5.3.1	Arquitetura do Oziris	38
5.3.2	Uso do <i>Framework</i>	41

6	Resultados	43
6.1	Estudo climatológico para Minas Gerais	43
6.2	Desenvolvimento de Aplicações <i>WebGIS</i>	43
6.3	Testes com o Sistema	44
6.4	Aplicação do Sistema	45
7	Conclusões	47

Lista de Figuras

3.1	Componentes de um Sistema de Informação (CAPUTI, 2006) . . .	11
3.2	Exemplo de <i>GIS</i> : ArcGIS ESRI (ESRI, 2007)	12
3.3	Exemplo de <i>GIS</i> : GRASS (GRASS, 2007)	13
3.4	Esquema básico de uma aplicação Ajax (GARRETT <i>et al.</i> , 2005) .	14
3.5	Comparação de um serviço <i>Web</i> sem Ajax com um usando Ajax (GARRETT <i>et al.</i> , 2005)	15
4.1	Esquema conceitual de SOA (ENDREI <i>et al.</i> , 2004)	19
4.2	Modelo conceitual de SOA (NGOLO, 2009)	19
4.3	Estrutura básica de uma mensagem SOAP (CURBERA <i>et al.</i> , 2002)	21
4.4	Mensagem SOAP transportada sobre HTTP (CURBERA <i>et al.</i> , 2002)	22
4.5	Formato da Request-line (FIELDING <i>et al.</i> , 1999)	24
4.6	Formato da Request-line (FIELDING <i>et al.</i> , 1999)	25
4.7	Exemplo de uma requisição HTTP com cabeçalho	25
4.8	Exemplo de comunicação consumidor/provedor (PAUTASSO, 2007)	26
4.9	Exemplos de URI para <i>Resources</i> (NUNES; DAVID, 2005)	27

5.1	Posicionamento das estações climatológicas usadas	33
5.2	Precipitação Normal Total (1961-1990) para o mês de abril.	34
5.3	Exemplo da matriz obtida para o mês de abril	35
5.4	Lavras-MG e a superfície de precipitação total normal para o mês de abril	36
5.5	Arquitetura do <i>WebService REST</i>	37
5.6	Exemplo de uso do <i>WebService</i> para proximidade de Lavras-MG .	37
5.7	Arquitetura do Oziris: Núcleo	39
5.8	Arquitetura do Oziris: Serviços	40
6.1	Precipitação normal total para o mês de abril	44
6.2	Precipitação normal total para o mês de dezembro	45
6.3	Tela do sistema em uso.	46

Lista de Tabelas

2.1	Tabela com o cálculo do balanço hídrico para Campina Grande, PB (PEREIRA, 2005).	7
4.1	Tempo (em segundos) de 1000 requisições (NEWMARCH, 2005).	29
4.2	Tamanho em bytes de uma requisição (NEWMARCH, 2005). . . .	29
6.1	Resultados dos testes de usabilidade do sistema proposto.	45

Resumo

A obtenção de informações climatológicas para ajudar na tomada de decisão no planejamento agrícola é uma grande dificuldade encontrada pelos setores ligados à agricultura, dentre essas informações está o Balanço Hídrico Climatológico. O Balanço Hídrico é uma importante ferramenta que o agricultor tendo acesso poderá ter melhores chances de prever a disponibilidade de recursos hídricos em sua localidade e assim aumentar o seu índice de sucesso. O sistema proposto nesse trabalho utiliza de tecnologias como a Internet para facilitar o acesso dos setores produtores no estado de Minas Gerais a essa informação.

Palavras-Chave: WebGIS, WebMapping, Webservice, WebDesktop, REST, Balanço hídrico, Plantas Oleaginosas, Biodiesel

Capítulo 1

Introdução

O biodiesel é um combustível produzido a partir de óleos vegetais ou de gorduras animais. Dezenas de espécies vegetais presentes no Brasil podem ser usadas na produção do biodiesel, entre elas soja, dendê, girassol, babaçu, amendoim. É possível também, usar mais de uma fonte vegetal no mesmo biodiesel (ANP, 2008).

Da produção atual de biodiesel mais de 85% tem como origem o óleo vegetal produzido a partir de culturas como soja e algodão (ANP, 2009). Essas culturas são tradicionalmente feitas por grandes produtores rurais, indicando que produtores rurais de pequeno e médio porte não tem uma grande participação no fornecimento de matéria prima para o biodiesel.

Outra questão que também deve ser considerada é a finalidade das culturas, por exemplo, em culturas como as de soja que o foco maior é a produção de alimentos, a divisão da produção para atender o mercado de biodiesel poderá ser usada como argumentação para aumento nos preços de gêneros alimentícios. Então seria interessante que a base produtiva do biodiesel pudesse ter como origem outras oleaginosas que não sofram esse tipo de conflito.

A partir de janeiro de 2010 será obrigatório no Brasil a adição de 5% de biodiesel no diesel mineral. Até então a mistura obrigatória é de 4% e, com esse aumento, serão necessários mais 2,4 bilhões de litros de biodiesel (RIBEIRO, 2009).

Para atender a demanda crescente por biodiesel, sem afetar outros segmentos, será necessário que o país tenha outras fontes sustentáveis de óleos vegetais que, além de atender as exigências governamentais, possam ser produzidas por produtores rurais de pequeno e médio porte para que haja distribuição de renda e valorização do setor rural. Mas algumas perguntas vêm à mente, quais plantas oleaginosas podem ser usadas, onde elas devem ser cultivadas e quando elas devem ser cultivadas. Para ajudar a responder essas perguntas o balanço hídrico é uma importante ferramenta para auxiliar no planejamento agrícola (CAPUTI, 2006).

1.1 Balanço Hídrico

O balanço hídrico é a contabilização das quantidades de água que entram e saem de uma certa porção do solo em um determinado intervalo de tempo. O resultado é a quantidade líquida de água que nele permanece disponível às plantas (TOMASELLA; ROSSATO, 2005). O balanço hídrico pode ser determinado de forma bastante simples desde que se tenha acesso a dados como temperatura e precipitação (DANTAS; CARVALHO; FERREIRA, 2007).

Uma dificuldade no cálculo do balanço hídrico é a aquisição desses dados, pois esses são obtidos em estações experimentais e são pontuais, o que dificulta a aplicação deles em outras localidades que não as mesmas de onde os dados foram coletados. Além disso, a disponibilidade de dados climatológicos é pequena devido ao pequeno número de estações climatológicas instaladas no território brasileiro. Como exemplo, são disponibilizados os valores das Normais Climatológicas pelo Instituto Nacional de Meteorologia para apenas 30 localidades em todo o território do Estado de Minas Gerais, que apresenta 853 municípios

1.2 Objetivos

Este trabalho teve como objetivo propor um sistema *web* auxiliar no planejamento agrícola de propriedades situadas no estado de Minas Gerais, utilizando uma interface amigável, onde os responsáveis pelo planejamento agrícola possam facilmente realizar o cálculo do balanço hídrico e prontamente ter informações relevantes para ajudar na tomada de decisão.

1.2.1 Cálculo do Balanço Hídrico

Para a realização do cálculo do balanço hídrico foi utilizado o modelo Thornthwaite e Mather (1955), pois este modelo é o comumente usado devido as suas características como por exemplo a previsão da variação temporal do armazenamento de água no solo (NAGAOKA *et al.*, 2008).

1.2.2 Desenvolvimento da plataforma *WebGIS*

Para a interação de usuário com o sistema foi adotada uma interface *WebGIS*. Esse tipo de interface foi usada pois combina visualização de dados tabulares, análise espacial, visualização de mapas e sistema gerenciador de banco de dados, com o objetivo de ajudar o usuário do sistema a entender as relações entre as entidades existentes no problema como também ajudá-lo na tomada de decisão (YINGWEI; XIAOLIN; ZHUOQUN, 2001).

1.2.3 Desenvolvimento do *WebService* para o cálculo do Balanço Hídrico

Para que a interface *WebGIS* funcione é necessário uma central que forneça os serviços e dados para o processamento do balanço hídrico. Essa central foi um *WebService REST*. Um *WebService REST* é um conjunto de *URIs* que através de métodos *HTTP* (como **POST**, **GET** e **DELETE** por exemplo) pode realizar um conjunto de operações e retornar o resultado de diferentes formas para quem o usa (KOPECKY; GOMADAM; VITVAR, 2008).

Capítulo 2

Balanço Hídrico

As atividades agrícolas estão sempre expostas às variações de elementos climáticos, dentre esses elementos podemos destacar a precipitação que possui uma grande variabilidade, e o sucesso ou insucesso de um empreendimento agrícola está diretamente relacionado à disponibilidade de água no solo (DANTAS; CARVALHO; FERREIRA, 2007).

O método de cálculo do balanço hídrico climatológico desenvolvido por Thornthwaite & Matter (1955) é uma maneira de se prever a variação do armazenamento de água no solo. Partindo-se do suprimento natural de água ao solo, da demanda atmosférica, e da capacidade de água disponível, o balanço hídrico fornece estimativas da evapotranspiração real, da deficiência e excedente hídrico e do armazenamento de água no solo (CAPUTI, 2006).

O conceito central do balanço hídrico é considerar o solo como um reservatório de capacidade fixa, onde a água é armazenada até o máximo da capacidade de campo, e será retirada somente pela ação das plantas (CAPUTI, 2006).

Alguns conceitos importantes para o entendimento do cálculo do balanço hídrico:

Evapotranspiração: É o processo total de transferência de água do sistema solo-planta para a atmosfera, envolvendo evaporação direta da água do solo e depositada na superfície das plantas e transpiração das plantas (DANTAS; CASTRO NETO, 2003).

Evapotranspiração potencial: Significa a água transferida para a atmosfera a partir de uma extensa superfície coberta por vegetação baixa e uniforme, em crescimento ativo e cobrindo totalmente o terreno bem suprido de umidade. Ou seja, em nenhum instante a demanda atmosférica é restringida por falta d'água no solo (TOMASELLA; ROSSATO, 2005).

Evapotranspiração Real: É aquela que ocorre numa superfície vegetada, independente de sua área, de seu porte e das condições de umidade do solo. Portanto, é aquela que ocorre em qualquer circunstância. (TOMASELLA; ROSSATO, 2005).

Deficiência hídrica: É a diferença entre a evapotranspiração potencial e a real (CAPUTI, 2006).

Excedente hídrico: É a diferença entre a precipitação e a evapotranspiração potencial, quando o solo atinge a sua capacidade máxima de retenção de água (CAPUTI, 2006).

Capacidade de Campo: É o limite máximo de armazenamento de água disponível no solo (TOMASELLA; ROSSATO, 2005).

2.1 Método de THORNTHWAITE e MATHER (1955)

(THORNTHWAITE, 1948) propôs o primeiro método para o cálculo do balanço hídrico com fins climatológicos. Para o cálculo, considerou que toda a entrada de água no solo se dava pela precipitação (**P**) e toda saída pela evapotranspiração potencial (**EP**), tendo o solo capacidade para armazenar 100 mm de água disponível.

Em 1955, (THORNTHWAITE; MATHER, 1955) propuseram um novo roteiro para o cálculo do Balanço Hídrico, incluindo algumas modificações:

1. A demanda de água pela atmosfera é dada pela **EP**, sendo a resposta do sistema solo-planta dada pela evapotranspiração real (**ER**). Neste caso, considera-se um decréscimo da relação **ER/EP** à medida que decresce o teor de água no solo (DANTAS; CASTRO NETO, 2003).

2. A capacidade de água disponível no solo passou a variar em função da cultura com a qual se está trabalhando. Considera-se para este caso o teor de água no solo entre a capacidade de campo e o ponto de murcha permanente na profundidade do solo onde se encontra pelo menos 80% do sistema radicular da cultura. Deve-se pensar que, de acordo com a textura do solo, este terá maior ou menor capacidade de retenção de água, o que será compensado pelo fato de que o sistema radicular das plantas explora maior volume de solo em solos arenosos, que possuem menor capacidade de retenção de água da cultura (profundidade efetiva do sistema) (DANTAS; CASTRO NETO, 2003).

2.1.1 Planilha de cálculo do balanço hídrico

O resultado do cálculo do balanço hídrico pode ser expresso como uma tabela onde cada linha representa a contabilização das entradas e perdas de água para cada mês do ano, a Tabela 2.1 mostra um exemplo.

MÊS	P	EP	P-EP	Neg Acum	ARM	ALT	ER	DEF
JAN	41	108	-67	-408	5	-3	44	64
FEV	55	109	-54	-462	3	-2	57	52
MAR	100	115	-15	-477	3	0	100	15
ABR	129	107	+22	-201	25	+22	107	0
MAI	95	95	0	-201	25	0	95	0
JUN	107	80	+27	-110	52	+27	80	0
JUL	124	62	+62	-12	114	+62	62	0
AGO	58	78	-20	-32	97	-17	75	3
SET	38	77	-39	-71	71	-26	64	13
OUT	17	102	-85	-156	36	-35	52	50
NOV	19	108	-89	-245	18	-18	37	71
DEZ	21	117	-96	-341	8	-10	31	86

Tabela 2.1: Tabela com o cálculo do balanço hídrico para Campina Grande, PB (PEREIRA, 2005).

A Tabela 2.1 é composta dos seguintes dados para os 12 meses normais: precipitação (**P**), Evapotranspiração Potencial (**P**), Diferença entre P e EP (**E-EP**), Negativa Acumulada (**Neg Acum**), Armazenamento de água no solo (**ARM**), Vari-

ação entre o Armazenamento do mês atual em relação ao mês anterior (**ALT**),
Evapotranspiração Real (**ER**) e Deficiência (**DEF**)

Capítulo 3

Geoinformação na Internet

A Internet é uma poderosa ferramenta para o compartilhamento de informações e serviços, há diversas formas de apresentá-los aos usuários, como também há diversas técnicas e/ou tecnologias para que os usuários possam interagir. Neste capítulo são apresentadas conceitos comumente usados para a interação do usuário com a geoinformação através da Internet.

3.1 *Web 2.0*

A *World Wide Web (WWW)* que também é conhecida como *Web* é uma estrutura de documentos interligados que são acessados através da Internet (SHAH, 2009).

O termo *Web 2.0* é usado para classificar um novo conjunto de serviços *on-line* que permita aos usuários uma forma de interação mais rica. Esses serviços são caracterizados por simularem no ambiente *Web* os conteúdos e interfaces típicos do ambiente *desktop*¹, ou seja, uma das idéias de um serviço *Web 2.0* é tornar a experiência do usuário ao usar um serviço *Web* mais próxima daquela que ele experimentaria ao usar um serviço semelhante no ambiente *desktop*.

¹Nesse texto esse termo é usado para indicar aplicações que são executadas apenas no computador do usuário.

Os usuários da *Web* até o surgimento dos serviços *Web 2.0* eram chamados de navegantes, ou internautas, com os novos serviços oferecidos passaram a dispensar a instalação de aplicativos em seus computadores. O usuário agora passará a interagir, selecionar e controlar as informações para ampliar o seu papel na *Web* (MEIRELLES; MOURA, 2007).

(PRIMO, 2007) informa que a *Web 2.0* tem repercussões sociais importantes, pois potencializam o trabalho coletivo, produção e circulação de informação e conhecimento.

Os serviços *on-line* oferecidos como *Web 2.0* oferecem inúmeras possibilidades os usuários. Um usuário possuindo apenas um *WebBrowser*, atualizado e que suporte aos padrões W3C (*World Wide Web Consortium*), pode acessar uma grande gama de serviços, dos mais variados tipos e finalidades, de um único ponto de acesso sem grandes custos ou a necessidade de manter vários softwares atualizados.

Agora os usuários da *Web* têm acesso a serviços mais poderosos no contexto da *Web*, e mais acessíveis financeiramente. A criação de serviços de geoinformações agora é mais interessante e prática.

3.2 *WebGIS*

Um *WebGIS* é um *GIS* para *Web*, mas o que é um *GIS*?

3.2.1 *GIS - Geographic Information System*

Antes de definir um *GIS* é importante informar que esse tipo de aplicação é um tipo especial de sistema de informação.

Um sistema de informação segundo (CAPUTI, 2006) pode ser definido como um conjunto de elementos inter-relacionados que interagem para coletar, processar, armazenar, recuperar e distribuir a informação com a finalidade de facilitar o planejamento, o controle, a coordenação, a análise e o processo decisório em empresas e outras organizações.

Um sistema de informação de acordo com (WORBOYS; DUCKHAM, 2004) é uma associação de pessoas, computadores, dados, procedimentos trabalhando em conjunto para coletar, gerenciar e distribuir informações importantes para indivíduos ou organizações.

Um sistema de informação é composto por três componentes, com a finalidade de converter dados em informação. Uma característica necessária é a retroalimentação, pois nesta atividade o sistema é subsidiado com informações do próprio sistema além de ajudar no refinamento da entrada de dados (CAPUTI, 2006), os componentes podem ser vistos na Figura 3.1.

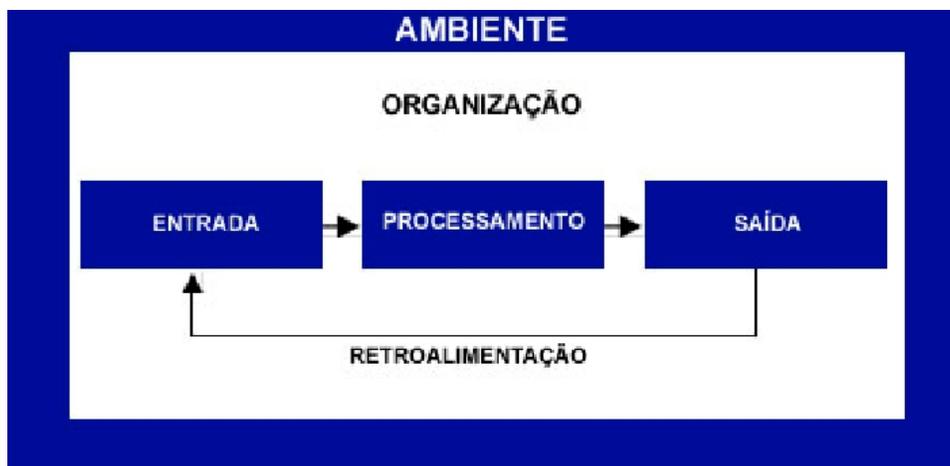


Figura 3.1: Componentes de um Sistema de Informação (CAPUTI, 2006)

Um *GIS* é um sistema de informação que permite capturar, modelar, armazenar, recuperar, compartilhar, manipular, analisar e apresentar dados geograficamente referenciados (WORBOYS; DUCKHAM, 2004). As Figuras 3.2 e 3.3 mostram duas telas de dois *GIS desktop*.

3.2.2 *WebGIS*

WebGIS é um tipo de aplicação *Web 2.0* que faz a integração de um *GIS* com a Internet. Um *WebGIS* transfere para a *Web* informação geográfica provendo funcionalidades de análise espacial (TAN *et al.*, 2008).

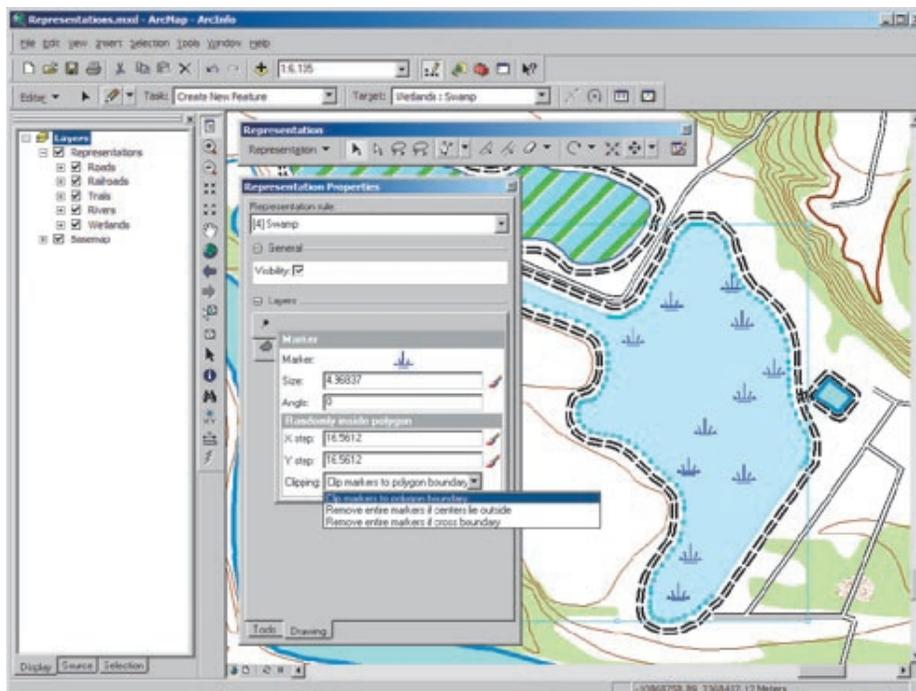


Figura 3.2: Exemplo de GIS: ArcGIS ESRI (ESRI, 2007)

Um problema para as aplicações *WebGIS* está na diversidade de sistemas operacionais, *WebBrowsers*, largura de banda e clientes que podem acessá-los. Esses problemas tornam difícil a adoção de uma solução *WebGIS* para o trabalho no campo profissional onde os aplicativos *GIS desktop* são utilizados (DALI; JIX-IAN, 2001).

Na atualidade existem algumas tecnologias para o desenvolvimento de serviços *WebGIS*, uma delas é utilizando o Adobe Flex (ADOBE, 2009) como mostrado por (WANG *et al.*, 2008) ou utilizando tecnologias *standards* como: Javascript, XML, DHTML, CSS e XMLHttpRequest (TAN *et al.*, 2008).

Para o desenvolvimento de um serviço *WebGIS* usando as tecnologias *standards*, o cerne do desenvolvimento está no uso de Asynchronous Javascript And XML (Ajax).

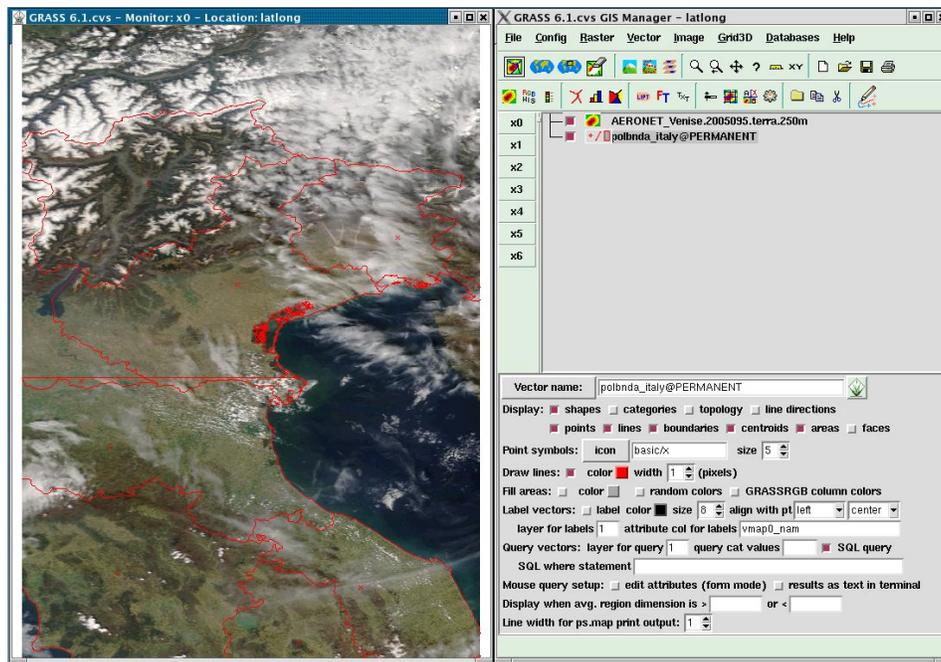


Figura 3.3: Exemplo de GIS: GRASS (GRASS, 2007)

Ajax não é definido como uma nova tecnologia, nem como um novo conjunto de tecnologias, na realidade o termo Ajax é usado para indicar o uso de várias tecnologias já existentes e que combinadas fornecem uma plataforma para o desenvolvimento de serviços *Web* (GARRETT *et al.*, 2005). Os principais grupos de tecnologias usadas em Ajax são:

Apresentação Tecnologias como HTML e CSS.

Interação *Document Object Model*, usado para modificar os elementos visuais na interface com o usuário como também captura de eventos.

Intercâmbio de dados Uso intensivo de JSON e XML.

Integração Javascript, responsável por fazer todos os elementos anteriores trabalharem em conjunto.

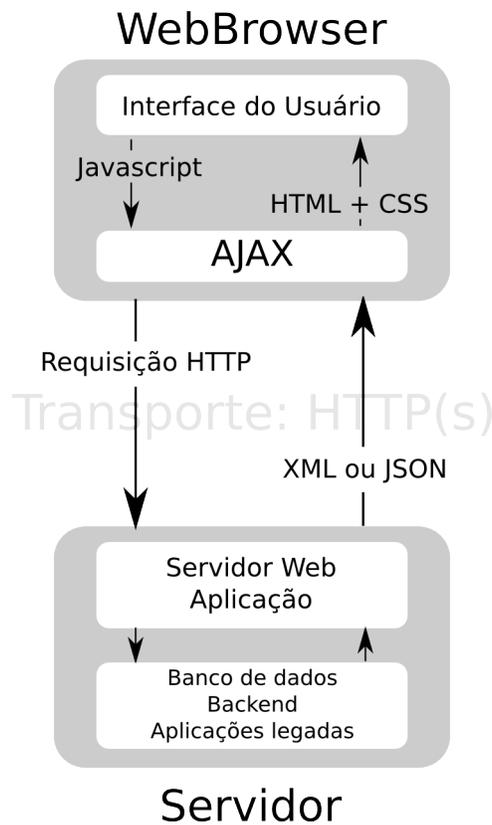
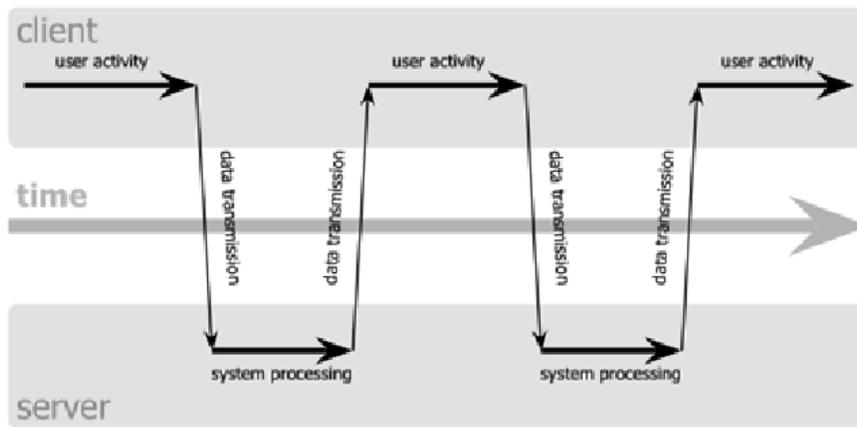


Figura 3.4: Esquema básico de uma aplicação Ajax (GARRETT *et al.*, 2005)

Como pode ser notado pela Figura 3.4 o uso da estratégia Ajax para o desenvolvimento de um serviço *Web* traz a vantagem de ser possível a transferência assíncrona de dados entre o cliente e o servidor. Essa característica é de grande importância para o domínio de serviços *WebGIS*, pois normalmente o volume de dados é grande e um esquema síncrono de transferência de dados pode dificultar ou até inviabilizar o serviço.

De acordo com a Figura 3.5 é possível notar que durante o tempo em que um usuário de um *WebGIS* estiver usando-o, uma solicitação de recursos e/ou submissão de dados poderá ser feita a qualquer momento sem que o usuário obrigatoriamente tenha que interromper suas atividades esperando o resultado do processamento anterior.

classic web application model (synchronous)



Ajax web application model (asynchronous)

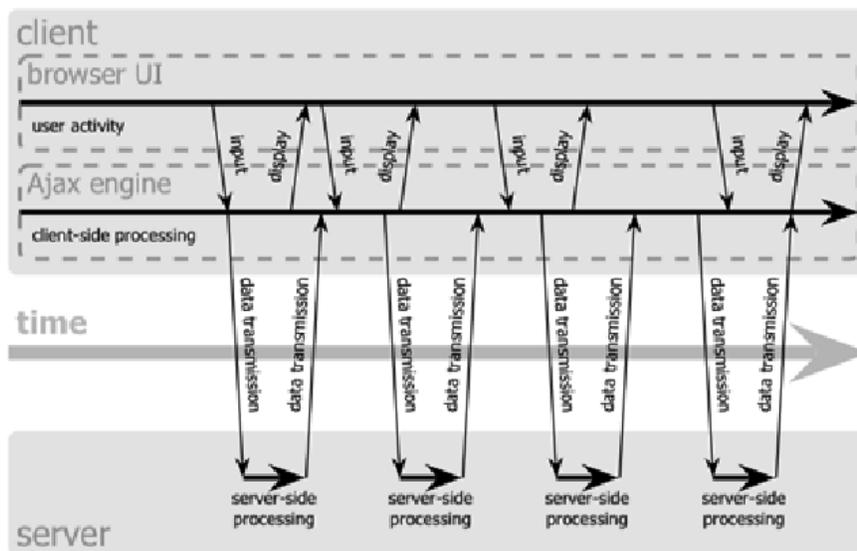


Figura 3.5: Comparação de um serviço *Web* sem Ajax com um usando Ajax (GARRETT *et al.*, 2005)

Capítulo 4

WebService REST

Um *WebService* pertence a uma família de sistemas computacionais que tem o objetivo de prover serviços para outros sistemas computacionais. Nesse capítulo serão mostrados alguns conceitos relativos a essa família de sistemas computacionais e também sobre *WebServices* e finalmente *WebServices REST*.

4.1 SOA - *Service Oriented Architecture*

A expressão “crise do software” foi criada na década de 1960 e fala da qualidade do software, não somente sobre sistemas que não funcionam corretamente. A expressão também é usada para se referir a outros problemas (REZENDE, 2006), como:

- Como um sistema é construído.
- Como um sistema é implantado, nesse caso pode ser a forma como é feita a substituição de um sistema antigo já em operação por um novo, ou a implantação de um sistema totalmente novo.
- Como é provida a manutenção de um sistema já em uso, visando atender as demandas esperadas e como isso pode ser feito levando em conta a complexidade crescente dos sistemas e o aumento da demanda.

Na tentativa de solucionar essas questões a indústria de Tecnologia de Informação (TI) criou várias estratégias e uma delas é a SOA (ENDREI *et al.*, 2004).

SOA é um paradigma de sistemas de informação que trás grandes vantagens, pois nesse paradigma sistemas já estabelecidos podem ser usados de diferentes formas para aumentar a eficiência no uso deles sem que os mesmos sejam alterados (NGOLO, 2009).

O funcionamento de um sistema desenvolvido sobre SOA como mostrado por (ENDREI *et al.*, 2004) é feito pelos seguintes componentes:

Service São entidades lógicas, é uma espécie de contrato definido por um conjunto de interfaces conhecidas.

Service provider Um software que implementa um serviço.

Service consumer Um software que usa um provedor, também chamado de cliente. O Consumidor pode ser também um outro provedor.

Service locator Um tipo de *Service provider* responsável por localizar outro *Service provider*.

Service broker Outro tipo de *Service provider* responsável por fazer a distribuição das requisições de serviço para um ou mais *Service provider*.

As interfaces do *Service* são as definições do que um serviço pode fazer, como deve ser chamado e outros detalhes para que ele seja consumido por um *Service consumer* corretamente.

Com o uso do conceito de serviços é possível implementar aplicações como um conjunto de serviços que interagem entre si (NGOLO, 2009). Com o uso de SOA é possível construir sistemas que possam prover funcionalidades para usuários finais e também para outros serviços (ENDREI *et al.*, 2004).

A Figura 4.1 e a Figura 4.2 mostram um modelo conceitual de um conjunto de sistemas construídos sobre SOA.

Existem várias tecnologias que implementam SOA entre elas estão: CORBA, DCOM, GRID Computing e WebServices (NGOLO, 2009).

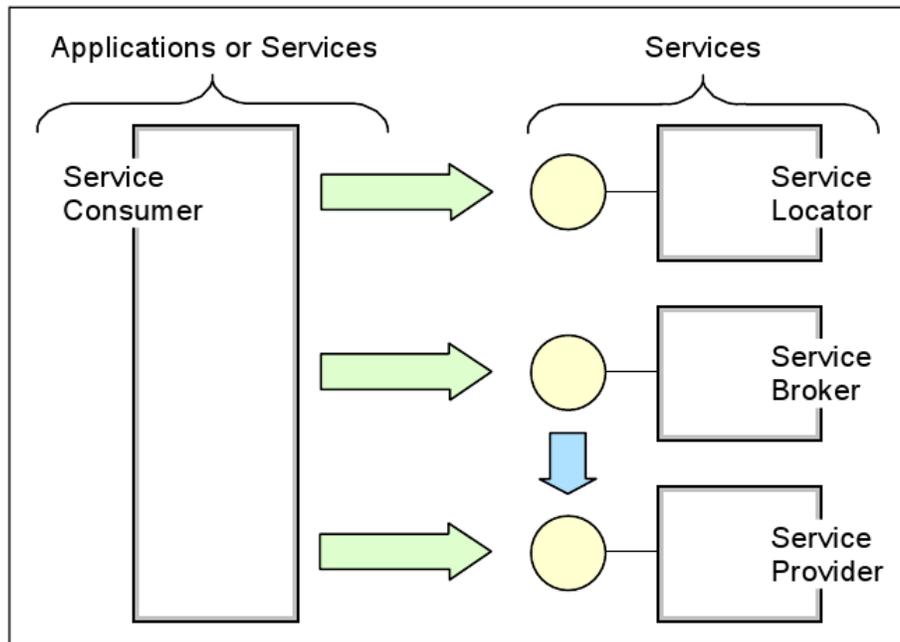


Figura 4.1: Esquema conceitual de SOA (ENDREI *et al.*, 2004)

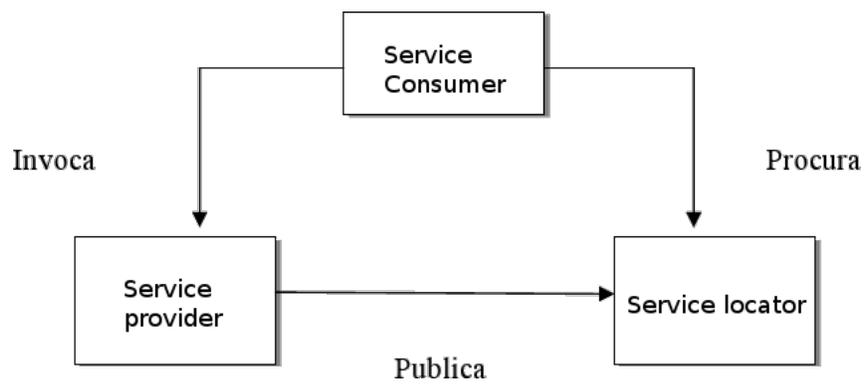


Figura 4.2: Modelo conceitual de SOA (NGOLO, 2009)

4.2 WebServices

Um WebService é uma aplicação que tem a capacidade de auto-descrição e está disponível na internet de forma a ser acessível por toda a rede (ARAÚJO, 2005).

Um WebService também permite que sistemas heterogêneos se comuniquem por meio de troca de mensagens (NGOLO, 2009).

Como um WebService procura fazer a interligação de vários sistemas, e esses sistemas podem ser construídos sobre plataformas tecnológicas diferentes, é necessário que exista um protocolo de comunicação (ARAÚJO, 2005).

Na atualidade há dois grandes padrões para a construção de WebServices de acordo com (MURACEVIC; KURTAGIC, 2009):

SOAP - *Simple Object Access Protocol* Protocolo baseado em *eXtensible Markup Language* (XML) mantido por (W3C, 2000).

REST - *Representational State Transfer* Modelo idealizado por (FIELDING, 2000) que usa ao máximo o protocolo *HyperText Transfer Protocol* (HTTP) (FIELDING *et al.*, 1999).

4.3 WebService SOAP

O ambiente Web é intrinsecamente distribuído e por natureza heterogêneo, os mecanismos de comunicação com os WebServices devem ser independentes de plataforma tecnológica, seguros e rápidos. O XML é conhecido por sua grande capacidade de codificar dados independente de plataforma, internacionalização e força para a transferência de dados. Por essas virtudes do XML ele foi escolhido como base para a criação do protocolo de comunicação SOAP (CURBERA *et al.*, 2002).

SOAP é uma linguagem XML usada para descrever mensagens trocadas entre aplicações. É baseada em regras de XML Schema e XML Namespace, o que por consequência oferece independência em relação aos sistemas operacionais usados e linguagens de programação (ARAÚJO, 2005).

Como o SOAP define um novo protocolo de comunicação então ele pode ser usado sobre outros protocolos de comunicação como o HTTP (CURBERA *et al.*, 2002).

Um mensagem SOAP é composta por três elementos:

Envelope Em XML é necessária a existência de um elemento principal ou elemento raiz (W3C, 2008). Em uma mensagem SOAP esse elemento é o Envelope.

Header O Header é um mecanismo padrão para extensão da mensagem, que pode ser usado para tratar de questões como autenticação ou gerenciamento de transações (W3C, 2000).

Body É o conteúdo da mensagem propriamente dito.

A Figura 4.3 mostra a estrutura básica de uma mensagem SOAP.

```
<SOAP:Envelope xmlns:SOAP=
  "http://schemas.xmlsoap.org/soap/envelope/">

  <SOAP:Header>
    <!-- content of header goes here -->
  </SOAP:Header>

  <SOAP:Body>
    <!-- content of body goes here -->
  </SOAP:Body>

</SOAP:Envelope>
```

Figura 4.3: Estrutura básica de uma mensagem SOAP (CURBERA *et al.*, 2002)

Uma das formas mais usuais do SOAP é no contexto de *Remote Procedure Call* (RPC)¹ (ARAÚJO, 2005).

¹Tecnologia usada para que um processo em um sistema execute rotinas em outros sistemas

Para o uso de SOAP-RPC o conteúdo da mensagem SOAP deve descrever quais tipos podem ser trocados entre o consumidor e o provedor do serviço, além de informar qual operação será executada e quais são os parâmetros de entrada (CURBERA *et al.*, 2002). A Figura 4.4 mostra um exemplo de mensagem SOAP transportada através do protocolo HTTP.

```
POST /travelservice
SOAPAction: "http://www.acme-travel.com/flightinfo"
Content-Type: text/xml; charset="utf-8"
Content-Length: nnnn

<SOAP:Envelope xmlns:SOAP=
"http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP:Body>
    <m:GetFlightInfo
      xmlns:m="http://www.acme-travel.com/flightinfo"
      SOAP:encodingStyle=
        "http://schemas.xmlsoap.org/soap/encoding/"
      xmlns:xsd="http://www.w3.org/2001/XMLSchema"
      xmlns:xsi=
        "http://www.w3.org/2001/XMLSchema-instance">

      <airlineName xsi:type="xsd:string">UL
    </airlineName>

      <flightNumber xsi:type="xsd:int">506
    </flightNumber>

    </m:GetFlightInfo>
  </SOAP:Body>
</SOAP:Envelope>
```

Figura 4.4: Mensagem SOAP transportada sobre HTTP (CURBERA *et al.*, 2002)

4.3.1 Os protocolos WSDL e UDDI

Um Webservice SOAP é dividido em três áreas: comunicação, descrição e localização (CURBERA *et al.*, 2002). O protocolo SOAP como mostrado é responsável pela comunicação.

4.3.1.1 WSDL - *Web Service Definition Language*

O protocolo WSDL (W3C, 2001) pode ser entendido como um vocabulário escrito em XML usado para descrever um Webservice (NGOLO, 2009).

As informações seguintes estão descritas em um documento WSDL (ARAÚJO, 2005):

- Informações sobre quais métodos o serviço oferece.
- Os tipos de dados de entrada associados aos métodos e os tipos de saída.
- Informação sobre qual protocolo que deverá ser usado para invocar o serviço.
- Endereço do Serviço para que o cliente possa usá-lo.

4.3.1.2 UDDI - *Universal, Description, Discovery and Integration*

O UDDI(OASIS, 2002) também é protocolo baseado em XML usado para criar um sistema unificado de registro, publicação e localização de Webservices (CURBERA *et al.*, 2002). Esse protocolo é controlado pelo consórcio OASIS.

4.4 *Webservice REST*

O conceito de uma arquitetura REST foi definido primeiramente por (FIELDING, 2000), em seu trabalho REST é definido como um estilo de arquitetura para sistemas hipermídia distribuídos. A arquitetura REST tem levantado muito interesse da comunidade devido a sua simplicidade e facilidade de uso (NGOLO, 2009).

A arquitetura REST toma como ponto de partida o ambiente Web, nesse contexto um Webservice REST faz uso intensivo de tecnologias usadas nativamente na Web como HTTP, XML, HTML e XHTML (NGOLO, 2009).

Uma diferença entre um Webservice REST e um Webservice SOAP está na comunicação entre o consumidor do serviço e o provedor. Ao contrário do SOAP que define uma nova camada, que normalmente é usada sobre o HTTP, um Webservice REST usa somente o HTTP como protocolo de comunicação. De acordo com (VINOSKI, 2008) o protocolo HTTP é suficientemente poderoso para tratar de questões como: localização, formato da mensagem e operações que se deseja usar.

Como um Webservice REST não obedece ao paradigma SOAP ele não usa protocolos como WSDL e UDDI. São usadas somente as diretivas primárias do protocolo HTTP (NGOLO, 2009).

4.4.0.3 HTTP - *HiperText Transfer Protocol*

O HTTP é protocolo de transporte mais utilizado na Internet. Principalmente por sua facilidade de compreensão, já que é baseado em texto e também por estar implementado virtualmente em todas as plataformas (NGOLO, 2009).

Quando um cliente inicia uma conexão com um servidor, por exemplo uma conexão iniciada por um *WebBrowser*, o protocolo HTTP define que a primeira linha deve trazer informações sobre qual método HTTP o cliente deseja utilizar, qual é a *Uniform Resource Identifier (URI)*² do recurso solicitado e a versão do protocolo que deve ser usada. O método HTTP indica que operação deve ser realizada sobre o recurso identificado pela URI (FIELDING *et al.*, 1999). A Figura 4.5 mostra o formato de uma linha de requisição e a Figura 4.6 mostra um exemplo de requisição usando o método POST.

```
Method SP Request-URI SP HTTP-Version CRLF
```

Figura 4.5: Formato da Request-line (FIELDING *et al.*, 1999)

²Define a localização de algum recurso dentro de um servidor

Na Figura 4.5 **SP** indica caracter de espaço e **CRLF** é a combinação dos caracteres de retorno de carro e quebra de linha.

```
POST /noticias HTTP/1.1
```

Conteúdo de alguma notícia que deverá ser gravado no servidor como "filho" da URI /noticias.

Figura 4.6: Formato da Request-line (FIELDING *et al.*, 1999)

Também é definido no protocolo HTTP uma seção reservada para transportar informações sobre as requisições como formato da mensagem, identificação do cliente e data e hora da modificação do conteúdo da URI. Cada cabeçalho consiste de uma linha onde o nome do cabeçalho é seguido por dois pontos “:” e pelo valor do cabeçalho, a Figura 4.7 mostra um cabeçalho informando ao servidor que é esperado um arquivo texto puro de uma URI através do método GET.

```
GET /noticias HTTP/1.1  
Accept: text/plain
```

Figura 4.7: Exemplo de uma requisição HTTP com cabeçalho

O Protocolo HTTP fornece 4 métodos básicos de interação com um servidor HTTP, esses métodos são: *PUT*, *GET*, *POST* e *DELETE* (FIELDING *et al.*, 1999).

A semântica de cada um dos 4 métodos apresentados é mostrada abaixo (MURACEVIC; KURTAGIC, 2009):

PUT Usado para armazenar uma entidade sob uma determinada URI.

GET Usado para recuperar qualquer informação identificada por uma URI.

POST Usado para indicar que a entidade enviada para o servidor de origem deve ser considerada como “filha” do recurso indentido por uma URI.

DELETE Usado para indicar que a entidade identificada por uma URI deve ser removida.

Como mostrado por (MURACEVIC; KURTAGIC, 2009), esses 4 métodos podem ser facilmente associados com as operações comuns em sistemas de informação, essas operações são: *Create*, *Retrieve*, *Update* e *Delete* (CRUD).

A Figura 4.8 mostra como é feita a utilização do protocolo HTTP para realizar a comunicação entre o consumidor e o provedor de um serviço.

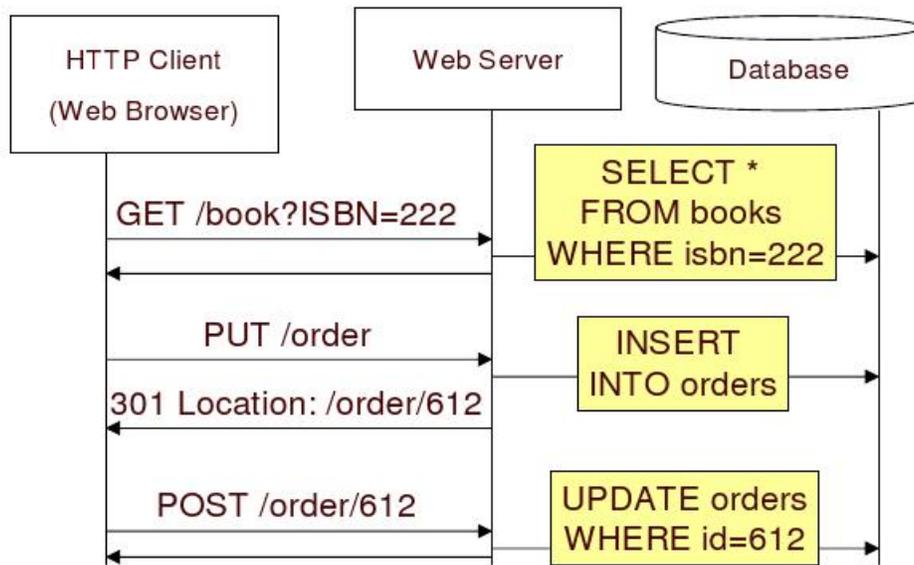


Figura 4.8: Exemplo de comunicação consumidor/provedor (PAUTASSO, 2007)

4.5 Webservice REST - Elementos Arquiteturais

Um Webservice REST é baseado em quatro princípios (PAUTASSO; ZIMMERMANN; LEYMAN, 2008):

Identificação de *Resources* através de URI: Uma URI é usada para identificar algum objeto/recurso em servidores *Web*, logo para identificar algum *Resource* do Webservice é feito o uso de URI.

Interface Uniforme: Todas as operações de WebService são descritas como operações de CRUD usando os métodos HTTP PUT, GET, POST e DELETE. A informação recuperada ou enviada para o servidor indica o estado do *Resource*.

Mensagens auto-descritivas: Os Resources são desacoplados de sua representação, ou seja, o conteúdo de uma Resource pode ser obtido de várias maneiras como por exemplo: XHTML, JSON e XML.

Estados são mantidos por links: Toda comunicação feita entre o provedor do serviço e o consumidor não possui estado, toda a informação necessária para a compreensão de uma mensagem está contida nela mesma. Para o caso de ser necessária a existência de estado são usadas técnicas para injetar essas informações na conversação entre provedor e consumidor como Cookies e Hidden fields.

Os elementos centrais de um WebService são os *Resources* e *Representations*.

4.5.1 Resources

Os *Resource* são as entidades de interesse em um WebService, são os objetos sobre os quais os serviços oferecidos por um provedor serão realizados. A identificação de um *Resource* é feita através de uma URI. A Figura 4.9 mostra alguns exemplos de URI que identificam alguns *Resources*.

```
...  
http://univ.example.com/studentInfo/123456787  
http://univ.example.com/studentInfo/123456788  
http://univ.example.com/studentInfo/123456789  
...
```

Figura 4.9: Exemplos de URI para *Resources* (NUNES; DAVID, 2005)

Um Resource ao longo do tempo pode ter vários estados (FIELDING, 2000). O estados do *Resource* podem ser resultados das operações feitas sobre ele. Esses estados podem representar a informação contida em um *Resource*.

4.5.2 Representação

Para cada estado de um *Resource* há um conjunto de Representações. Uma representação é uma sequência de bytes com os dados contidos em um estado do *Resource* e também com metadados sobre os dados (FIELDING, 2000). Em outras palavras, é a forma como o consumidor do *WebService* deve tratar a resposta de um serviço, através do protocolo HTTP o resultado de uma requisição para um *Resource* traz informações sobre o que foi devolvido como resposta, essa resposta pode ser uma imagem, um documento XML e outro tipo de dado.

4.6 Comparação de *WebServices REST* com *WebServices SOAP*

Com a popularização de *WebServices REST* um número crescente de desenvolvedores estão buscando as vantagens da utilização de estilo REST, como por exemplo o auto nível de abstração para as aplicações em rede (MURACEVIC; KURTAGIC, 2009).

De acordo com (MURACEVIC; KURTAGIC, 2009) a Amazon possui alguns *WebServices* e todos têm uma versão SOAP e REST, 85% do uso dos seus *WebServices* é feito através da versão REST.

A utilização de *WebServices REST* tem algumas vantagens em relação a *WebServices SOAP*, essas vantagens estão descritas abaixo (NEWMARCH, 2005):

- O SOAP cria um novo protocolo de comunicação, normalmente sobre o HTTP, isso provoca um aumento na complexidade do serviço o que também aumenta a possibilidade de erros.
- Semanticamente o uso dos métodos HTTP GET e POST já dizem o propósito da requisição que é feita para o servidor, uma requisição GET não deve ser encarada como algo que provocará mudanças no servidor por exemplo. Em SOAP as requisições são feitas usando o método POST, isso pode ofuscar a compreensão do que a requisição faz, pois a mensagem SOAP pode apenas

solicitar uma informação por exemplo. Saber o que uma requisição pretende fazer é uma ferramenta para ajudar em sistemas de *firewall*³.

(NEWMARCH, 2005) fez a comparação de duas implementações de *WebServices* para UPnP (*Universal Plug and Play*), uma REST e outra em SOAP, as duas implementações foram escritas em Java e executadas em ambientes Linux e foram obtidos os seguintes resultados:

Tempo de execução Foram feitas 1000 requisições ao *WebServices* as que o consumidor do serviço estava em execução no mesmo sistema que o provedor do serviço são as *standalone* e as que os consumidores eram outros sistemas são as *network*, A Tabela 4.1 sintetiza os resultados.

Tipo	SOAP/Xerces	Soap/KXML	REST
standalone	196	137	67
network	165	114	64

Tabela 4.1: Tempo (em segundos) de 1000 requisições (NEWMARCH, 2005).

Tamanho da resposta Os testes determinaram o volume necessário de informação para enviar e receber dados do *WebServices*. A Tabela 4.2 mostra a quantidade de *bytes* necessários para realizar a comunicação com os provedores do serviço e detalha o acréscimo dado pelo empacotamento da requisição para ser transportada pelo protocolo de comunicação usado em cada uma das implementações.

Momento	SOAP	REST
Requisição/Dados	350	4
Requisição/Pacote	535	84
Resposta/Dados	365	11
Resposta/Pacote	526	171

Tabela 4.2: Tamanho em bytes de uma requisição (NEWMARCH, 2005).

³Dispositivo usado para controlar e impedir o tráfego nocivo entre redes.

Capítulo 5

Metodologia

5.1 Balanço Hídrico Climatológico

O Balanço Hídrico Climatológico Local foi elaborado pelo método de Thornthwaite & Mather (1955) (DANTAS; CASTRO NETO, 2003). Para o cálculo do balanço hídrico climatológico o usuário do sistema deverá fornecer os seguintes dados.

- Coordenadas geográficas do local, sendo as coordenadas a latitude e longitude expressas em graus e décimos. No sistema são aceitas apenas coordenadas que estão contidas no estado de Minas Gerais.
- Altitude em metros do local em relação ao nível do mar.

5.1.1 Temperatura

Para o cálculo do Balanço Hídrico Climatológico são necessários dados sobre as Temperaturas Médias Normais para cada mês do ano (DANTAS; CASTRO NETO, 2003). Para a determinação da temperatura média normal, para locais situados no estado de Minas Gerais, será usado o método apresentado por (SEDIYAMA; MELO JR, 1998), onde são necessários apenas informações sobre a posição geográfica do local e sua altitude.

5.1.2 Precipitação

Para o cálculo do Balanço Hídrico Climatológico também é necessário a informação das Precipitações Totais Normais para o local de interesse (DANTAS; CASTRO NETO, 2003). Para se obter essa informação para qualquer local em Minas Gerais foi criado um Banco de Dados Espacial onde está armazenado um conjunto de 12 matrizes, uma para cada mês do ano, obtidas pela interpolação espacial dos dados de precipitação total mensal das estações climatológicas do estado de Minas Gerais e daquelas que são vizinhas ao estado de Minas Gerais e estão situadas em outros estados da Federação.

Para o desenvolvimento dessa Base de Dados foram usados os seguintes softwares:

Banco de Dados: Foi usado o PostgreSQL 8.3 (POSTGRESQL, 2008) com a extensão espacial PostGIS 1.3 (REFRACTIONS, 2008).

Geração da Matriz: Para a criação da Matriz com a estimativa da Precipitação Média para Minas Gerais foi usado o Surfer 6.04 (Golder Software, 1997).

Importação dos dados para o Banco de Dados: Foi criado um conjunto de objetos na linguagem de *script* PHP (PHP, 2009) responsáveis por transferir dos dados gerados no Surfer e armazená-los em Banco de Dados.

5.1.2.1 Construção das matrizes

Para a construção das matrizes de Precipitação Total Normal Mensal foram utilizados dados das Normais Climatológicas (1961 - 1990) (BRASIL, 1992) para as estações climatológicas que disponibilizaram estas informações para o estado de Minas Gerais e também as estações adjacentes ao estado de Minas Gerais. A Figura 5.1 mostra o posicionamento de todas as estações utilizadas, das quais estão instaladas no seguintes municípios de Minas Gerais (BRASIL, 1992):

Aimorés, Araçuaí, Araxá, Bambuí, Barbacena, Belo Horizonte, Caparaó, Capinópolis, Caratinga, Conceição do Mato Dentro, Diamantina, Espinosa, Governador Valadares, Ibirité, Itamarandiba, João Pinheiro, Juiz de Fora, Lavras, Machado, Monte Azul, Montes Claros, Paracatu, Patos de Minas, Pedra Azul, Pompeu, São Lourenço, Sete Lagoas, Teófilo Otoni, Uberaba e Viçosa.

As estações adjacentes a Minas Gerais estão nos seguintes municípios:

Campos do Jordão-SP, Franca-SP, São Carlos-SP, São Simão-SP, Itaperuna-RJ, Cachoeiro do Itapemirim-ES, Carinhanha-BA, Guaratinga-BA e Brasília-DF.

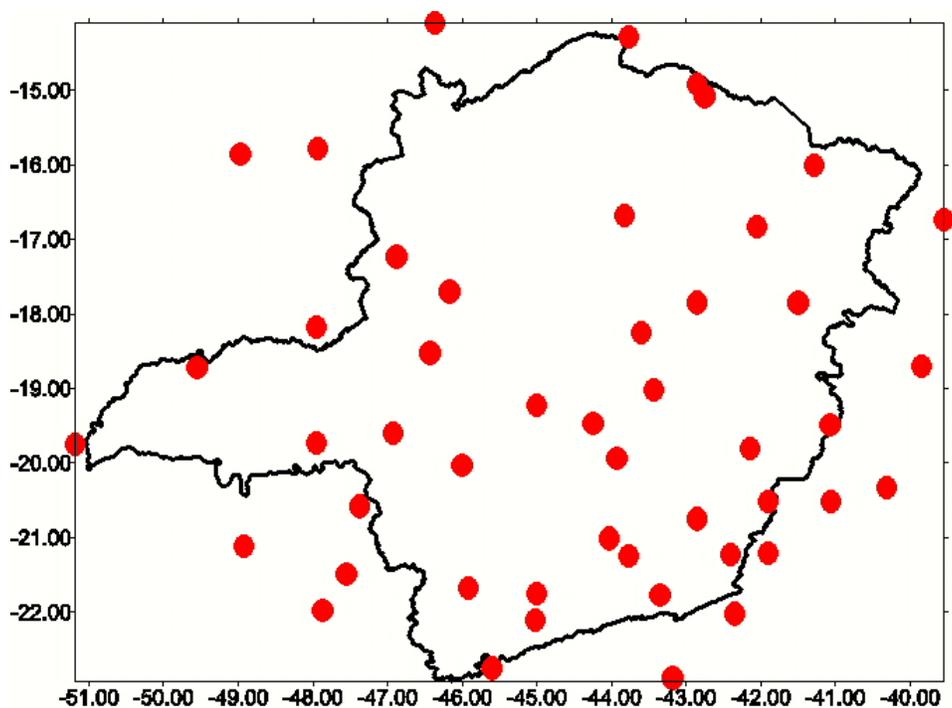


Figura 5.1: Posicionamento das estações climatológicas usadas

Com a informação da Precipitação Total Normal Mensal para cada uma dessas estações junto com a sua localização geográfica, foi criada um conjunto de pontos em um espaço tridimensional onde as coordenadas geográficas representam as coordenadas X,Y e a precipitação é interpretada como o valor do ponto em Z. A Figura 5.2 mostra as estações com cores diferentes de acordo com a precipitação registrada.

Esse conjunto de dados foi processado no Surfer 6.04, com o objetivo de construir 12 matrizes onde para um par ordenado (x,y) fosse possível obter a informação da Precipitação Total Normal Mensal para cada mês do ano. Para a construção de cada matriz foi realizada uma interpolação dos dados já existentes

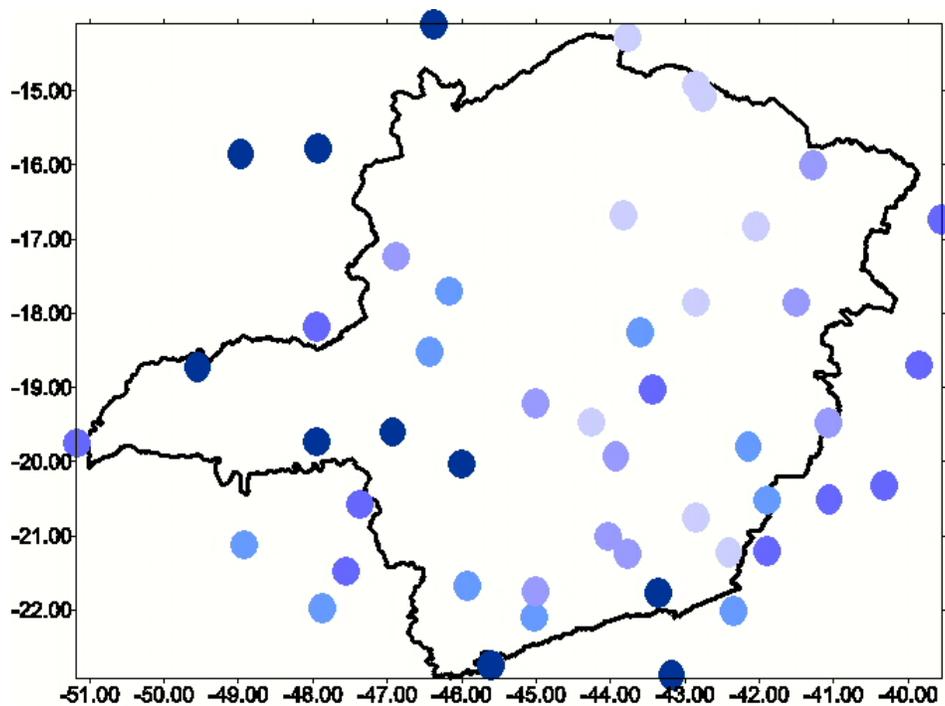


Figura 5.2: Precipitação Normal Total (1961-1990) para o mês de abril.

usando o método da Krigagem. Esse método foi escolhida pois como demonstrado por (MELLO *et al.*, 2003) “A krigagem tem capacidade de produzir melhores estimativas em termos de interpolação, porque está embasada em duas premissas: não-tendenciosidade do estimador e variância mínima das estimativas”. A Figura 5.3 dá um exemplo da matriz obtida para o mês de abril. Para a construção dessa grade foi determinado que cada ponto deveria estar distante um do outro de aproximadamente 0,03 graus em X e Y.

5.1.2.2 Importação das Matrizes para o Banco de Dados

Para a importação dos dados para o Banco de Dados foi criado um conjunto de classes escritas em PHP, com o objetivo de processar os arquivos gerados pelo Surfer, arquivos GRID (Golder Software, 1997), e armazenar os seus conteúdos

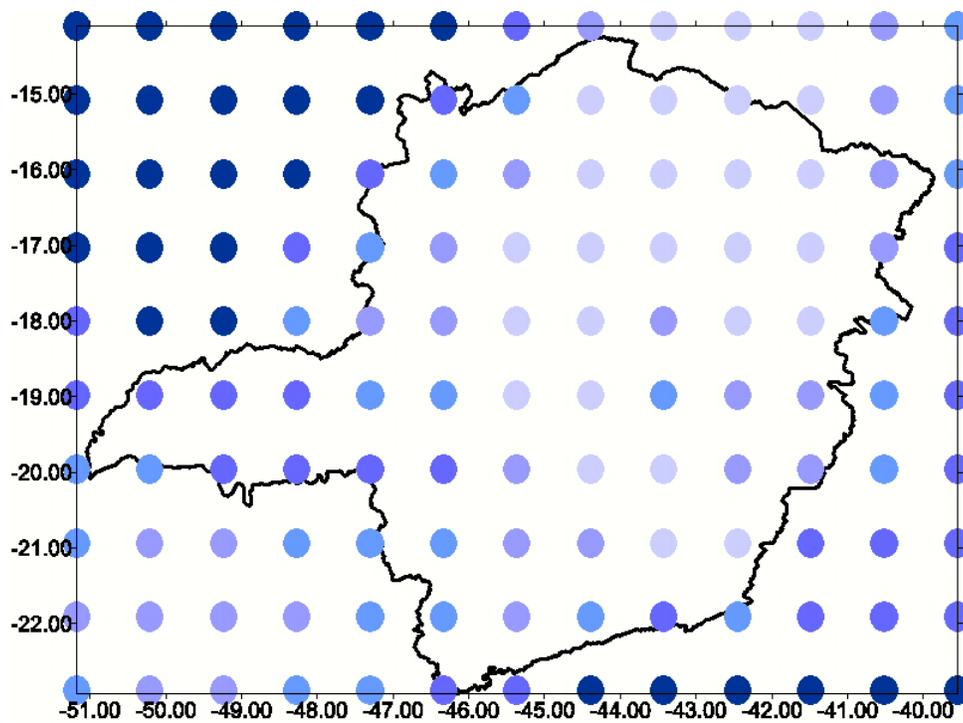


Figura 5.3: Exemplo da matriz obtida para o mês de abril

de forma que pudessem ser utilizados dentro do Banco de Dados de forma simplificada.

Para obter a informação da precipitação normal total mensal para cada ponto em Minas Gerais é necessária a representação de uma superfície plana e não de um conjunto de pontos. Para a construção dessa superfície a técnica utilizada foi a cada 4 pontos da matriz gerada pelo Surfer construir um quadrilátero e o valor de precipitação associado a esse quadrilátero ser a média da precipitação para os 4 pontos utilizados. A Figura 5.4 mostra o município de Lavras-MG ao centro e a cobertura dos quadriláteros que formam a superfície para o mês de abril.

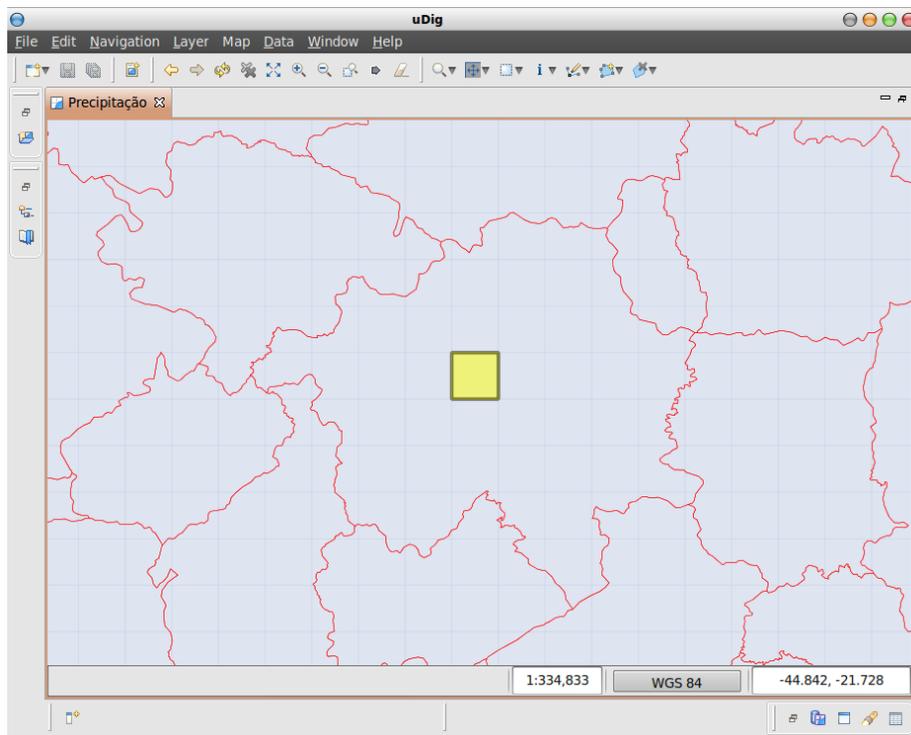


Figura 5.4: Lavras-MG e a superfície de precipitação total normal para o mês de abril

5.2 *WebService*

Para o desenvolvimento do provedor do *WebService REST* foi utilizada a linguagem Java (Sun Microsystems, 2005), usando como suporte a implementação RESTEasy (BURKE, 2009) da especificação JAX-RS (Sun Microsystems, 2008), essa especificação define um conjunto de regras e estilos para o desenvolvimento de *WebServices REST* na linguagem Java. A arquitetura do *WebService* é mostrada na Figura 5.5.

Para o uso do *WebService* foi definido que o consumidor deve invocar o *resource WaterBalance* através do método HTTP **GET**. Os parâmetros de entrada são listados a seguir:

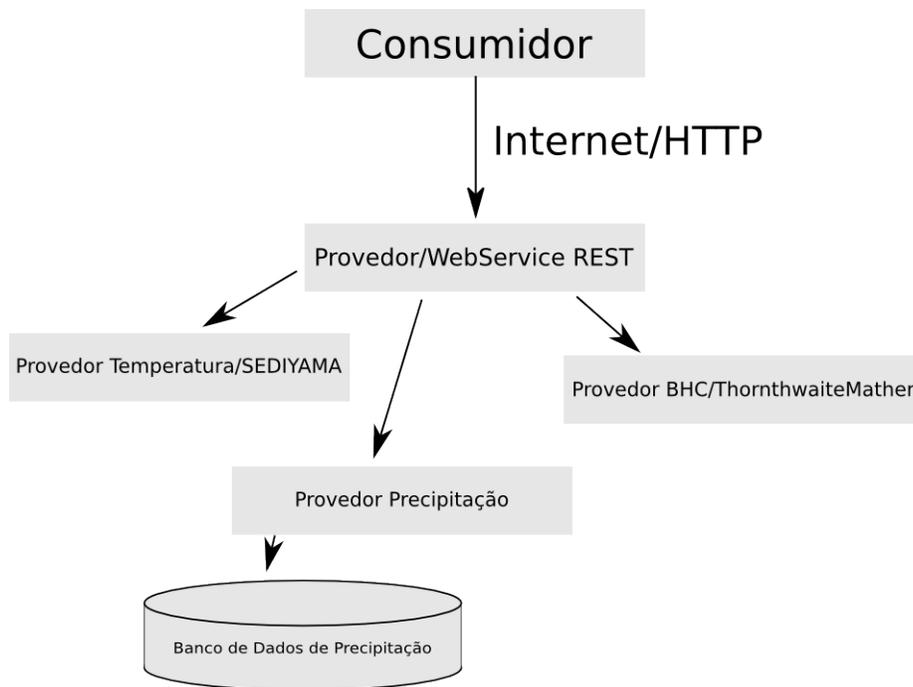


Figura 5.5: Arquitetura do *WebService REST*

- x** Valor da longitude do ponto de interesse, expresso em graus e décimos.
- y** Valor da latitude do ponto de interesse, expresso em graus e décimos.
- z** Valor da altitude do ponto de interesse, expresso em metros.

Todos os parâmetros são recebidos como *MatrixParams* (BURKE, 2009), a Figura 5.6 mostra um exemplo de como deve ser a URL que identifica o *resource* juntamente com os parâmetros de entrada para o cálculo do Balanço Hídrico Climatológico.

`http://oleo.ufla.br/WaterBalance;x=-45.01;y=-21.18;z=918`

Figura 5.6: Exemplo de uso do *WebService* para proximidade de Lavras-MG

5.3 WebGIS

Como o usuário do sistema precisa de algum nível de interação com dados espaciais, para facilitar a visualização dos locais onde se deseja realizar os cálculos para o balanço hídrico, a construção de um *WebGIS* foi a solução adotada.

Para que novas funcionalidades possam ser implementadas ao longo da vida do *WebGIS*, foi proposta a construção de um *framework*¹ para desenvolvimento de interfaces com o usuário para aplicações *WebGIS*.

O *framework* criado recebeu o nome de **Oziris**, esse *framework* é totalmente escrito em JavaScript e é independente de tecnologias usadas em servidores, ou seja, o *framework* foi desenvolvido com as funcionalidades básicas para que os sistemas desenvolvidos sobre ele pudessem realizar e/ou depender de qualquer tecnologia usada em servidores *Web*. O *framework* foi projetado para ser executado em *WebBrowsers* que seguem as especificações W3C.

5.3.1 Arquitetura do Oziris

O Oziris é baseado no *framework* OpenLayers 2.8 (OPENLAYERS, 2009), o OpenLayers é usado como suporte para as funcionalidades características de um *WebGIS*. Também é usado o *framework* Mootools 1.2.3 (PROIETTI, 2009) para prover as funcionalidades comuns em aplicações *Web 2.0*.

A arquitetura do *framework* é mostrada nas Figuras 5.7 e 5.8, nas figuras os retângulos em cinza indicam as partes do *framework* que devem ser implementadas para a utilização em um novo sistema.

O núcleo do Oziris é representado pelo conjunto de componentes responsáveis por prover os serviços mais essenciais para a execução de uma aplicação *WebGIS*. Abaixo está uma breve descrição desses componentes:

RequestManager Responsável por fazer toda a comunicação AJAX do *WebGIS* com os Servidores *Web*.

PluginManager As funcionalidades do sistema, como interação do usuário através de mapas e cálculo do balanço hídrico são representados dentro do Oziris

¹Arcabouço de soluções para conjunto de problemas similares

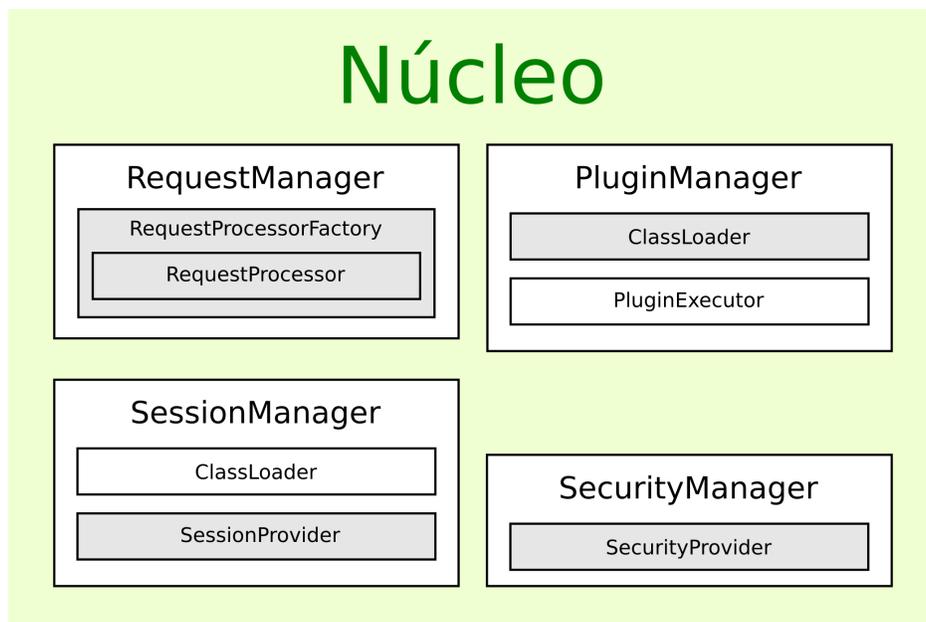


Figura 5.7: Arquitetura do Oziris: Núcleo

como *plugins*. O *PluginManager* é responsável por carregar e gerenciar o ciclo de vida dos *plugins*.

SecurityManager Responsável por fazer os controles de autenticação e acesso do usuário ao *WebGIS*.

SessionManager É a parte do sistema responsável por determinar quais *plugins* devem ser carregados e outros dados do usuário que usa o *WebGIS*.

O serviços são os componentes que os *plugins* mais fazem uso, abaixo são descritos brevemente esses componentes:

MapManager Responsável por controlar os mapas que o usuário usa no *WebGIS*. Esse componente é responsável por gerenciar o ciclo de vida dos mapas.

FIOSystem Componente responsável por simular um ambiente próximo ao sistema de arquivos usados em Sistemas Operacionais em Desktop.

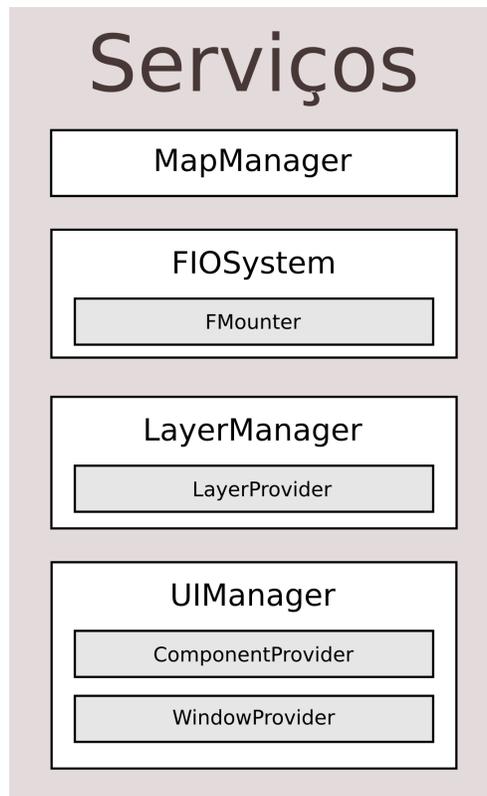


Figura 5.8: Arquitetura do Oziris: Serviços

LayerManager Um mapa é composto de um conjunto de *layers*, essas *layers* podem ser entendidas como imagens de objetos dispostos espacialmente sobre a superfície terrestre. É responsabilidade do *LayerManager* criar e controlar o ciclo de vida de uma *layer* que o usuário visualiza dentro de um mapa.

UIManager Para isolar o *framework* e os *plugins* das complicações das diversas implementações de componentes de interface com o usuários em JavaScript, é usada o componente *UIManager*, assim todos os *plugins* desenvolvidos sobre o *framework* têm um único meio para criação de componentes de interface com o usuário. Essa característica além de simplificar a criação de novas funcionalidades provê uma padronização da interface com o usuário.

5.3.2 Uso do *Framework*

Para a criação do sistema *Web* foi utilizado o *framework* proposto como base. Dentro do *framework* foram criados dois *plugins*, um responsável por localizar o ponto de interesse para o cálculo do Balanço Hídrico Climatológico e outro responsável por fazer a comunicação com o *WebService REST* proposto, e com os dados providos pelo *WebService* gerar uma interface com o usuário para que ele possa ver o resultado do cálculo do Balanço Hídrico Climatológico de forma amigável.

Capítulo 6

Resultados

O desenvolvimento de um *WebService* para o cálculo do balanço hídrico em Minas Gerais cria uma nova ferramenta para que os produtores e técnicos agrícolas possam planejar e executar suas atividades com mais eficiência.

6.1 Estudo climatológico para Minas Gerais

Com a confecção do conjunto de 12 matrizes para construir o banco de dados das precipitações normais totais mensais se obteve a estimativa da precipitação para todo o estado de Minas Gerais para todos os meses do ano. A Figura 6.1 mostra o mapa gerado com esses dados para o mês de abril e a Figura 6.2 mostra o mapa gerado para o mês de dezembro.

6.2 Desenvolvimento de Aplicações *WebGIS*

Com o desenvolvimento do *framework* Oziris a construção do serviço *web* para o usuário realizar o cálculo do balanço hídrico se tornou fácil e rápido, pois todo o ambiente provido pelo *framework* disponibilizou todos os recursos necessários para a construção dos mesmos *plugins*. A Figura 6.3 mostra o cálculo do balanço

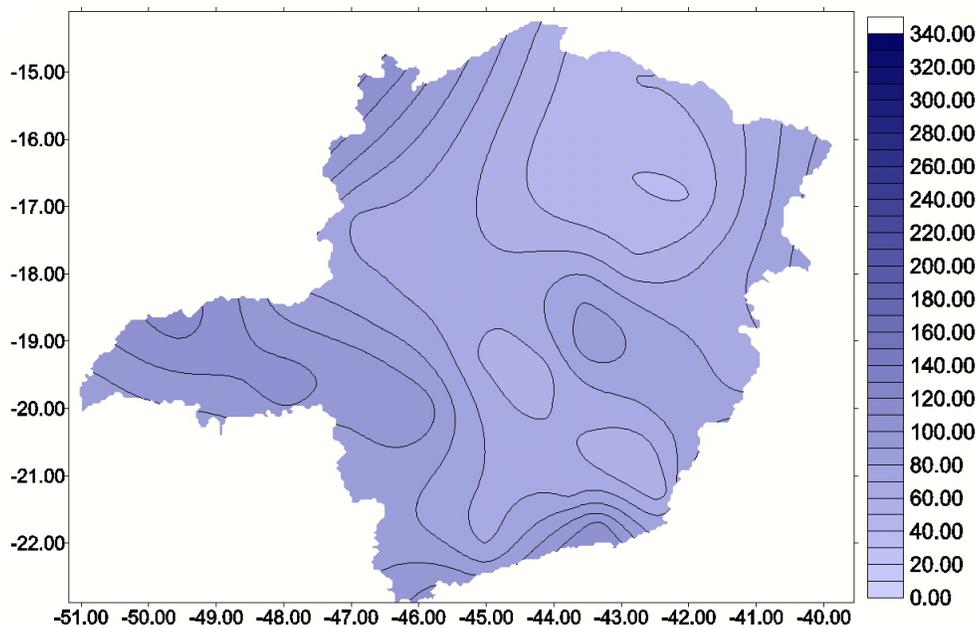


Figura 6.1: Precipitação normal total para o mês de abril

hídrico para um ponto dentro da Universidade Federal de Lavras utilizando o sistema proposto.

6.3 Testes com o Sistema

Foram realizados testes de usabilidade do sistema com estudantes de agronomia e engenharia agrícola da Universidade Federal de Lavras. Esse público foi escolhido pois é um público com o interesse na obtenção de dados climáticos para o planejamento agrícola. Os testes foram realizados com 3 estudantes que atribuíram notas de 0 a 5 (0 péssimo; 5 ótimo), sendo o resultado da avaliação do uso do sistema apresentado na Tabela 6.1.

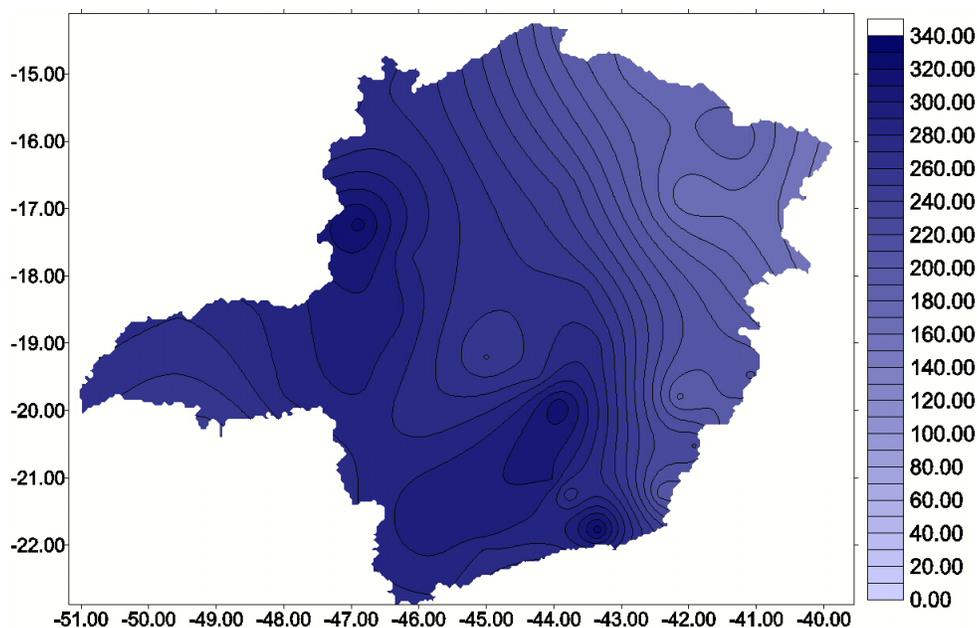


Figura 6.2: Precipitação normal total para o mês de dezembro

Interface	Localização	Entendimento	Aplicabilidade
4	3	5	5
5	4	5	4
3	5	5	5

Tabela 6.1: Resultados dos testes de usabilidade do sistema proposto.

6.4 Aplicação do Sistema

A facilidade de uso do sistema e sua aplicabilidade no aumento da possibilidade de sucesso de investimento em produção agrícola poderá se traduzir em aumento da produtividade no estado de Minas Gerais. Este pressuposto será verificado na segunda etapa do sistema e quando for disponibilizado ao público, quando será implementado também um processo de acompanhamento e avaliação do resultado da aplicação do sistema.

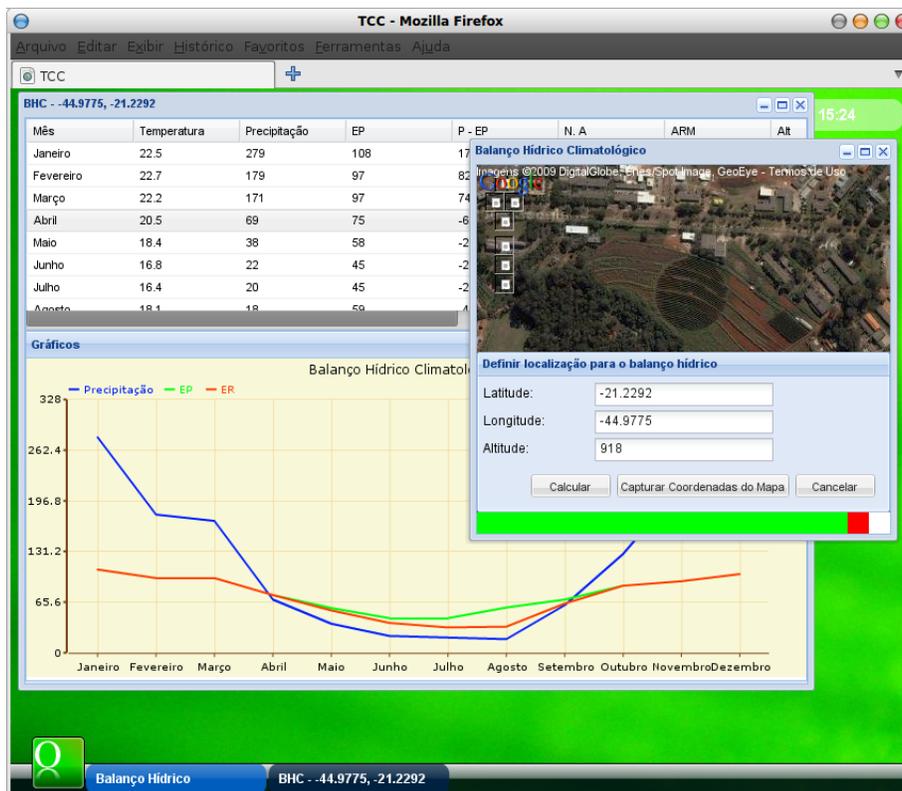


Figura 6.3: Tela do sistema em uso.

Capítulo 7

Conclusões

Com o desenvolvimento do *WebService* e do *WebGIS* propostos foi possível determinar o cálculo do balanço hídrico para diversas localidades no estado de Minas Gerais.

Os resultados dos testes feitos com os estudantes de agronomia e engenharia agrícola mostraram que a utilização do sistema é fácil e prática, assim o uso dele por qualquer usuário pode ser feito de forma satisfatória em qualquer localidade.

O sistema proposto provê para os agricultores do estado de Minas Gerais mais uma fonte de informação e serviços que poderão ser usados no processo de tomada de decisão no planejamento das atividades agrícolas. Isso implica que o sistema poderá contribuir para o aumento do sucesso das atividades agrícolas desenvolvidas no estado de Minas Gerais.

Referências Bibliográficas

ADOBE. *Adobe Flex*. 2009. Acessado em 5 de novembro de 2009. Disponível em: <<http://www.adobe.com/products/flex>>.

ANP, A. *Biocombustíveis » Biodiesel*. 2008. Acessado em 3 de Novembro de 2009. Disponível em: <<http://www.anp.gov.br/biocombustiveis/biodiesel.asp>>.

ANP, A. N. do Petróleo Gás Natural e B. *Boletim Mensal de Biodiesel - SRP*. Setembro 2009. Acessado em 3 de Novembro de 2009. Disponível em: <http://www.anp.gov.br/doc/biodiesel/Boletim_Mensal_de_Biodiesel_setembro09.pdf>.

ARAÚJO, M. A. *Web services na informação geográfica*. Dissertação (Mestrado) — Universidade do Minho, Braga, 2005.

BRASIL. *Normais Climatológicas (1961 - 1990)*. [S.l.]: Ministério da Agricultura e Reforma Agrária, 1992.

BURKE, B. *RESTEasy - RESTful Web Services for Java*. 2009. Acessado em 15 de Novembro de 2009. Disponível em: <http://www.jboss.org/file-access/default/members/reteasy/freezone/docs/1.2.GA/userguide/pdf/RESTEasy_Reference_Guide.pdf>.

CAPUTI, E. *Balanço hídrico no estado de São Paulo: Geoinformação na Internet*. Dissertação (Mestrado) — Instituto Agrônomo de Campinas, Campinas, SP, Julho 2006.

CURBERA, F.; DUFTLER, M.; KHALAF, R.; NAGY, W.; MUKHI, N.; WEERAWARANA, S. Unraveling the web services web: An introduction to soap, wsdl, and uddi. *IEEE Internet Computing*, IEEE Computer Society, v. 6, n. 2, p. 86–93, 2002. ISSN 1089-7801.

DALI, T.; JIXIAN, H. Structure and technology on application-webgis. *Info-tech and Info-net, 2001. Proceedings. ICII 2001 - Beijing. 2001 International Conferences on*, v. 1, p. 254–260 vol.1, 2001.

DANTAS, A. A. A.; CARVALHO, L. G.; FERREIRA, E. Classificação e tendências climáticas em lavras, mg. *Ciência e Agrotecnologia*, scielo, v. 31, p. 1862–1866, 2007. ISSN 1413-7054.

DANTAS, A. A. A.; CASTRO NETO, P. Balanço hídrico climatológico e classificação climática. Nota de Aula. 2003.

ENDREI, M.; ANG, J.; ARSANJANI, A.; CHUA, S.; COMTE, P.; KROGDAHL, P.; LUO, M.; NEWLING, T. *Patterns: service-oriented architecture and web services*. [S.l.]: IBM Corp., International Technical Support Organization, 2004.

ESRI. *ArcGIS® Desktop - Tools for Authoring, Editing, and Analyzing Geographic Information*. 2007. Acessado em 4 de novembro de 2009. Disponível em: <<http://www.esri.com/library/brochures/pdfs/arcgis-desktop.pdf>>.

FIELDING, R.; GETTYS, J.; MOGUL, J.; FRYSTYK, H.; MASINTER, L.; LEACH, P.; BERNERS-LEE, T. Hypertext Transfer Protocol-HTTP/1.1. 1999.

FIELDING, R. T. *Architectural Styles and the Design of Network-based Software Architectures*. Tese (Doutorado) — UNIVERSITY OF CALIFORNIA, IRVINE, 2000.

GARRETT, J. *et al.* Ajax: A new approach to web applications. *Adaptive Path*, 2005.

Golder Software. *Surfer - 6.04*. [S.l.], 1997.

GRASS. *GRASS - Geographic Resources Analysis Support System*. 2007. Acessado em 5 de novembro de 2009. Disponível em: <<http://grass.osgeo.org>>.

KOPECKY, J.; GOMADAM, K.; VITVAR, T. hrests: An html microformat for describing restful web services. *Web Intelligence and Intelligent Agent Technology, 2008. WI-IAT '08. IEEE/WIC/ACM International Conference on*, v. 1, p. 619–625, 2008.

MEIRELLES, J. C.; MOURA, M. Web 2.0: novos paradigmas projetuais e informacionais. *InfoDesign Revista Brasileira de Design da Informação*, p. 12–18, 2007.

MELLO, C.; LIMA, J.; SILVA, A.; MELLO, J.; OLIVEIRA, M. Krigagem e inverso do quadrado da distância para interpolação dos parâmetros da equação de chuvas intensas. *Revista Brasileira de Ciência do Solo*, SciELO Brasil, v. 27, n. 5, p. 925–933, 2003.

MURACEVIC, D.; KURTAGIC, H. Geospatial soa using restful web services. *Information Technology Interfaces, 2009. ITI '09. Proceedings of the ITI 2009 31st International Conference on*, p. 199–204, 2009. ISSN 1330-1012.

NAGAOKA, A. K.; FRANÇA, F. A.; CASTRO NETO, P.; FRAGA, A. C. Levantamento das condições climatológicas do estado de santa catarina para avaliar o potencial de produção de plantas oleaginosas. *5o Congresso Biodiesel de Plantas Oleaginosas, Óleos, Gorduras e Biodiesel*, Julho 2008.

NEWMARCH, J. A restful approach: clean upnp without soap. In: *Proceedings of the 2nd IEEE Consumer Communications and Networking Conference (CCNC)*. [S.l.: s.n.], 2005. p. 134–138.

NGOLO, M. Arquitectura orientada a serviços rest para laboratórios remotos. FCT-UNL, 2009.

NUNES, S.; DAVID, G. Uma arquitectura web para serviços web. 2005.

OASIS. *UDDI Version 2.03 Data Structure Reference*. 2002. Disponível em: <<http://uddi.org/pubs/DataStructure-V2.03-Published-20020719.htm>>.

OPENLAYERS. *OpenLayers: Free Maps for the Web*. 2009. Acessado em 16 de Novembro de 2009. Disponível em: <<http://www.openlayers.org/>>.

PAUTASSO, C. Soap vs. rest bringing the web back into web services. *IBM Zurich Research Lab*, Citeseer, 2007.

PAUTASSO, C.; ZIMMERMANN, O.; LEYMANN, F. Restful web services vs. big web services: making the right architectural decision. In: *WWW '08: Proceeding of the 17th international conference on World Wide Web*. New York, NY, USA: ACM, 2008. p. 805–814. ISBN 978-1-60558-085-2.

PEREIRA, A. R. Simplificado o balanço hídrico de thornthwaite-mather. *Bragantia*, sciELO, v. 64, p. 311–313, 2005. ISSN 0006-8705.

PHP. *The PHP Group - PHP Manual*. [S.l.], 2009. Disponível em: <<http://www.php.net/manual/>>.

POSTGRESQL. *PostgreSQL Global Development Group - PostgreSQL 8.3.8 Documentation*. [S.l.], 2008. Disponível em: <<http://www.postgresql.org/docs/8.3/interactive/index.html>>.

PRIMO, A. O aspecto relacional das interações na web 2.0. *Revista da Associação Nacional dos Programas de Pós-Graduação em Comunicação*, Agosto 2007.

PROIETTI, V. *Mootools - A compact javascript framework*. 2009. Acessado em 16 de Novembro de 2009. Disponível em: <<http://www.mootools.net/>>.

REFRACTIONS. *Refractions Research - PostGIS 1.3.6 Manual*. [S.l.], 2008. Disponível em: <<http://postgis.refractions.net/documentation/manual-1.3/>>.

REZENDE, D. *Engenharia de software e sistemas de informação*. [S.l.]: Brasport, 2006.

RIBEIRO, J. *Adição de 5% de biodiesel no diesel passa a ser obrigatória em 2010*. Outubro 2009. Acessado em 3 de Novembro de 2009. Disponível em: <<http://g1.globo.com/>>.

SEDIYAMA, G. C.; MELO JR, J. C. Modelos para estimativa das temperaturas normais mensais médias, máximas, mínimas e anual no estado de minas gerais. *Engenharia na Agricultura*, v. 6, n. 1, p. 57–61, 1998.

SHAH, D. *A Complete Guide to Internet and Web Programming*. [S.l.]: Wiley India Pvt. Ltd., 2009.

Sun Microsystems. *The Java Language Specification, Third Edition*. [S.l.]: Sun Microsystems, 2005.

Sun Microsystems. *JAX-RS: Java API for RESTful Web Services*. 2008. Acessado em 15 de Novembro de 2009. Disponível em: <<https://jsr311.dev.java.net/>>.

TAN, X.; ZHOU, M.; ZUO, X.; CUI, Y. Integration webgis with ajax and xml based on google maps. In: . [S.l.: s.n.], 2008. p. 376–379.

THORNTHWAITE, C. W. An approach toward a rational classification of climate. *Geography Review*, v. 38, p. 55–94, 1948.

- THORNTHWAITE, C. W.; MATHER, J. R. The water balance. *Publication in Climatology - Drexel Institute of Technology. Laboratory of Climatology*, v. 8, 1955.
- TOMASELLA, J.; ROSSATO, L. *Balanço Hídrico*. [S.l.]: Instituto Nacional de Pesquisas Espaciais, 2005. (Tópicos em Meio Ambiente e Ciências Atmosféricas).
- VINOSKI, S. Restful web services development checklist. *Internet Computing, IEEE*, v. 12, n. 6, p. 96–95, 2008. ISSN 1089-7801.
- W3C. *Simple Object Access Protocol (SOAP) 1.1*. 2000. Acessado em 10 de Novembro de 2009. Disponível em: <<http://www.w3.org/TR/2000/NOTE-SOAP-20000508/>>.
- W3C. *Web Services Description Language (WSDL) 1.1*. 2001. Acessado em 10 de Novembro de 2009. Disponível em: <<http://www.w3.org/TR/wsdl/>>.
- W3C. *Extensible Markup Language (XML) 1.0 (Fifth Edition)*. 2008. Acessado em 10 de Novembro de 2009. Disponível em: <<http://www.w3.org/TR/REC-xml/>>.
- WANG, L.; HU, D.; WANG, Z.; SONG, H. Research and realization of ria webgis based on flex. *Jisuanji Yingyong/ Journal of Computer Applications*, Chengdu Institute of Computer Applications, P. O. Box 237, Chengdu, 610041, China., v. 28, n. 12, p. 3257–3260, 2008.
- WORBOYS, M.; DUCKHAM, M. *GIS: A computing perspective*. [S.l.]: CRC, 2004.
- YINGWEI, L.; XIAOLIN, W.; ZHUOQUN, X. Design of a framework for multi-user/application oriented webgis services. *Computer Networks and Mobile Computing, 2001. Proceedings. 2001 International Conference on*, p. 151–156, 2001.