

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DEPARTAMENTO ACADÊMICO DE INFORMÁTICA
CURSO DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS**

JOSÉ MARIA FARIA

**FERRAMENTA WEB PARA PUBLICAÇÃO DE MAPAS PARA
AGRICULTURA DE PRECISÃO**

TRABALHO DE CONCLUSÃO DE CURSO

**PONTA GROSSA
2015**

JOSÉ MARIA FARIA

**FERRAMENTA WEB PARA PUBLICAÇÃO DE MAPAS PARA
AGRICULTURA DE PRECISÃO**

Trabalho de Conclusão de Curso apresentado como requisito parcial à obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas, do Departamento Acadêmico de Informática, da Universidade Tecnológica Federal do Paraná.

Orientador: Prof. M.Sc. Luiz Rafael Schmitke

**PONTA GROSSA
2015**



TERMO DE APROVAÇÃO

FERRAMENTA WEB PARA PUBLICAÇÃO DE MAPAS PARA AGRICULTURA DE PRECISÃO

por

JOSE MARIA FARIA

Este **Trabalho de Diplomação** foi apresentado em **6 de novembro de 2015** como requisito parcial para a obtenção do título de **Tecnólogo em Análise e Desenvolvimento de Sistemas**. Os candidatos foram arguidos pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho aprovado.

Prof. M.Sc. Luiz Rafael Schmitke
Prof. Orientador

Prof. M.Sc. Mathias Talevi Betim
Membro titular

Prof. Dr. Richard Duarte Ribeiro
Membro titular

- O Termo de Aprovação assinado encontra-se na Coordenação do Curso -

AGRADECIMENTOS

Certamente estes parágrafos não irão atender a todas as pessoas que fizeram parte dessa importante fase de minha vida. Portanto, desde já peço desculpas àquelas que não estão presentes entre essas palavras, mas elas podem estar certas que fazem parte do meu pensamento e de minha gratidão.

Agradeço ao meu orientador pela sabedoria com que me guiou nesta trajetória.

Aos meus colegas de sala.

A Secretaria do Curso, pela cooperação.

Gostaria de deixar registrado também, o meu reconhecimento à minha família, pois acredito que sem o apoio deles seria muito difícil vencer esse desafio.

Enfim, a todos os que por algum motivo contribuíram para a realização desta pesquisa.

RESUMO

FARIA, José M. **Ferramenta Web para Publicação de Mapas para Agricultura de Precisão**. 2015. 47f. Trabalho de Conclusão de Curso (Tecnologia em Análise e Desenvolvimento de Sistemas) - Universidade Tecnológica Federal do Paraná. Ponta Grossa, 2015.

Esse trabalho destinou-se ao estudo de tecnologias que possam ser aplicadas no desenvolvimento de aplicações para a Agricultura de Precisão. O Estudo foi elaborado com o objetivo de oferecer um repositório multiplataforma e interoperável de dados e informações. Para permitir que o conteúdo seja acessado por qualquer conjunto de *software* e *hardware* operados pelos usuários, foram especificados o HTML – *HyperText Markup Language* e o HTTP – *HyperText Transport Protocol*, como mecanismos para transmitir arquivos textuais entre diferentes plataformas e sistemas. Camadas são formas de podermos organizar dados em um SIG, essas camadas são definidas para cada tema proposto, e para esses temas teremos as respectivas camadas necessárias para trabalhar com dados cartográficos operacionais conectados na internet (BERNERS-LEE, 1990), utilizando o *Openlayers* (OL), uma rica API para contruir aplicações geográficas. Essa biblioteca possui vários recursos para todas as necessidades para SIG WEB ou WEB GIS. Essa biblioteca *JavaScript* permite a manipulação de mapas, podendo eles serem armazenados em arquivos geos como KML, GML, *Shapefile* entre outros. O OL pode ser executado localmente ou remotamente através dos chamados *Middlewares*, como *GeoServer* ou *MapServers* e diversos outros *Gis Open Source* para publicação de dados espaciais na web.

Palavras-chave: HTML. HTTP. AGRICULTURA DE PRECISÃO. OPENLAYERS. JAVASCRIPT. GIS. MIDDLEWARES.

ABSTRACT

FARIA, José M. **Web Tool for Maps Publication for Precision Agriculture**. 2015. 47f. Trabalho de Conclusão de Curso (Tecnologia em Análise e Desenvolvimento de Sistemas) - Federal Technology University - Paraná. Ponta Grossa, 2015.

This work intended to study technologies that can be applied on the development of applications for Precision Agriculture. The study was elaborated with the objective to offer a multiplatform and interoperable repository of data and information. To allow the content may be accessed in any set of software and hardware operated by the users were specified HTML – Hypertext Markup Language and HTTP – Hypertext Transport Protocol, as ways to transmit textual files between different platforms e systems. Layers are ways we can organize data in a SIG, these layers are defined for each proposed theme, and for these themes we will have respective layers needed to work with operational cartographic data connected through the internet (BERNERS-LEE, 1990) using Open Layers (OL), a rich API to build geographic applications. This library owns many resources to all the needs of SIG WEB or WEB GIS. This JavaScript library allows manipulation of maps, been able to storage them in geo files as .KML, .GML, Shapefile among others. The OL may be executed locally or remotely through the so called Middlewares, like Geoserver or Mapservers and many others Gis Open Source for publishing spatial data in the web.

Keywords: HTML. HTTP. PRECISION AGRICULTURE. OPEN LAYERS. JAVASCRIPT. GIS. MIDDLEWARES.

LISTA DE FIGURAS

Figura 1: Tabela SGBD relacional (a) e Tabela SGBDG (b)	14
Figura 2: Camada de um SIG.....	15
Figura 3: Modelos de representação de dados em um SIG	17
Figura 4: Exemplo de uma mensagem SOAP	20
Figura 5: Exemplo de requisição HTTP.....	21
Figura 6: Exemplo de resposta HTTP	22
Figura 7: Código de exemplo Java.....	23
Figura 8: Código de exemplo <i>Javascript</i>	24
Figura 9: Exemplo de código HTML	25
Figura 10: Exemplo de código CSS	26
Figura 11: Exemplo de código <i>OpenLayers</i>	27
Figura 12: Exemplo de mapa gerado com <i>OpenLayers</i>	28
Figura 13: Tela QGIS	31
Figura 14: Tela do sistema <i>Farm Work</i>	32
Figura 15: Visualização de um mapa gerado no geopublicador.....	35
Figura 16: Portal de acesso ao geopublicador	36
Figura 17: Tela de login do geopublicador	36
Figura 18: Tela com os menus do sistema.....	37
Figura 19: Tela de busca de fazendas	38
Figura 20: Tela de busca de glebas	38
Figura 21: Imagem do mapa selecionado	39
Figura 22: Mapa de amostra	39
Figura 23: Mapa de aplicação	40
Figura 24: Mapa de aplicação com legendas.....	41
Figura 25: Mapa em PDF	41
Figura 26: Download de mapa para aplicação em equipamento.....	42

LISTA DE SIGLAS

AP	<i>Agricultura de Precisão</i>
CSS	<i>Cascading Style Sheets</i>
GML	<i>Geography Markup Language</i>
GPS	<i>Global Positioning System</i>
GWT	<i>Google Web Toolkit</i>
HTML	<i>HyperText Markup Language</i>
HTTP	<i>HyperText Transport Protocol</i>
KML	<i>Keyhole Markup Language</i>
OL	<i>OpenLayers</i>
SHP	<i>Shapefile</i>
WFS	<i>Web Feature Services</i>
WFS-T	<i>Web Feature Services Transact</i>
WMS	<i>Web Maps Service</i>
BDG	<i>Bando de Dados Geográficos</i>
NASA	<i>National Aeronautics and Space Administration</i>

LISTA DE ACRÔNIMOS

AJAX	<i>Asynchronous JavaScript and XML</i>
CRUD	<i>Create Read Update Delete</i>
GEOJSON	<i>Geo JavaScript Object Notation</i>
REST	<i>Representational State Transfer</i>
RIA	<i>Rich Internet Application</i>
SIG	Sistema de Informação Geográfica
SOA	Sistema Orientado a Serviço

SUMÁRIO

1. INTRODUÇÃO	11
1.1. OBJETIVOS	12
1.2. JUSTIFICATIVA	12
2. EMBASAMENTO TEÓRICO	13
2.1. BANCO DE DADOS GEOGRÁFICOS	13
2.2. SIG: SISTEMA DE INFORMAÇÕES GEOGRÁFICAS	14
2.2.1. Modelos de Representação de Dados em um SIG	16
2.2.2. Padronização de Dados em um SIG	17
2.3. ARQUITETURAS ORIENTADAS A SERVIÇO	19
2.3.1. SOAP: <i>Simple Object Access Protocol</i>	19
2.3.2. REST: <i>Representational State Transfer</i>	20
2.4. MIDDLEWARES	20
2.4.1. GeoServer	21
2.4.2. MapServer	21
2.5. ARQUITETURA WEB	21
2.6. LINGUAGENS DE PROGRAMAÇÃO E MARCAÇÃO	22
2.6.1. Java	22
2.6.2. Javascript	24
2.6.3. HTML: <i>HyperText Markup Language</i>	24
2.6.4. CSS: <i>Cascading Style Sheets</i>	25
2.7. FERRAMENTAS	26
2.7.1. <i>Open Layers</i>	26
2.7.2. <i>Google Web Toolkit</i>	29
3. METODOLOGIA	30
3.1. COLETA DE DADOS	30
3.2. DESENVOLVIMENTO DO SOFTWARE	30
4. SISTEMAS SIMILARES	31
4.1. QGIS	31
4.2. FARM WORK	32
5. GEOPUBLICADOR	33
5.1. ESTRUTURA	33
5.2. VISUALIZAÇÃO DOS MAPAS	34
5.3. TELAS	35
6. PESQUISA DE SATISFAÇÃO	43
7. CONCLUSÃO	44
7.1. TRABALHOS FUTUROS	45
REFERÊNCIAS	46

1. INTRODUÇÃO

A Agricultura de Precisão (AP) é um sistema de gerenciamento agrícola baseado na variação espacial e temporal da unidade produtiva e visa ao aumento de retorno econômico, à sustentabilidade e à minimização do efeito ao ambiente, hoje em dia existem várias ferramentas e tecnologias voltadas a Agricultura de Precisão, essas ferramentas permitem o uso racional dos insumos agrícolas e também a redução do impacto ambiental decorrente dessa atividade.

Seu objetivo principal é detectar, monitorar e manejar a variabilidade espacial e temporal dos sistemas de produção agropecuários buscando a otimização. Nessa atividade existe muita variabilidade espacial, esses atributos são relacionados à textura do solo, fertilidade, controle de pragas e produtividade. Em todos esses atributos existem valores diferentes nos diversos pontos da lavoura, dependendo das dimensões, relevo, material de origem, clima local, profundidade. Hoje em dia é muito comum o agrônomo efetuar aplicações no solo com taxas fixas de adubos e nutrientes, porém a agricultura de precisão levando em conta essa variabilidade espacial propõe aplicações com taxas variáveis. Essas aplicações são prescritas de acordo com a necessidade de cada subárea, dividindo-se a gleba de acordo com essa homogeneidade.

A gleba é dividida em sub-áreas e marcada com pontos numéricos, em cada ponto é coletado uma pequena amostra do solo, essa amostra é levada a um laboratório para confecção de uma análise de solo. Com o resultado dessa análise o agrônomo pode tomar decisões de o que e como aplicar.

Alguns *softwares* voltados a Agricultura de Precisão importam os dados da análise de solo, geram mapas de aplicação, mapas de cobertura, e após analisados, enviam informações para o equipamento no campo, essas informações são enviadas através de *Pen-drivers*, que após atualizadas no equipamento esses fazem comunicação com satélite e efetuam aplicações em taxas variadas no solo de acordo com as quantidades programadas.

Um dos problemas encontrados pelos agrônomos na atualidade, apesar de toda essa tecnologia, é disponibilizar esses mapas de aplicações e cobertura para os proprietários, gerentes, ou gestores das fazendas, para que possam acompanhar em tempo real a situação de suas áreas e também a quantidade de insumos que serão aplicados.

Desta forma este trabalho busca desenvolver um geopublicador em ambiente *Web*, de forma que os mapas gerados fiquem disponíveis em tempo real para todos os interessados.

1.1. OBJETIVOS

O objetivo geral deste trabalho consiste em criar um geopublicador para plataforma *Web*, de forma que esta ferramenta ajudará a disponibilizar mapas de aplicações e coberturas para os proprietários, gestores, gerentes ou quaisquer pessoas que precisem ou queiram visualizar esses mapas em tempo real, desde que tenham um acesso à Internet. Os objetivos específicos são:

- Analisar softwares similares a fim de levantar requisitos necessários para desenvolvimento do Geopublicador Web;
- Estudar conceitos relacionados a Sistemas de Informações Geográficas;
- Estudar ferramentas necessárias para o desenvolvimento do Geopublicador Web;
- Desenvolver GeoPublicador Web.

1.2. JUSTIFICATIVA

A agricultura de precisão vem ganhando mais adeptos e inovação tecnológica. A grande maioria dos softwares *open source* são desenvolvidos para *desktop*, com base de dados local, ficando assim os mapas gerados escravos a um equipamento.

Com um publicador de mapas e os arquivos *shape* na *Web*, os resultados ficam acessíveis com mais rapidez, agilizando todo o processo para os usuários.

Desta forma este trabalho busca desenvolver um geopublicador em ambiente *Web*, de forma que os mapas gerados fiquem disponíveis em tempo real para todos os interessados.

2. EMBASAMENTO TEÓRICO

Este capítulo descreve os conceitos necessários para o entendimento do trabalho proposto. A seção 2.1 apresenta informações sobre banco de dados geográficos. A seção 2.2 apresenta um breve conceito de sistema de informações geográficas (SIG). A seção 2.3 apresenta conceitos relacionados a arquitetura orientada a serviços (SOA).

A seção 2.4 apresenta conceitos relacionados a *middlewares* necessários para o funcionamento da aplicação. A seção 2.5 apresenta superficialmente um pouco da arquitetura *Web* e seu funcionamento. A seção 2.6 apresenta as linguagens de programação e marcação necessárias para o desenvolvimento do trabalho e por fim, a seção 2.7 apresenta as ferramentas utilizadas para estruturação e exibição de mapas. A seção 2.8 apresenta a metodologia utilizada para o desenvolvimento e levantamento de resultado da aplicação.

2.1. BANCO DE DADOS GEOGRÁFICOS

Segundo (DATE 2004, p.10), “Um banco de dados é uma coleção de dados persistentes, usadas pelos sistemas de aplicação de uma determinada empresa”.

Então um banco de dados é o responsável por armazenar informações de uma determinada aplicação em um local sendo esse local um repositório, fonte de dados para uma organização.

Um sistema convencional depende de um banco de dados para armazenar e disponibilizar informações, os SIG também dependem de um banco de dados, porém a extensão espacial é fundamental, essa arquitetura ficou conhecida como Banco de Dados Geográficos (BDG).

Dados geográficos são aqueles que possuem uma dimensão espacial, uma localização, diretamente ligada ao mundo geográfico real, como mostra a Figura 1, onde compara-se uma tabela de um banco de dados relacional, A, com um geográfico, B. Dados de inventários cadastrais, dados coletados de ambientes e os modelos numéricos de terreno (VINHAS, 2006). Banco de Dados Geográficos são coleções de dados georreferenciados manipulados por Sistema de Informações Geográficas (SIG).

aula.endereco	aula.endereco
rua : varchar(255)	rua : varchar(255)
cep : char(8)	cep : char(8)
bairro : varchar(255)	bairro : varchar(255)
cidade : varchar(255)	cidade : varchar(255)
	#x : float
	#y : float

Figura 1: Tabela SGBD relacional (a) e Tabela SGBDG (b)
Fonte: Autoria Própria

Um banco de dados com informações geográficas além de dados normais ainda possui campos com dados dos tipos:

- *Point*: (0 0);
- *Linestring*: (0 0,1 1,1 2);
- *Polygon*: (0 0,4 0,4 4,0 4, 00);
- *Multipoint*: (0 0, 1 2);
- *Multilinestring*: ((0 0,1 1,1 2), (2 3,3 2,5 4));
- *Multipolygon*: (((0 0,4 0,4 4,0 4,0 0), (1 1,2 1,2 2,1 2,1 1)), ((-1,-1,-1-2,-2 -2,-2 -1,-1 -1)));
- *Geometrycollection*: (*Point* (2 3), *Linestring*((2 3,3 4))).

2.2. SIG: SISTEMA DE INFORMAÇÕES GEOGRÁFICAS

Sistema de Informações Geográficas (SIG) provém ferramentas de software bastante poderosas para realizar o tratamento de dados geográficos. A principal diferença de um SIG para um sistema de informação convencional é sua capacidade de armazenar tanto os atributos descritivos como as geometrias dos diferentes tipos de dados geográficos (CASANOVA ET al., 2005).

A Figura 2 demonstra as diversas camadas e dados que um SIG pode gerenciar, cada camada possui uma ou mais ferramentas para permitir a sua manutenção.

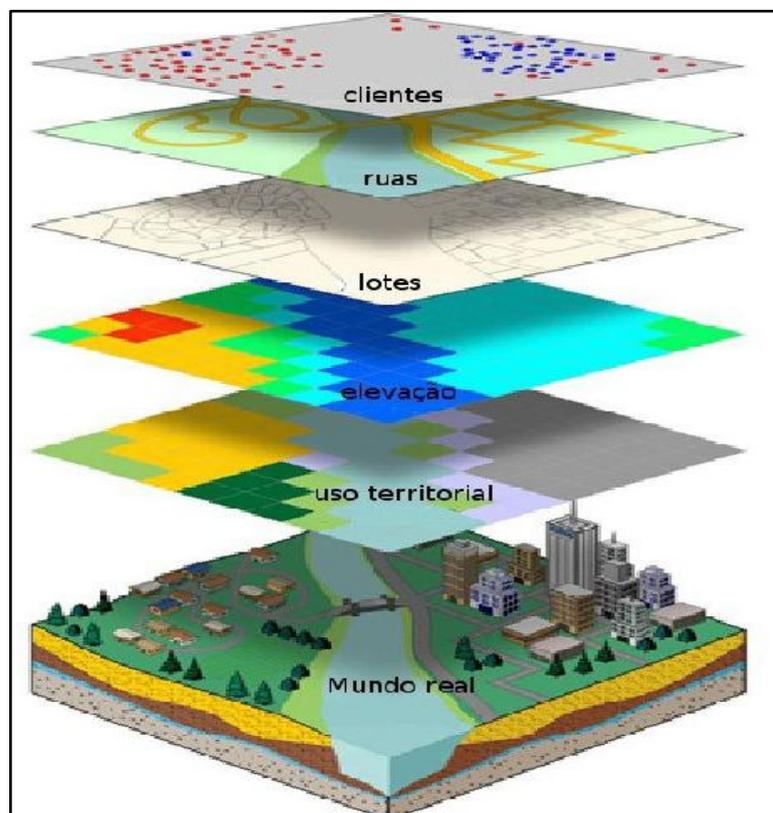


Figura 2: Camada de um SIG
Fonte: (JAIRO, 2014)

Essas ferramentas podem recuperar, armazenar, transformar, adquirir e emitir informações espaciais e o conjunto de dados descrevem objetos do mundo real e seu posicionamento.

Um sistema de SIG, pode ser utilizado para estudos de meio ambiente ou dos recursos naturais, em pesquisas de previsão de fenômenos naturais ou ainda no apoio a decisões de planejamento, considerando que os dados armazenados representam um modelo do mundo real (BURROUGH, 1986). Em resumo as principais características são:

- Uma única base de dados, com dados espaciais e informações de cadastro rurais e urbanas, podendo conter imagens de satélite, entre outras;
- Gerar mapas com manipulação de dados através de algoritmos;
- Recuperar, visualizar, consultar, plotar conteúdo da base de dados geocodificados.

2.2.1. Modelos de Representação de Dados em um SIG

Os dados armazenados em um sistema SIG, são tratados por procedimentos de análise geográfica, esses procedimentos podem estar em domínio vetorial ou domínio matricial (*rasters*), esse conjunto de procedimentos foi tratado por (TOMLIN, 1990). Denominado “Algebra de Mapas”, que são a base de implementações em diferentes sistemas.

A representação vetorial é utilizada para representar objetos pequenos em dimensões espaciais e usa duas coordenadas para informar sua localização, embora o tamanho ou dimensão de um objeto pode não ter muita importância, tendo como principal sua localização pontual.

Um exemplo típico é a representação de uma gleba em uma fazenda ou um lote em uma área urbana, que podem ser mostrados ou representados por um ponto na base espacial e trazer informações de área, proprietário, o seu uso e o seu valor venal.

Porém se tivermos um conjunto ordenado de pontos, e interligamos esses pontos por um segmento de reta (polígono aberto), podemos definir seus nomes como um ponto inicial e um ponto final que chamamos de nós e os pontos intermediários de vértices. Essas representações vetoriais geralmente servem para expressar graficamente: estradas, cursos de água, redes de saneamento, linhas de transmissão de energia elétrica.

Também pode-se ter um conjunto de pontos interligados por segmento de retas e ordenados tendo a diferença de o ponto inicial e final coincidirem, associando essa formação a um polígono que representa um área de uso de solo como: Talhões, lotes, quadras, propriedades rurais entre outros.

Essa forma de representação depende do propósito, ou seja a forma desse objeto no mundo real.

Na agricultura, a forma de representação é o *raster* ou modelo matricial, o talhão é representado por uma matriz composta por colunas e linhas que definem uma célula ou um quadrado, cada célula tem seus atributos (valores), esses além de conter um centróide (ponto), com um par de referência geolocalizada (latitude e longitude), possui valores dentro da área de abrangência que pode conter informações nutricionais do solo. A representação de um *raster* na agricultura é o

quadrado, porém também são representados por triângulos equiláteros ou hexágonos regulares.

Após a coleta de informações nutricionais no ponto (coordenada geolocalizada, latitude e longitude), esses dados podem ser interpolados e o resultado desse processo é um valor preciso de quanto uma determinada área, precisa de um determinado nutriente.

Na Figura 3 pode-se visualizar os dois tipos de modelos de representação de dados em um SIG.

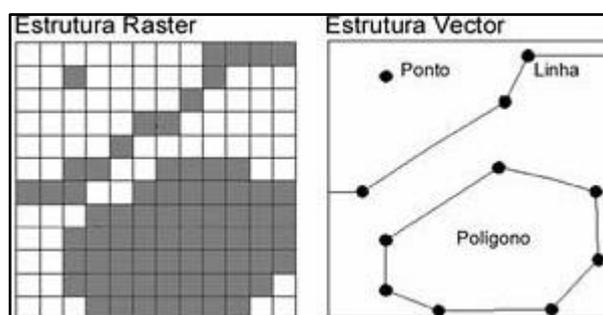


Figura 3: Modelos de representação de dados em um SIG
 Fonte: (SABOYA, 2005)

2.2.2. Padronização de Dados em um SIG

O processo de coleta de dados iniciou nos anos 50, os EUA e Inglaterra começaram a desenvolver automação no processo de coleta e armazenamento de dados espaciais, ainda que não seja considerado um processo de sistema de informação, foi determinante para que nos anos 80 a compilação desses dados representasse um sistema de informação geográfica.

A partir dos anos 90 o uso do geoprocessamento tornou-se uma ferramenta para apoiar a tomada de decisão e no final dessa década essas informações já estavam em um ambiente web.

Com a crescente busca por dados geográficos, o mercado precisava de uma padronização, em 1994 foi fundada a *Open Geospatial Consortium* (OGC), um consórcio de 369 companhias, agências governamentais e universidades que definiu padrões de intercâmbio de dados e especificações de interface, essa padronização e a criação da OGC foi fundamental para o mercado de sistemas de informações geográficas (SIG) que antes bancavam altos custos por soluções e informações proprietárias.

Em 1995 a *Oracle* entra para o consórcio, visualizando uma grande oportunidade de usar seu software de base de dados para armazenar as informações espaciais, dessa união surgiu um software livre para geoprocessamento o *OpenGis*.

Em 1996 é a vez da *Microsoft* entrar para o consórcio e também contribuir para o desenvolvimento e oportunidades do geoprocessamento para computação móvel.

Em 2000 surgiram serviços e transformações de coordenadas com o *Web Map Server* (WMS) e o *Geographic Markup Language* (GML), dando a um simples XML (*Extensible Markup Language*) a capacidade de suportar informações de geoprocessamento ubíquo e *standards-based*.

Em 2004 surge um padrão internacional baseado na especificação da interface *OpenGis Web*, o *Web Map Service* (WMS), que foi habilitado pelo comitê de informações geográficas da ISO: ISO/TC211 *Geographic Information/Geomatics*.

Essa especificação abstrata foi definida em 2 modelos, o primeiro é o essencial, que representa o mundo real, e o segundo representa o abstrato e como um sistema SIG irá representar, esse modelo é o mais complexo.

O modelo abstrato foi dividido em dezessete tópicos e para cada modelo abstrato existe um modelo essencial, desses cabe mencionar o primeiro tópico, Geometria de Feições (*Feature Geometry*). A feição é o “átomo”, de uma informação geográfica, é a representação abstrata dos fenômenos do mundo real, é uma localização relativa na terra (ponto, linha, polígono).

O quinto tópico, Feições do *OpenGis* (*The OpenGis Feature*), mostra uma rodovia constituída em muitos segmentos, ou um segmento de uma rodovia entre intercessões consecutivas, ou uma imagem georeferenciada ou um pixel de uma imagem de satélite georeferenciada.

O decimo terceiro tópico o Serviço de catálogo (*Catalog Services*), é um banco de dados especializado com informações de fontes geoespaciais, essas fontes devem ter interfaces de feições, coleções de feições, catálogos e metadados do *OpenGis*.

Com o objetivo da padronização nos softwares e informações de geoprocessamento surgiram algumas especificações:

- GML – *Geography Markup Language*;
- WMS – *Web Map Service*;

- SFS – *Simple Feature* – SQL;
- WFS – *Web Feature Service*.

Um dos protocolos mais utilizados para renderização de mapas entre os mais populares *middlewaresopensource* no mercado é o WMS, nesse protocolo é possível trafegar mapas pelo protocolo HTTP sem muitas complicações.

O WFS (*Web Feature Services*) outro protocolo que evoluiu para ser um mediador transacional de atributos alfa-numéricos e espaciais, alguns padrões OGC foram muito bem aceitos pela comunidade como GML, GeoJSON, KML e outros.

Para serviços como leitura, podemos utilizar o WFS de forma convencional, que possibilita a consulta e atributos espaciais e de metadados, podendo ser usado em conjunto à recursos de filtros com CQL, que torna possível operações lógicas com cláusula da SQL ANCI e funções para topologias, métricas para interação como atributos espaciais.

Também é possível fazer o recurso SO WFS-T (*Web Feature Service Transact*) que possibilita o CRUD (*Create, Read, Update e Delete*) em fontes de dados configurada por algum *middleware* que suporte o protocolo WFS.

2.3. ARQUITETURAS ORIENTADAS A SERVIÇO

Esta seção descreve os conceitos de duas arquiteturas basedas em serviços, a seção 2.3.1 descreve a arquitetura SOAP (*Simple Object Access Protocol*) e a seção 2.3.2 descreve a arquitetura REST (*Representational State Transfer*).

2.3.1. SOAP: *Simple Object Access Protocol*

O SOAP está longe de ser aposentado, apesar de ser o avô das interfaces de serviços *WEB*, tendo a vantagem de utilizar qualquer meio de transporte existente, diferente do *REST* que só utiliza HTTP/HTTPS.

Baseado em *Extensible Markup Language* (XML), ele envia junto com seu envelope o conteúdo da mensagem, informações de como processá-las, e ainda um conjunto de regras de codificação. O Serviço RPC é executado e o retorno também é através de um documento XML formatado.

Na Figura 4 pode-se observar a estrutura do arquivo XML com informações para execução do soap.

```

<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
  <env:Header>
    <n:alertcontrol xmlns:n="http://example.org/alertcontrol">
      <n:priority>1</n:priority>
      <n:expires>2001-06-22T14:00:00-05:00</n:expires>
    </n:alertcontrol>
  </env:Header>
  <env:Body>
    <m:alert xmlns:m="http://example.org/alert">
      <m:msg>Pick up Mary at school at 2pm</m:msg>
    </m:alert>
  </env:Body>
</env:Envelope>

```

Figura 4: Exemplo de uma mensagem SOAP
Fonte: (W3C, 2007)

2.3.2. REST: *Representational State Transfer*

O Rest é muito simples e pode ser adotado em praticamente qualquer cliente ou servidor com suporte a HTTP/HTTPS. A utilização de infraestrutura existente e a facilidade de aprendizado também é um dos fatores favoráveis a sua utilização.

REST é uma técnica de engenharia de software, o termo se originou em 2000, em uma teste de doutorado escrita por Roy Fielding, um dos principais autores da especificação da arquitetura *REST*, muito utilizado por sites da Web.

Utiliza-se recursos *REST* quando acessa-se uma URL através de uma requisição HTTP, as quais são descritas abaixo:

- *GET* – Recupera informações sobre o recurso identificado pela URL;
- *POST* – Adiciona informação usando o recurso da URL;
- *PUT* – Adiciona ou modifica um recurso da URL passada;
- *DELETE* - Remove o recurso representado pela URL passada;
- *HEAD, OPTIONS e TRACE* – Recupera metadados da URL passada.

2.4. MIDDLEWARES

Esta seção descreve os *middlewares* mais utilizados no desenvolvimento de aplicações *Web* que utilizem mapas, dados georreferenciados, entre outros. Estes são os responsáveis pela junção e consistência entre diferentes softwares, eles fazem a mediação, ocultando direntes protocolos e sistemas operacionais.

Na seção 2.4.1 será apresentado o *middleware GeoServer* e na seção 2.4.2 será apresentado o *MapServer*.

2.4.1. GeoServer

O Geoserver, e um aplicativo (SOA) , que nas versões mais novas, tem uma “extensão *REST*, que por sua vez, fornece uma interface *RESTful*, por meio do qual, os clientes podem configurar uma instância *GeoServer* através de chamadas HTTP simples. Com isso os clientes podem configurar programaticamente os dados geridos pelo *GeoServer*” (GEOSERVER, 2009).

2.4.2. MapServer

O *Mapaserver*, e um aplicativo (SOA), também não é uma aplicação de SIG completa, foi criado pela universidade de Minnesota com o apoio da *NASA*, algumas empresas particulares, a Univali (empresa brasileira). Também possui integração com o *Postgis*.

2.5. ARQUITETURA WEB

Ao navegar na Internet os navegadores interpretam um conteúdo recebido disponibilizarm para o usuário, este conteúdo é recebido e transferido através do HTTP (*Hypertext Transfer Protocol*) que é o principal protocolo de comunicação entre o navegador e o servidor.

Host: "google.com.br"
User-Agent: "Mozilla/5.0 (Windows NT 10.0; WOW64; rv:42.0) Gecko/20100101 Firefox/42.0"
Accept: "text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8"
Accept-Language: "pt-BR,pt;q=0.8,en-US;q=0.5,en;q=0.3"
Accept-Encoding: "gzip, deflate"
Cookie: "PREF=ID=1111111111111111:FF=0:LD=pt-BR:TM=1445269985:LM=1445270474:V=1:S=7"
Proxy-Authorization: "Basic bHVpenl6MXFhejJ3c3g="
Connection: "keep-alive"

Figura 5: Exemplo de requisição HTTP
Fonte: Autoria Própria

A navegação na Internet funciona basicamente através de uma requisição do navegador, Figura 5, que pode ser *GET*, *POST* e uma resposta, Figura 6, do

servidor que contém cabeçalhos, o código HTML que será interpretado pelo navegador e possíveis códigos de sucesso ou falha.

Cache-Control: "public, max-age=2592000"
Connection: "keep-alive"
Content-Length: "222"
Content-Type: "text/html; charset=UTF-8"
Date: "Thu, 26 Nov 2015 19:59:33 GMT"
Expires: "Sat, 26 Dec 2015 19:59:33 GMT"
Location: "http://www.google.com.br/"
Server: "gws"
Via: "1.1 [Angus-Server] (squid/3.2.3)"
X-Cache: "MISS from [Angus-Server]"
X-Frame-Options: "SAMEORIGIN"
X-XSS-Protection: "1; mode=block"

Figura 6: Exemplo de resposta HTTP
Fonte: Autoria Própria

2.6. LINGUAGENS DE PROGRAMAÇÃO E MARCAÇÃO

Esta seção apresenta as linguagens de programação e marcação necessárias para o desenvolvimento de aplicações *Web* do lado servidor e do lado cliente. A seção 2.6.1 apresenta a linguagem de programação do lado servidor Java.

A seção 2.6.2 apresenta a linguagem de programação *Web* do lado cliente, *Javascript*. As seções 2.6.3 e 2.6.4 apresentam linguagens de marcação, para a estrutura e para apresentação visual, respectivamente.

2.6.1. Java

O Java surgiu num universo de programação com deficiência em gerenciamento de memória, alto custo de desenvolvimento e problemas para uma aplicação rodar em vários sistemas operacionais. O Java foi criado pela antiga *Sun Microsystems* e mantida através de um comitê¹.

A *Sun* criou uma equipe conhecida como *Green Team* para desenvolver inovações tecnológicas em 1992. O líder desse time era James Gosling, que é considerado o pai do Java. A idéia foi criar uma VM (máquina virtual) para pequenos

¹<http://www.jcp.org>

dispositivos, aparelhos eletrônicos, aparelhos de tv e celulares e dominar o mercado de eletrônico.

Na época não deu certo por haver muita concorrência e jogo de interesses, porém hoje sabemos que o Java domina o mercado dos aparelhos eletrônicos.

Em 2009 a *Oracle* comprou a *Sun*, fortalecendo a marca. A *Oracle* sempre foi, junto com a *IBM*, uma das empresas que mais investiram e fizeram negócios através do uso da plataforma Java. Em 2014 surge a versão Java 8 com mudanças interessantes na linguagem.

O Java utiliza a VM (Máquina virtual), onde existe entre o SO (Sistema Operacional) e a aplicação uma camada responsável por “traduzir” o que sua aplicação deseja fazer para as respectivas chamadas do SO onde está rodando no momento. Dessa forma a maneira com que o SO abre uma aplicação no Windows é igual no Linux, fazendo com que um desenvolvedor não se preocupe com detalhes do SO.

Na Figura 7 pode-se ter um exemplo de código na linguagem Java demonstrando uma classe.

```
public class Pessoa{

    public String nome;
    public int idade;

    public Pessoa(String nome, int idade){
        this.nome = nome;
        this.idade = idade;
    }

    public String toString(){
        return "Pessoa: " + this.nome + " com idade " + this.idade;
    }

}
```

Figura 7: Código de exemplo Java
Fonte: Autoria Própria

2.6.2. Javascript

O *Javascript* é uma linguagem de script que é incorporada a um documento HTML. Ele foi a primeira linguagem de script criada para a *Web* e trouxe junto com ele melhorias para o lado cliente das aplicações *Web* permitindo que estas se tornassem mais ricas e atraentes para os usuários.

Ele não requer um compilador, ao contrário da linguagem Java e outras linguagens de programação utilizadas durante o desenvolvimento *Web*.

Desenvolvido pela *Netscape* me 1995, era chamado de *LiveScript* que fornecia uma linguagem muito simples para o browser da *Netscape*, criticado por suas inúmeras falhas e falta de alertas de erros. A mudança de nome ocorreu em 4 de dezembro de 1995 com a associação da *Netscape* com a *Sun*, o *LiveScript* tornou-se o *Javascript*.

Com a entrada da *Microsoft* na criação de uma linguagem voltada a *script* com o *Jscript*, definiram uma padronização para linguagens de *script*, a ECMA 262 criado pela ECMA *European Computer Manufacturers Association* (NOUIRA, 2015).

Na Figura 8 tem-se um exemplo de código simples em *Javascript* que será executado no clique de um botão.

```
<button onclick="exemplo()">Botão</button>

<script>
  function exemplo(){
    alert("Código de exemplo");
  }
</script>
```

Figura 8: Código de exemplo Javascript
Fonte: Autoria Própria

2.6.3. HTML: *HyperText Markup Language*

O HTML é utilizado para formatação do conteúdo de uma página *Web*, uma linguagem de marcação de hipertexto dedicada a construção de páginas *Web* e a navegação entre as diferentes páginas.

Para construir uma página *web* com HTML, é necessário apenas um editor de texto e o conhecimento dos códigos que compõe a linguagem, esses códigos são conhecidos como “*tags*” e servem para indicar a função de cada elemento da página *web*, elas funcionam como comandos de formatação de texto, com formulários, links, imagens, tabelas.

Essa linguagem é identificada pelos navegadores, que processam a codificação do texto e apresentam a página conforme especificada. Foi criada por Tim Barnes Lee na década de 1990 e é controlada pela W3C (*Word Wide Web Consortium*).

As versões de evolução da HTML incluem XHTML, com a sintaxe mais rigorosa, baseada em XML e HTML5 a quinta versão do HTML que traz novos recursos a linguagem principalmente manipulação de conteúdo gráfico e multimídia.

Na Figura 9 tem-se exemplo de um código simples em HTML.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Título da Página</title>
  </head>
  <body>
  </body>
</html>
```

Figura 9: Exemplo de código HTML
Fonte: Autoria Própria

2.6.4. CSS: *Cascading Style Sheets*

CSS é um mecanismo simples para adicionar estilos, como por exemplo, fontes, cores, espaçamentos aos documentos Web (W3Schools, 2015) tendo como objetivo apresentar o documento HTML.

Na Figura 10 é mostrado um seletor css “a”, com a propriedade “*background-color*”, cor de fundo, com o valor de “*yellow*”, amarelo.

```
a { background-color : yellow }
```

Figura 10: Exemplo de código CSS
Fonte: Autoria Própria

2.7. FERRAMENTAS

Esta seção apresenta as ferramentas necessárias para apresentação de mapas na plataforma *Web*. A seção 2.7.1 descreve a ferramenta *Open Layers* e a seção 2.7.2 descreve o *Google Web Toolkit*, uma iniciativa Google para este tipo de tarefa.

2.7.1. *Open Layers*

A ferramenta *OpenLayers* surgiu em 2005 (*where 2.0*), motivado pela necessidade de negócios da *MetaCarta*.

Essa promissora API passou por várias revisões internas antes mesmo de ser lançado oficialmente, foi reescrita e lançada em 2006 no *where 2.0*, atualmente mais de 10.000 aplicações já a utilizam.

A API é utilizada para construir aplicações dinâmicas de *webmapping*, com a integração de várias tecnologias, como *Ajax*, *JavaScript* nos padrões web 2.0, com suporte a padrões de código aberto, licença no padrão BSD, um Projeto *Open Source Geospatial Foundation* (OSGeo).

A possibilidade da criação de mapas interativos, podendo conter informações geoespaciais e geográficas, conectados em vários servidores como *Google maps*, *GeoServer*, *OpenStreetMaps*, *Bing maps*, *Yahoo local maps*.

OpenLayers com sua arquitetura modular possui diversos tipos de camadas, como: *WMS*, *Vector*, *KML*, *Google*, *VirtualEarth*, imagens, entre outros. Possui também várias geometrias como, por exemplo, pontos lineares, polígonos, entre outros e vários controles para *zoom*, panorâmica, medir distâncias, modificar, entre outros.

Possui suporte à vários formatos como: WMS, WFS, OSM, KML, GeoJSON, GeoRSS entre outros. Sua comunicação também é diversificada com vários protocolos como: HTTP, SOS, WFS, OGC.

Em *OpenLayers*, um mapa, Figura 12, é um conjunto de camadas e várias interações e controles para lidar com a interação do usuário. Um mapa é gerado com três ingredientes básicos: marcação, declarações de estilo, e código de inicialização, como mostra a Figura 11.

```

3      <!doctype html>
4      <html lang="en">
5          <head>
6              <link rel="stylesheet" href="/ol.css" type="text/css">
7              <style>
8                  #map {
9                      height: 256px;
10                     width: 512px;
11                 }
12             </style>
13             <title>OpenLayers 3 example</title>
14             <script src="/loader.js" type="text/javascript"></script>
15         </head>
16         <body>
17             <h1>My Map</h1>
18             <div id="map"></div>
19             <script type="text/javascript">
20                 var map = new ol.Map({
21                     target: 'map',
22                     layers: [
23                         new ol.layer.Tile({
24                             title: 'Global Imagery',
25                             source: new ol.source.TileWMS({
26                                 url: 'http://demo.opengeo.org/geoserver/wms',
27                                 params: {LAYERS: 'nasa:bluemarble', VERSION: '1.1.1'}
28                             })
29                         })
30                     ],
31                     view: new ol.View({
32                         projection: 'EPSG:4326',
33                         center: [0, 0],
34                         zoom: 0,
35                         maxResolution: 0.703125
36                     })
37                 });
38             </script>
39         </body>
40     </html>

```

Figura 11: Exemplo de código *OpenLayers*
Fonte: (*OpenLayers*, 2015)



Figura 12: Exemplo de mapa gerado com *OpenLayers*
Fonte: (*OpenLayers*, 2015)

Como demonstrado na Figura 11, um mapa é gerado reunindo marcação, declarações de estilo, e código de inicialização. Com esse conjunto, podemos gerar mapas através da API do OpenLayers 3.0.

Atualmente na versão 3 do OL, foram utilizados novos recursos para execução no *frontend* das aplicações podendo facilmente se adaptar a *tablets*, *smartphones*, *tablets* e *notebooks* tratar a quantidade de pixels que o dispositivo irá estabelecer, e recursos para poder trabalhar com novos arquivos existentes no mercado novos padrões, trabalhar com *cache* que nas versões anteriores não existiam.

Na versão 3 do OL foram utilizados recursos como *bootstrap*, HTML, CSS e *Javascript*, bibliotecas populares que proporcionam um desenvolvimento ágil na *web* ou *mobile*.

Em um SIG, a representação de um objeto real por uma fonte de dados, seja ela um arquivo, banco de dados, entre outros pode ser chamada em inglês como *Layers*, no OL3 são *containers* que recebem os seus dados a partir de uma fonte de dados.

O “*ol.layer.tile*” é relacionado ao *layer* ou camada armazenada em um *middleware* que pode ser configurada como *cache* e outros parâmetros como “*ol.layer.image*”, que geralmente é usada para interpretar uma imagem não georeferenciada em um mapa *canvas* georeferenciado, outro parâmetro é o “*ol.layer.vector*”, que interpreta padrões de dados como GML, GeoJSON e renderiza-os como uma camada vetorial.

O Conjunto de camadas *Map*, “*ol.map*” servem para fins de renderização e visualização.

2.7.2. Google Web Toolkit

Com a crescente utilização da *Web*, o uso de linguagens como o *Javascript* ganharam força graças a utilização de aplicativos como *Gmail* e *Google Maps* tendo como maior benefício aprimorar a experiência do usuário.

O *JavaScript* e o *Ajax*, técnica para transferência de dados de forma assíncrona na *Web*, estão a disposição há anos, mas a grande maioria dos programadores os ignoravam pela dificuldade de aprenderem e de os utilizarem.

O *Google* com o *Google Web Toolkit* (GTW) ajudou os usuários a utilizarem essas linguagens e técnicas de forma mais rápida e eficiente e de forma gratuita. (STÉPHANIE ANTOINE, JULIE DUBOIS E JEAN PHILIPPE RETAILLÉ, 2006).

A maioria dos navegadores possuem diferenças na forma de tratar e prestar suporte a *JavaScript* e *Ajax*, o GTW ajuda a integrar todas essas linguagens para todos os navegadores porque transforma o código Java em *JavaScript* e *Ajax*, sendo assim o programador precisa apenas aprofundar seus conhecimentos em uma única linguagem/técnica de programação.

3. METODOLOGIA

Esta seção descreve como foi realizado o desenvolvimento do *software*, objetivo deste trabalho. A seção 3.1 descreve o processo de coleta de dados e a seção 3.2 descreve o desenvolvimento.

3.1. COLETA DE DADOS

A coleta de dados consistiu-se em dois momentos, o primeiro, foi um levantamento dos *softwares* que os agrônomos utilizavam na agricultura de precisão, um vídeo mostrando todos os processos realizados e quais as dificuldades encontradas, sendo uma delas a divulgação dos mapas.

O segundo um questionário com 6 (seis) perguntas objetivas, discutidas no capítulo 5 deste trabalho. Na elaboração do questionário procuramos investigar se o geopublicador conseguiu agilizar o processo de distribuição e aplicação dos mapas na agricultura.

Esse instrumento foi aplicado a dois agrônomos da cidade de Castro, em outubro de 2015. Após essa coleta de dados foi elaborado um resultado consolidado e transcrito no capítulo 5.

Utilizou-se a análise textual discursiva (ATD), (MORAES; GALIAZZI, 2007) para analisar os dados.

3.2. DESENVOLVIMENTO DO SOFTWARE

No desenvolvimento deste trabalho, as etapas seguidas foram gerar várias versões *beta*, essas versões foram disponibilizadas a um agrônomo para efetuar testes e levantar possíveis bugs no sistema.

Desde o início o GeoPublicador foi integrado com o portal www.sevengp.com.br, com isso o agrônomo usuário já efetuava testes no ambiente real.

Não foi adotada nenhuma metodologia de desenvolvimento de *software* e nem de gestão de projetos. Todo o desenvolvimento deu-se através de testes com o usuário e sobre os requisitos levantados na etapa de coleta de dados.

4. SISTEMAS SIMILARES

Este capítulo apresenta os sistemas similares utilizados para extração de requisitos funcionais e visuais que poderiam estar presentes no geopublicador proposto neste trabalho.

A seção 4.1 apresenta o sistema *QGIS* e a seção 4.2 apresenta o sistema *Farm Work*.

4.1. QGIS

Atualmente existe grande quantidade de *softwares* voltados a agricultura de precisão, entre eles o robusto *QGIS*, ferramenta baseada em *software* livre (*OpenSource*). Essa ferramenta utiliza um banco de dados *PostgreSQL*, com a extensão espacial *PostGIS*, que também é baseado em *software* livre.

O único detalhe a ser mencionado é que o *QGIS* é uma aplicação *desktop*.

Não foi utilizado nenhum requisito do *QGIS*, por não ser muito utilizado na região de Castro-PR, onde o geopublicador foi aplicado.

A Figura 13 mostra uma tela do sistema *QGIS*.

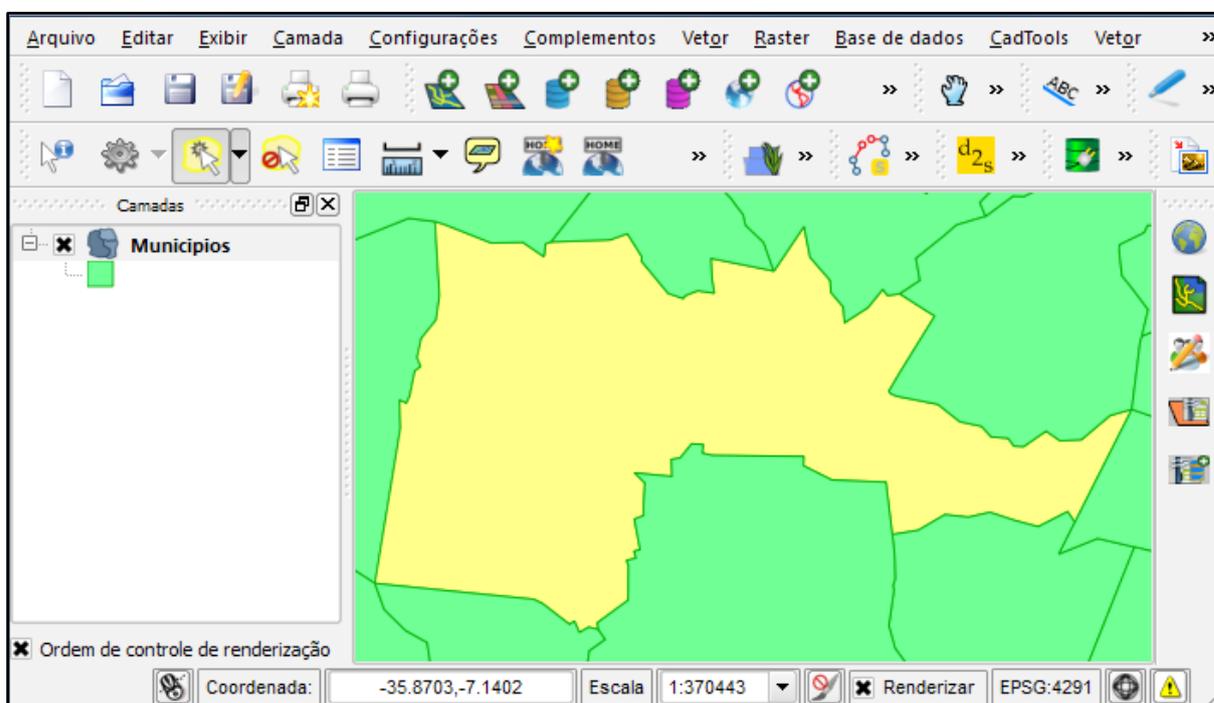


Figura 13: Tela QGIS
Fonte: (MEDEIROS, 2015)

4.2. FARM WORK

O *Farm Work* da *Trimble* é o mais famoso *software* para geoprocessamento e agricultura de precisão. Essa robusta ferramenta é utilizada para gerar mapas de aplicação e cobertura, através de importações de análise de solo com diversos tipos de mapas, os mapas que necessitam são interpolados utilizando o inverso da distância, tanto em formato *raster* (grades) quanto em formato vetorial (áreas).

Permite soluções de mapeamento de talhões e adição de camadas de informação aos mesmos. As funcionalidades incluem baixar imagens aéreas, desenhar ou importar contornos de campo, criar zonas de manejo, planejamento para a fazenda, além da importação e exportação dos dados para monitores e controladores da agricultura de precisão. Como podemos observar na figura 6 o *Farm Work* também é uma aplicação desktop.

Algumas das funcionalidades do GeoPublicador foi extraído do *Farm Work*, aplicativo mais utilizado pelos agrônomos da região de Castro-PR como:

- A forma de listar os projetos;
- A visualização centralizada do mapa;
- A visualização de faixa de valores;
- A forma de marcar o centroide no mapa;

A Figura 14 mostra uma tela do sistema *Farm Work*.

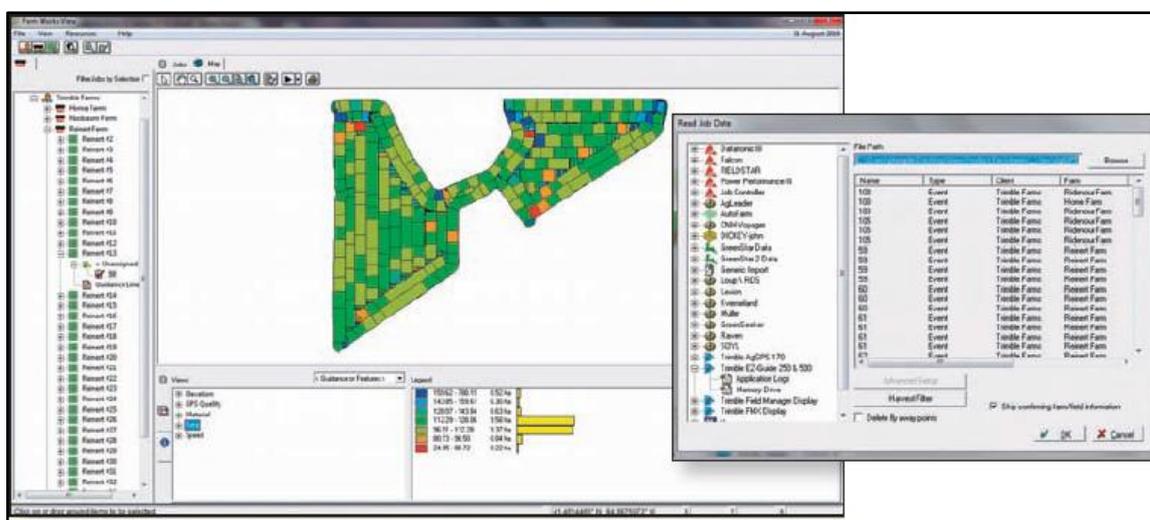


Figura 14: Tela do sistema *Farm Work*
Fonte: (GEOAGRI, 2012)

5. GEOPUBLICADOR

Este capítulo descreve o geopublicador proposto neste trabalho, desenvolvido para facilitar a divulgação dos resultados finais para agricultura de precisão. A seção 5.1 descreve a estrutura do geopublicador, a seção 5.2 descreve superficialmente o interpolador, já que este não é foco deste trabalho, a seção 5.3 descreve o processo de visualização dos mapas e a seção 5.4 apresenta as telas do sistema proposto.

5.1. ESTRUTURA

Com o objetivo de criar um facilitador para agrônomos e produtores rurais e/ou qualquer pessoa que tenha a necessidade de visualizar mapas de aplicações ou mapas de cobertura, surgiu o GeoPublicador *web*.

O GeoPublicador poderá ler vários tipos de arquivos geradores, porém nesse primeiro momento foi utilizado somente o arquivo *Shapefile* que é tido como padrão, e também o mais utilizado pelos *softwares* geradores de mapas.

O Geopublicador foi desenvolvido com banco de dados *OpenSource Postgres* com a utilização da extensão espacial *PostGis* e foi integrado com outras rotinas que complementam essa estrutura.

Foi contratado um *cloud server* na *Localweb*, empresa de hospedagem de serviços *Web*, onde foi instalado os seguintes *softwares*:

- *Tomcat 6.0.*
- *PostGris , com extensão espacial PostGis.*
- *GeoServer.*
- *OpenLayers*
- *Google Web Toolkit*

O domínio escolhido foi o www.sevengp.com.br sendo que o nome *seven* vem da palavra inglesa sete somando-se a sigla de Gestão da Produção.

Através do portal o usuário consegue executar cadastros básicos como produtor, fazenda, gleba, cultura, safra, siglas, elementos, fórmulas, entre outros.

Esses cadastros básicos vão incorporar a estrutura para a gestão da produção e da organização dos projetos, onde esses projetos são organizados dentro de cada safra, expl 2014/2015, 2015/2015.

É nessa estrutura que são salvos os arquivos *shapfiles*, gerados por qualquer outro software de Agricultura de Precisão (AP), podendo ser apenas contorno, ou mapas inteiros de aplicação ou cobertura.

Dentro dessa estrutura o usuário salva seus arquivos *shapfiles* pelo *software* interpolador no banco de dados. Selecionando o produtor, fazenda, gleba, cultura, safra e projeto que geralmente é salvo com o nome do mapa, expl contorno, aplicação NPK, entre outros.

Esse arquivo então poderá ser visualizado através de um computador pessoal, notebook, fablets, tables e smartphones.

5.2. VISUALIZAÇÃO DOS MAPAS

O GeoPublicador visualiza o mapa através do GWT (*Google Web Toolkit*) que tem como objetivo facilitar a vida dos desenvolvedores, ele possui um conjunto de ferramentas, API e componentes visuais que facilitam a criação de modernas e dinâmicas interfaces (JUNIOR, 2009).

No início do desenvolvimento, foi elaborado um vídeo com um agrônomo utilizando o *Farm Work*, através dele alguns requisitos e funcionalidades importantes foram estudadas para serem aplicadas tanto no Interpolador, como no GeoPublicador.

Com essa ferramenta o GeoPublicador já nasceu com o “DNA” herdado do famoso aplicativo, e ainda, ganhou outras funcionalidades importantes que facilitaram a utilização pelos agrônomos e empregados rurais.

Na Figura 15, mostra-se a sequência de uma escolha e a visualização de um mapa de aplicação e após selecionado. O mapa é do produtor EHK, Fazenda BV, Gleba 10C, safra 2013/2014 projeto 2013_BV_103_C1.

Após selecionado, é carregado o mapa (*Shapefile*) relacionado com a sequência da escolha, e algumas informações pertinentes ao projeto já são carregadas na tela, como informações das faixas de valores referente as cores do mapa tornando-se uma “legenda”.

Logo abaixo da tabela de faixa de valores é mostrado qual cultura, qual a área do mapa e que tipo de nutriente foi aplicado.

Cabe ainda ressaltar outras informações e funcionalidades do GeoPublicador, como a informação do elemento aplicado logo acima do mapa. No lado direito a barra de ferramentas com as seguintes opções:

- Exibir faixa de valores;
- Imprimir um PDF do mapa;
- Medir distâncias do mapa;
- Identificar feições, ou seja, saber qual valor de quantidade a ser aplicada em cada faixa de cores;
- Centralizar o mapa na tela;
- Exportar o *Shapefile* do projeto em *zip*.

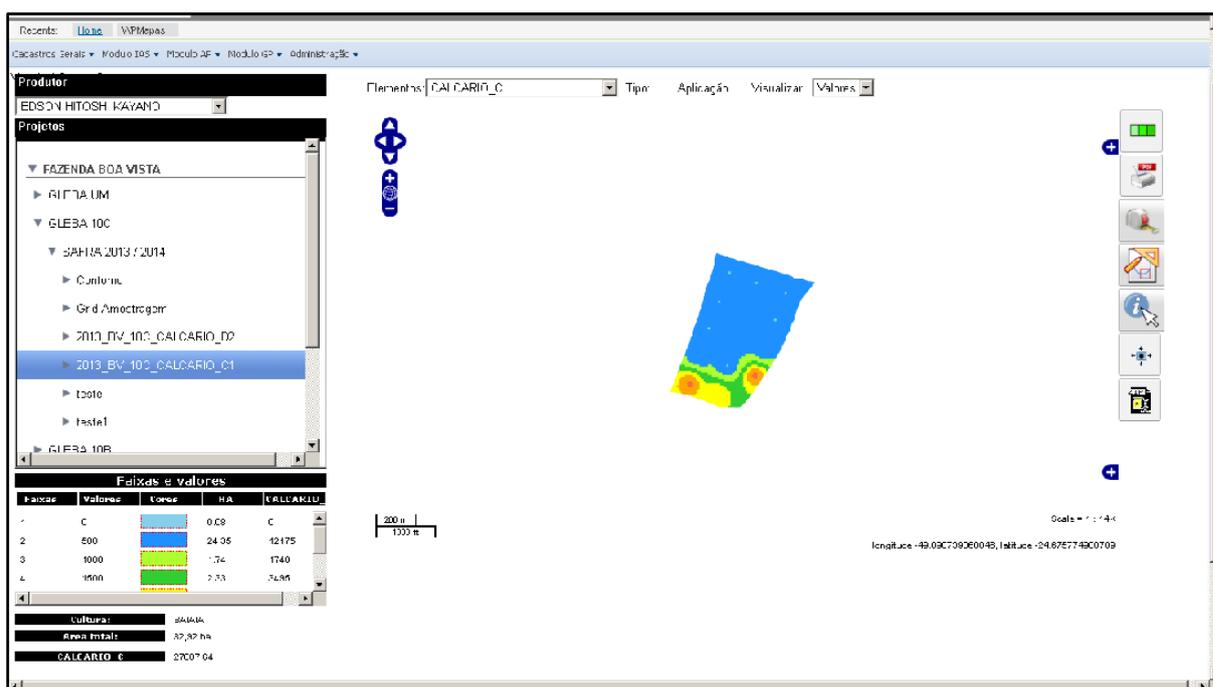


Figura 15: Visualização de um mapa gerado no geopublicador
Fonte: Autoria Própria

5.3. TELAS

A sequência de telas nesta seção mostra a organização dos “projetos”, para cada produtor que utiliza o portal *Sevengp*.

Na Figura 16 mostra-se o portal www.sevengp.com.br, é através dele que os usuários tem acesso ao GeoPublicador *Web*, além de outros módulos como a

Gestão de Produção e o IAS, onde os usuários podem gerar análise de solo simplificada para pequenos agricultores que não possuem ferramentas de Agricultura de Precisão e querem fazer aplicações customizadas.

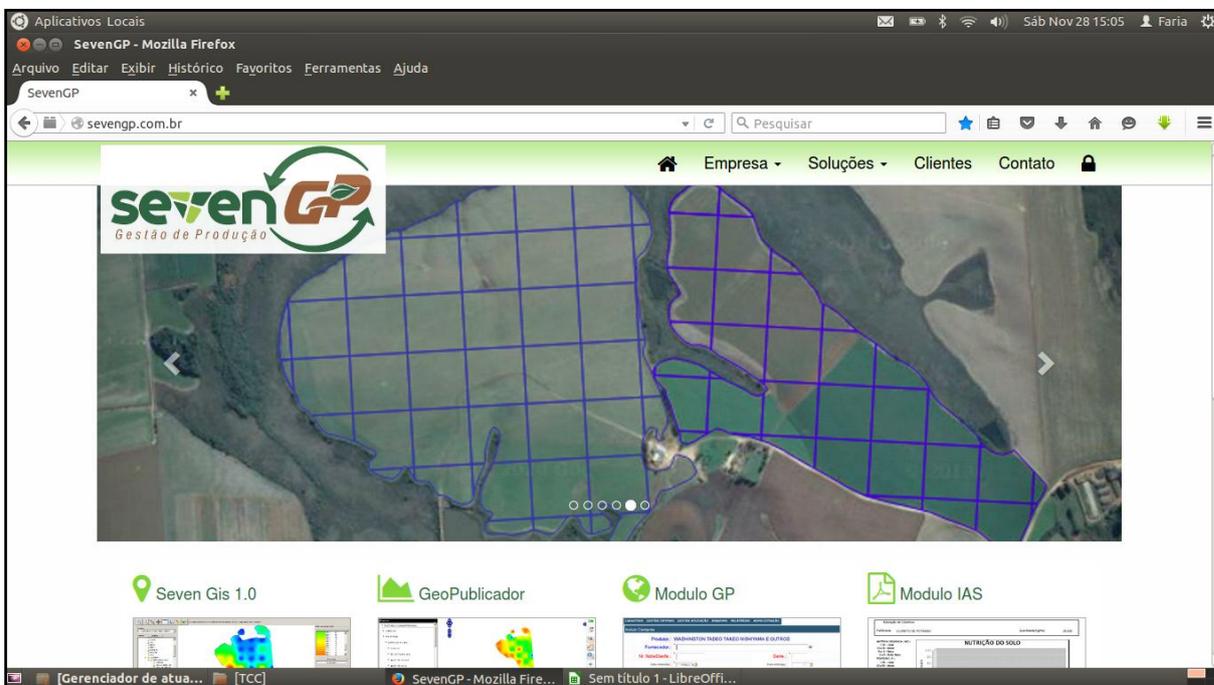


Figura 16: Portal de acesso ao geopublicador
Fonte: Autoria Própria

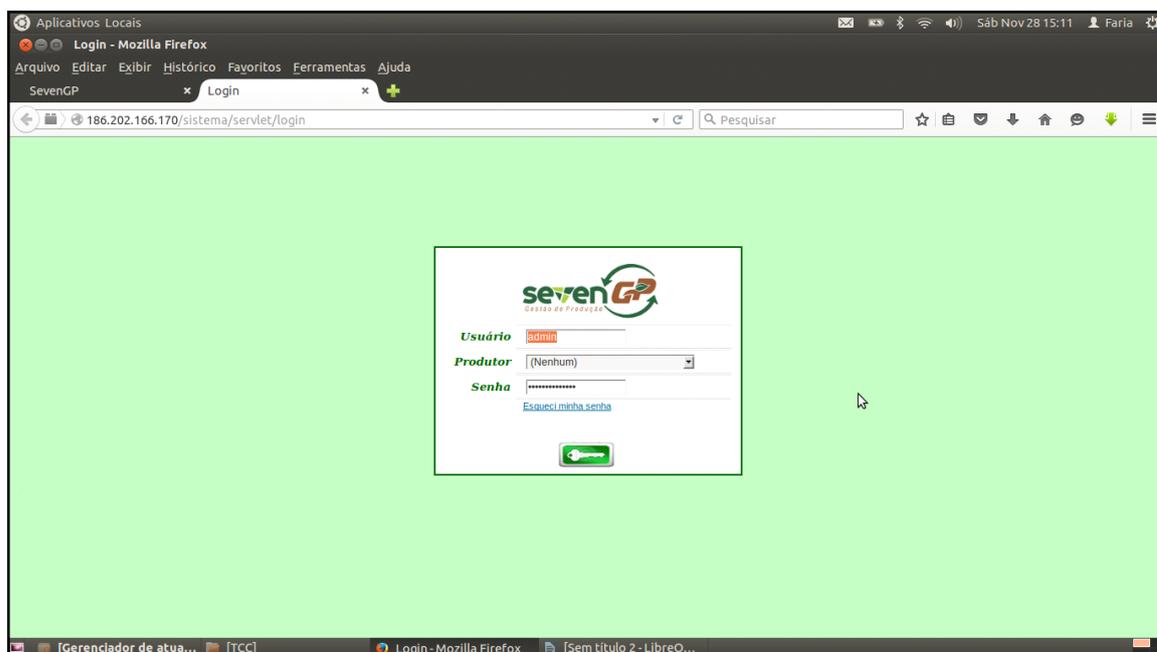


Figura 17: Tela de login do geopublicador
Fonte: Autoria Própria

A Figura 17 mostra o *login* de acesso, onde após informar o nome do usuário o site seleciona todos os produtores que o usuário tem controle, além desse filtro o sistema pode ser implantado em outro banco de dados e ser chamado pelo site do próprio cliente onde ele tem acesso total a cadastros, fórmulas e outras funcionalidades.

A Figura 18 exibe o menu do sistema, que pelo portal *SevenGp*, o usuário terá acesso apenas aos módulos por ele contratado, em caso de base de dados separada, esse menu é dinâmico podendo ser modificado de acordo com a necessidade de cada usuário.

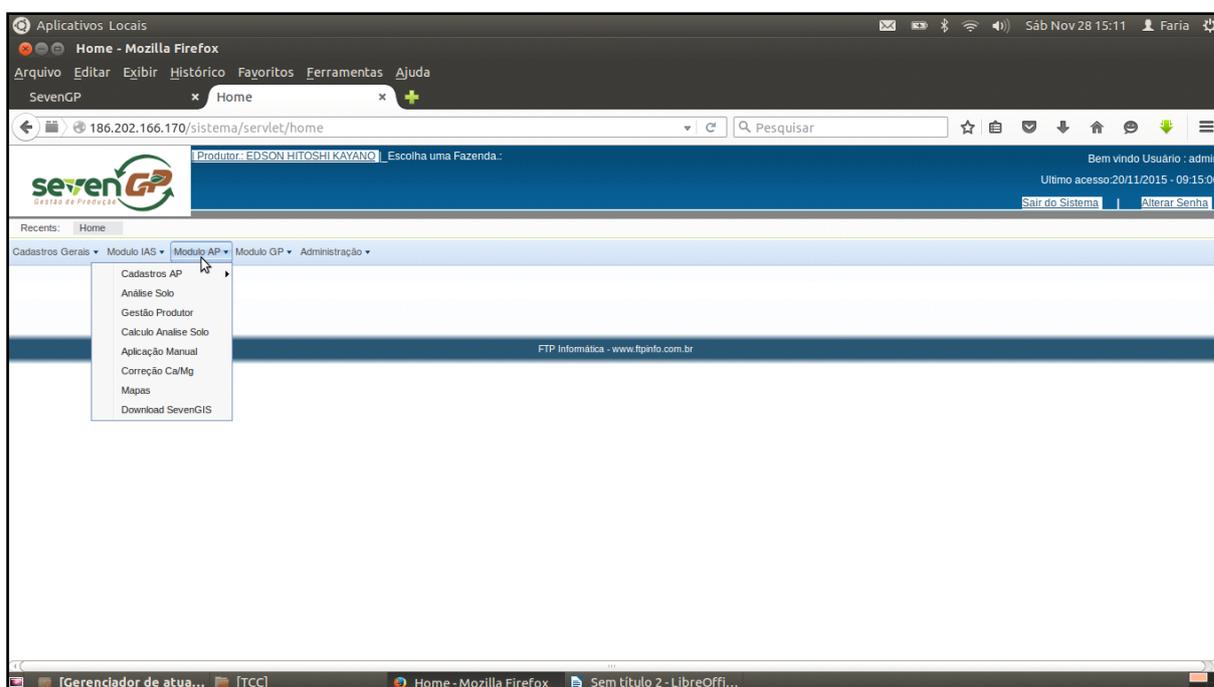


Figura 18: Tela com os menus do sistema
Fonte: Autoria Própria

A Figura 19 mostra o menu lateral a esquerda, onde o sistema busca no banco de dados todas as fazendas que o usuário tem acesso e disponibiliza a lista para sua escolha.

A Figura 20 mostra o menu lateral a esquerda, onde o sistema busca no banco de dados todas as glebas do produtor escolhido pelo usuário.

A Figura 21 exibe o menu lateral a esquerda, onde o sistema busca no banco de dados todas as safras e seus "projetos", como mapas de contorno que é mostrado após a seleção, no centro o GeoPublicador mostra o mapa selecionado.

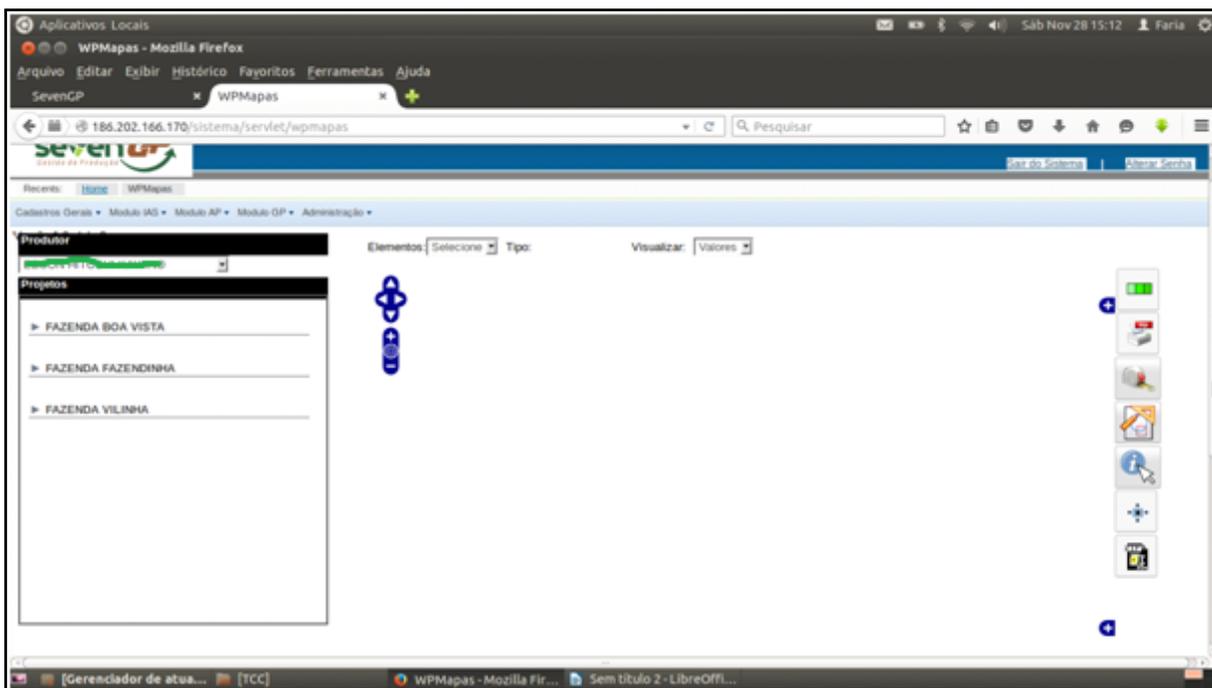


Figura 19: Tela de busca de fazendas
Fonte: Autoria Própria

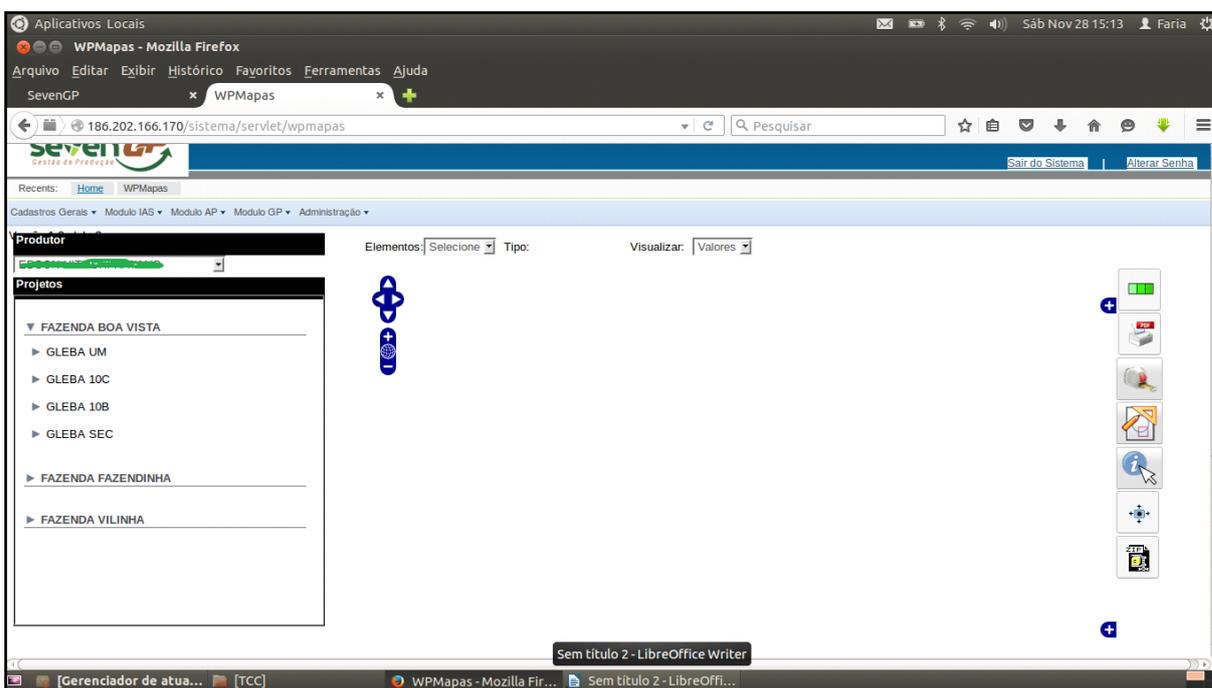


Figura 20: Tela de busca de glebas
Fonte: Autoria Própria

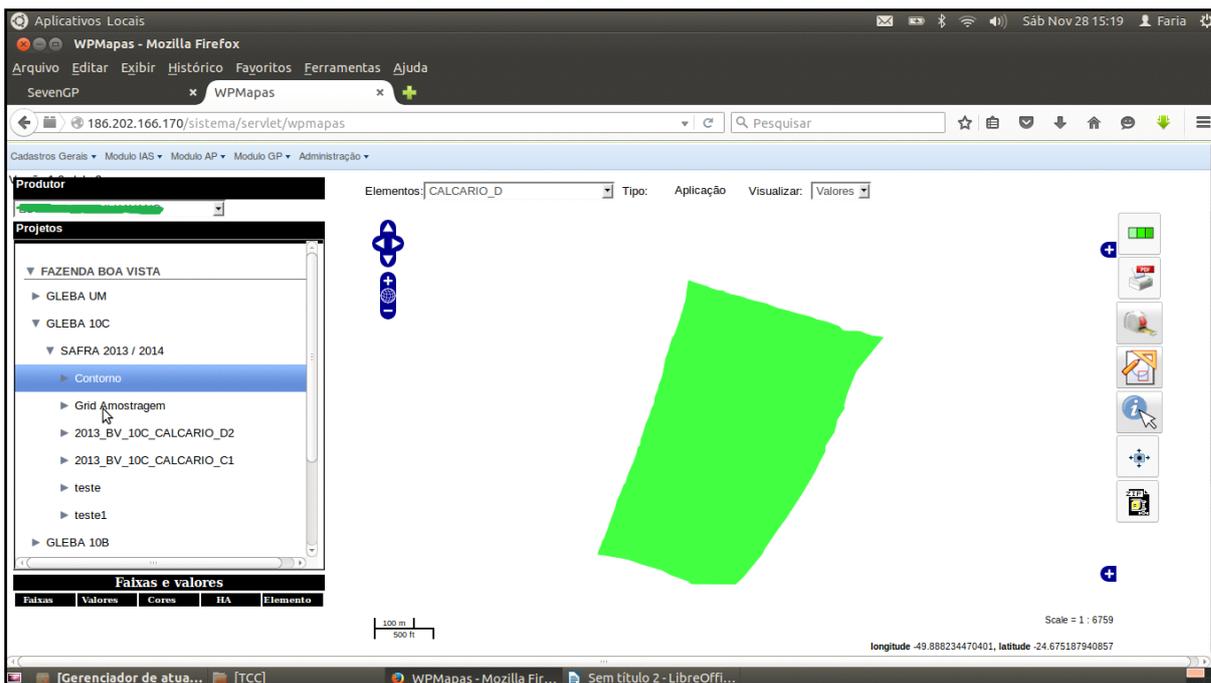


Figura 21: Imagem do mapa selecionado
Fonte: Autoria Própria

A Figura 22 mostra um mapa de amostra, o ponto exato, local onde vai ser retirada uma quantidade de terra e levada ao laboratório para efetuarem as amostras do solo.

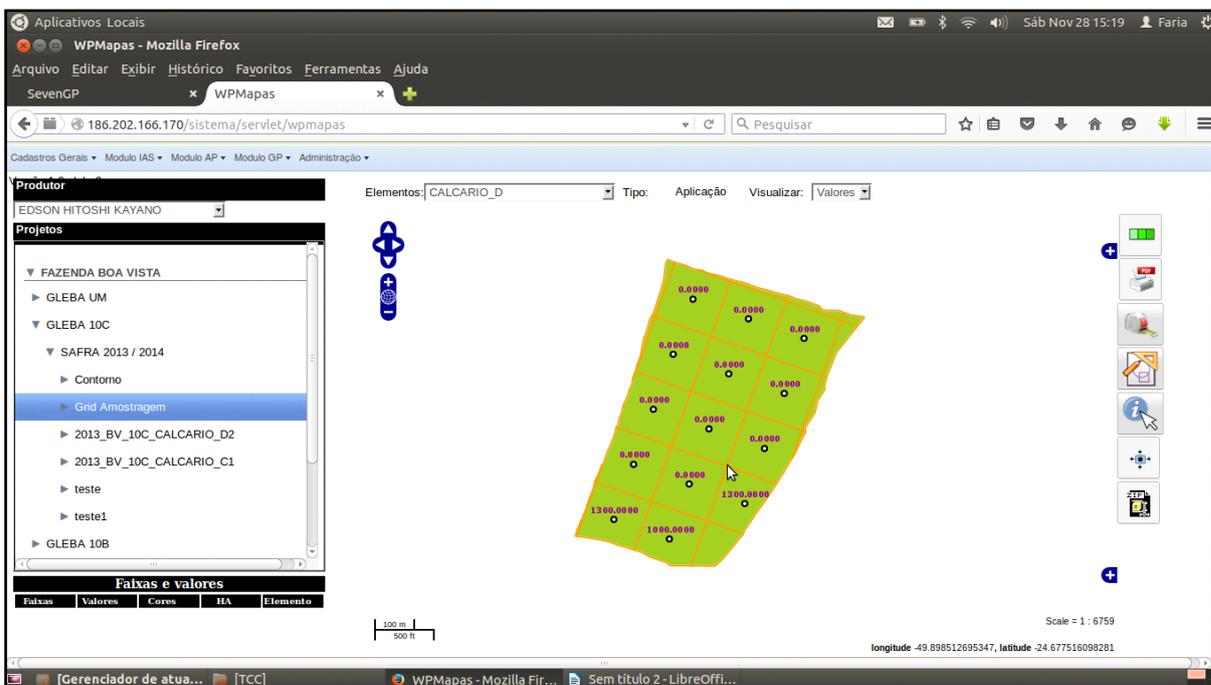


Figura 22: Mapa de amostra
Fonte: Autoria Própria

A Figura 23exibeum mapa de aplicação, no topo do GeoPublicador podemos ver qual elemento vai ser aplicado no solo, nesse caso o calcário, que também o “projeto” foi salvo com esse nome para facilitar sua localização.

Na Figura 23 observa-se como em todas as outras que o GeoPublicador mostrou os mapas selecionados, no lado direito e fica uma barra de ferramentas, que entre outras funcionalidades temos a impressão do mapa selecionado e o download de “*shapefile*” que será colocado no equipamento para aplicação.

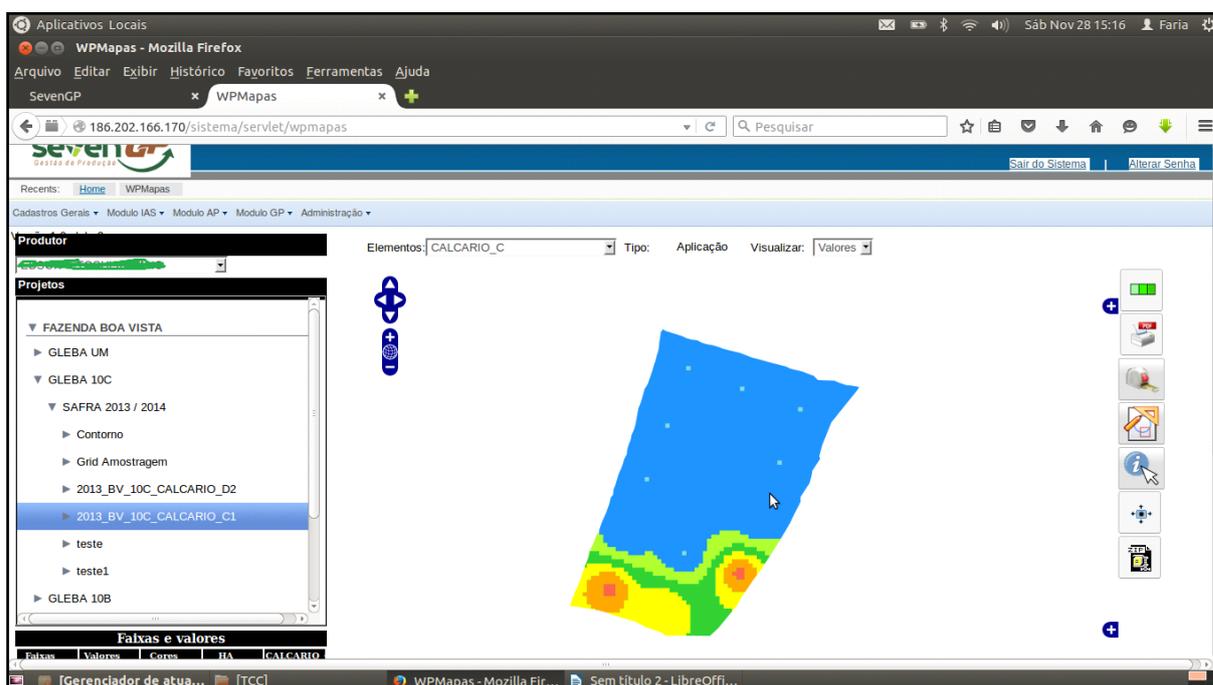


Figura 23: Mapa de aplicação
Fonte: Autoria Própria

A Figura 24 mostra um mapa de aplicação, e na parte inferior do GeoPublicador podemos ver as faixas de valores mostradas como legendas do mapa.

A cor mostrada na legenda é relacionada com a área no mapa onde vai ser aplicado, e a quantidade do elemento, esses valores vem do arquivo importado para o GeoPublicador.

A Figura 25 mostra um mapa de aplicação em PDF, que poderá ser impresso, do lado direito da tela onde é mostrado o PDF, uma barra de rolagem onde pode ser movido o mapa para visualização.

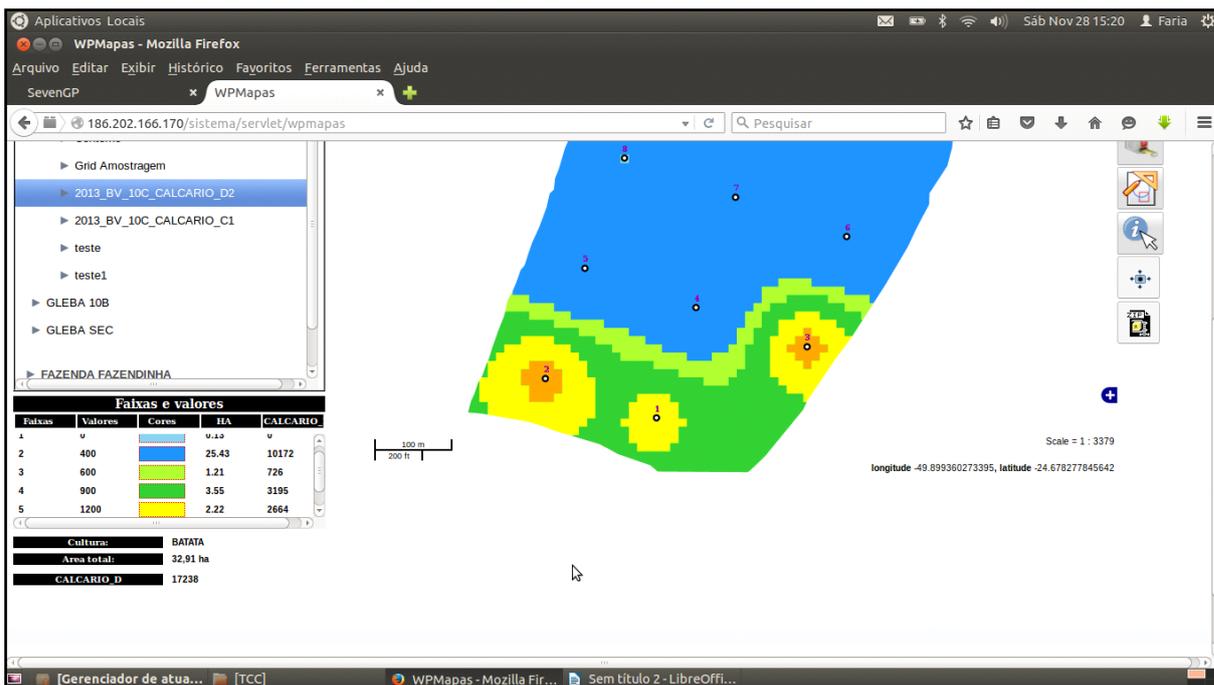


Figura 24: Mapa de aplicação com legendas
Fonte: Autoria Própria

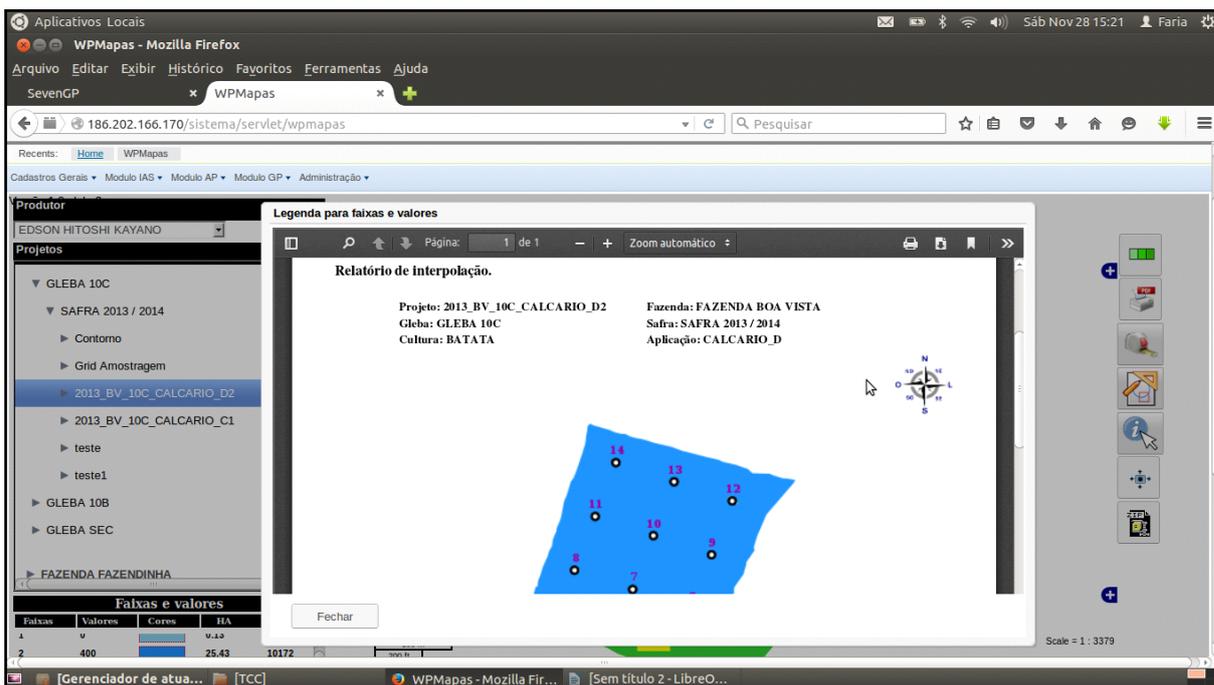


Figura 25: Mapa em PDF
Fonte: Autoria Própria

A Figura 26 mostra a chamada de *download* do arquivo *shapefile* para ser baixado e colocado no equipamento para aplicação do elemento e suas quantidades calculadas.

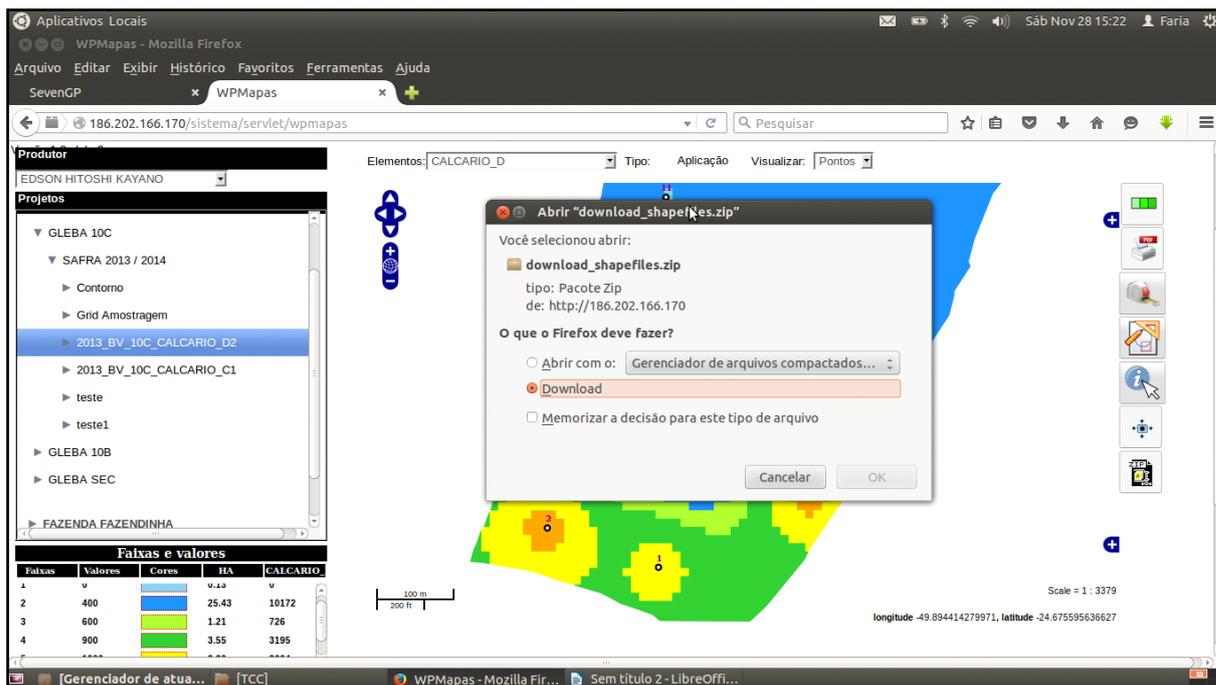


Figura 26: Download de mapa para aplicação em equipamento
Fonte: Autoria Própria

6. PESQUISA DE SATISFAÇÃO

Para avaliar superficialmente a satisfação dos usuários do geopublicador, foi realizada uma pesquisa para verificar se o sistema atendia ao seu propósito, de permitir visualizar e baixar mapas em tempo real, em qualquer lugar ou, se tornava as tarefas de seus usuários mais complexas que os sistemas similares apresentados no capítulo 3 deste trabalho.

Para tanto foram feitas as seguintes perguntas:

- Qual aplicativo para gerar mapas de amostragem e aplicação utilizava?
- Qual a principal dificuldade encontrada nesse aplicativo?
- As áreas em que trabalha possuem acesso a *Web* ?
- O GeoPublicador resolveu o problema na distribuição dos mapas de coleta e aplicação?
- Qual outra funcionalidade que pode ser destacada ?
- O que achou de poder armazenar todos os “projetos”, por safra, no próprio aplicativo?

Esta pesquisa foi respondida por dois agrônomos da cidade de Castro-PR, responsáveis por algumas fazendas na região. A amostra da pesquisa é pequena visto que focou-se apenas na região dos Campos Gerais, embora o sistema esteja em execução em outros pontos do Brasil.

Como todas as perguntas geram respostas subjetivas, tenta-se aqui expor de forma resumida que ambos os agrônomos consideraram o geopublicador eficaz no seu trabalho, permitindo gerar mapas e aplicar quase que de forma instântanea no equipamento presente na fazenda.

Também foi destacada a possibilidade de ajustar faixas de valores e a Visualização em tempo real.

7. CONCLUSÃO

Observou-se com a realização deste trabalho que a geotecnologia ou tecnologias espaciais e/ou geoprocessamento segundo alguns autores, são um conjunto de técnicas que atuam juntas para fornecer informações georeferenciadas e o GeoPublicador veio para agregar a essas geotecnologias.

A escolha do *Postgres* no início do projeto por ser um banco de dados robusto e *OpenSource* foi fundamental, pois sua extensão espacial o *PostGis* é de fácil instalação e possui muita informação para os desenvolvedores iniciantes.

Assim como o *OpenLayers* porque possui muita documentação, posts em português, vídeos, apostilas e até cursos online que ajudaram no estudo e desenvolvimento do projeto.

O GeoServer apesar de ter sido a parte mais complicada da implementação, foi uma escolha acertada, pois a integração com o *postgis* foi facilmente implementada, ele conta com a utilização de um arquivo xml que não trás somente informações geoespaciais facilitando assim o envio e recepção de outras informações.

O GTW (*Google Web Toolkit*) tornou a visualização mais eficiente com um conjunto de ferramentas que transformou o GeoPublicador numa ferramenta funcional.

A utilização desses *softwares* e APIs realmente facilitou o desenvolvimento do visualizador de mapas, e o resultado final já está sendo utilizado por alguns parceiros que ajudaram a financiar e tornar esse projeto uma realidade.

Sem dúvida a facilidade de visualizar um mapa que o agrônomo acabou de finalizar no seu escritório, e o funcionário que irá aplicar baixar esse projeto para colocar no equipamento e começa a aplicar o produto recomendado é o ponto forte desse projeto.

O Brasil é o país do agronegócio, o aumento populacional e por consequência a demanda por alimentos nos leva a previsão de que o Brasil alcançará a liderança mundial no fornecimento de alimentos e *commodities* que estão diretamente interligadas com o agronegócio.

A maioria das áreas cultiváveis já estão sendo utilizadas e não temos como aumentar essa área, então o conceito de agricultura de precisão ganha força e foco,

e softwares que ajudam nesse processo ganharam destaque e ajudaram o Brasil a alcançar essa liderança.

7.1. TRABALHOS FUTUROS

Durante o desenvolvimento deste trabalho observou-se algumas características e funcionalidades que poderiam estar presentes em uma nova versão do geopublicador e que podem se tornar novos trabalhos.

Tais características e funcionalidades são:

- Desenvolvimento de uma interface responsiva para o geopublicador, visto a crescente demanda por aparelhos como *smartphones*, *tablets* e *tablets* onde a mobilidade tornou-se uma necessidade.
- Manipulação e aumento do tamanho dos *grids*
- Alteração dos centróides utilizados no georreferenciamento

REFERÊNCIAS

BERNERS-LEE, Tim. **Information management: A proposal**. W3C archive, 1990. Disponível em: <<http://www.nic.funet.fi/index/FUNET/history/internet/w3c/proposal.html>>. Acesso em: 10 out. 2015.

CAKEPHP. **Entendendo o Model View Controller**. 2015. Disponível em: <<http://book.cakephp.org/2.0/pt/cakephp-overview/understanding-model-view-controller.html>>. Acesso em: 10 out. 2015.

DOERGE, T. A. **Management zone concepts. Site-Specific management guidelines**. 2000. Disponível em: <<http://www.farmresearch.com/ssmg/ssmg-02/002-print.ftm>>. Acesso em: 12 out. 2015.

FRAISSE, C. **Agricultura de Precisão: a tecnologia de GIS/GPS chega às fazendas**. *Revista Fator GIS*, Curitiba, n. 21, p. 28-33, 1998.

FRATERNALI, Piero; COMAI, Sara; BOZZON, Alessandro; CARUGHI, Giovanni T. **Engineering rich internet applications with a model-driven approach**. *ACM Trans. Web*, v. 4, n. 2, p. 1–47, bBbaixado, 2010. Disponível em: <<http://portal.acm.org/citation.cfm?id=1734200.1734204&coll=Portal&dl=GUIDE&CFID=89708622&CFTOKEN=40265052>>. Acesso em: 13 out. 2015.

GEOAGRI. **Farm Works Office / Trimble**. 2012. Disponível em: <<http://www.geoagri.com.br/Produtos.aspx?idSubCategoria=f3fd84e1-efc8-4b7c-b712-ab18ae0f74a8>>. Acesso em: 21 out. 2015.

GEOSERVER. **REST**. 2009. Disponível em: <<http://docs.geoserver.org/2.0.0/user/extensions/rest/index.html>>. Acesso em: 21 out. 2015.

JUNIOR, Paulo F. **Iniciando com o GWT**. 2009. Disponível em: <<http://javafree.uol.com.br/artigo/874221/Iniciando-com-o-GWT.html>>. Acesso em: 22 out. 2015.

LINAJE, Marino; PRECIADO, Juan C.; SANCHEZ-FIGUEROA, Fernando. **Domain-specific model for designing rich internet application user interfaces**. In: *Computer-Aided Design of User Interfaces VI*, Springer London, bBbaixado2 (pxx-linaje-f.pdf), 2009, p. 295–306. Disponível em: <<http://www.springerlink.com/content/k611w3t86k730682/>>. Acesso em: 10 out. 2015.

MAHEMOFF, Michael. **Ajax design patterns**. O'Reilly Media, Inc., 2006.

MARTINS, E. **Servidor de Aplicação**. 2015. Disponível em: <<https://sites.google.com/site/jvmtotal/home/conceitos/servidor-de-aplicacao>>. Acesso em: 20 out. 2015.

MEDEIROS, Anderson. **Agricultura Familiar e Softwares Livres de Geoprocessamento**. 2015. Disponível em: <<http://andersonmedeiros.com/proposta-metodologica-agricultura-familiar-geotecnologias/>>. Acesso em: 20 out. 2015.

MUNSON, Ethan V.; PIMENTEL, Maria G. **Specialized documents**. In: Web Accessibility, Springer London, 2008, p. 274–285 (Human-Computer Interaction Series, v.4). Disponível em: <<http://www.springerlink.com/content/x226q248w51j6qg0/>>. Acesso em: 20 out. 2015.

NOUIRA, L. **JavaScript – Introdução a Linguagem JavaScript**. 2015. Disponível em: <<http://br.ccm.net/faq/2680-javascript-introducao-a-linguagem-javascript>>. Acesso em: 21 out. 2015.

OPENLAYERS. **Open Layers 3 API Documentation**. 2014. Disponível em: <<http://openlayers.org/en/v3.0.0/apidoc/>>. Acesso em: 21 out. 2015.

OPENLAYERS. **Creating a Map**. 2015. Disponível em: <<http://openlayers.org/workshop/basics/map.html>>. Acesso em: 15 out. 2015.

SCHMIDT, Kay-Uwe; DÖRFLINGER, Jörg; RAHMANI, Tirdad; SAHBI, Mehdi; STOJANOVIC, Ljiljana; THOMAS, Susan M. **An user interface adaptation architecture for rich internet applications**. In: ESWC'08: Proceedings of the 5th European semantic web conference on The semantic web, Berlin, Heidelberg: Springer-Verlag, 2008, p. 736–750.

SCHRÖDER, E. P. **Tecnologias inovadoras em aviação agrícola**. In: GUEDES, J. V. C.; DORNELLES, S. H. B. **Tecnologia e segurança na aplicação de agrotóxicos**. Santa Maria: 1998. p. 87-93.

SLIDESHARE. **Introdução ao OpenLayers**. 2009a. Disponível em: <<http://pt.slideshare.net/fernandoquadro/introducao-ao-openlayers>>. Acesso em: 19 out. 2015.

SLIDESHARE. **Metadados: Histórico, Conceitos, Aplicações e Formatos**. 2009b. Disponível em: <<http://pt.slideshare.net/lulinha/metadados>>. Acesso em: 21 out. 2015.

VELASCO, Carlos A.; DENEV, Dimitar; STEGEMANN, Dirk; MOHAMAD, Yehya. **A web compliance engineering framework to support the development of accessible rich internet applications**. In: W4A '08: Proceedings of the 2008 international cross-disciplinary conference on Web accessibility (W4A), New York, NY, USA: ACM, 2008, p. 45–49.

WATANABE, William M. **Avaliação automática para aplicações ricas na web: um apoio aos testes de aceitação**. 2012. 62 f. Qualificação. Universidade de São Paulo. São Carlos, 2012.

TECNICAS INTELIGENTES DE PROGRAMAÇÃO . Disponível em: ftp://ftp.inf.puc-rio.br/pub/docs/techreports/07_04_gazola.pdf

BURROUGH, P.A. Principles of geographical information systems for land resources

assessment. Oxford, Clarendon Press, 1986. 193 p. BURROUGH, P. A.; MCDONNELL, R.A. Principles of geographical information systems. Oxford, Oxford University Press, 1998.