



RODOLFO LUIS MARCOS

**ESTÁGIO EM EMPRESA PARA
DESENVOLVIMENTO DE FERRAMENTAS E
ALGORITMOS PARA ANÁLISES
GEOESPACIAIS.**

LAVRAS –MG

2014

RODOLFO LUIS MARCOS

**ESTÁGIO EM EMPRESA PARA DESENVOLVIMENTO DE
FERRAMENTAS E ALGORITMOS PARA ANÁLISES GEOESPACIAIS**

Relatório final de Estágio Supervisionado
apresentado 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. André Pimenta
Freire

Co-Orientador

Prof. Me. Fausto Weimar
Acerbi Júnior.

LAVRAS

MINAS GERAIS – BRASIL

2014

RODOLFO LUIS MARCOS

**ESTÁGIO EM EMPRESA PARA
DESENVOLVIMENTO DE FERRAMENTAS E
ALGORITMOS PARA ANÁLISES
GEOESPACIAIS.**

Trabalho de Conclusão de Curso de
Graduação apresentado ao Colegiado do
Curso de Bacharelado em Ciência da
Computação, para obtenção do título de
Bacharel.

APROVADA em 21 de novembro de 2014.

Dr. Luiz Henrique Andrade Correia

Dr. Raphael Winckler de Bettio



Dr. André Pimenta Freire (Orientador)

Me. Fausto Weimar Acerbi Júnior (Coorientador)

**LAVRAS-MG
Novembro/2014**

Agradeço este trabalho à todas as pessoas que direta ou indiretamente estiveram comigo nestes quatro anos. Pois estes quatro anos foram sucessivos dias onde o carinho, suporte e paciência de quem conviveu comigo foram decisivos para o sucesso desta jornada. À Deus como maestro desta obra de vida.

Aos meus Pais; Ana Maria e José Marcos, bem como aos meus irmãos Fernando e Josiane e familiares que entenderam minhas escolhas e sempre me apoiaram nas minhas dificuldades, sobre todas as coisas.

À minha namorada Ivy Mayara, pelo carinho, paciência, e dedicação que trouxeram um significado especial nesta jornada.

Aos meus orientadores Fausto e André, assim como o pessoal do LEMAF, que acreditaram em mim e proporcionaram momentos de aprendizagem e companheirismo.

“Vós sois o sal da terra. Vós sois a luz do mundo.”

Jesus Cristo.

RESUMO

O Geoprocessamento surgiu como área de conhecimento que analisa e quantifica aspectos da superfície terrestre. Utilizando suas técnicas é possível definir elementos e obter diversas informações das mais variadas finalidades dos objetos da superfície. As ferramentas computacionais se destacam como base para os processos do geoprocessamento, criando várias possibilidades de utilização de seus recursos.

O trabalho realizado durante o estágio consistiu em desenvolver algoritmos e ferramentas computacionais através da integração com Sistemas de Informações Geográficas a fim de automatizar e facilitar a extração de informações de imagens providas de sensores a bordo de satélites.

Palavras-chave: Geoprocessamento, Análise Geoespacial, Extração de Informações, Automatização

SUMÁRIO

1. INTRODUÇÃO	1
2. ESTRUTURA DO DOCUMENTO.....	2
3. DESCRIÇÃO DO LOCAL DO ESTÁGIO: LEMAF.	3
4. ATIVIDADES DESENVOLVIDAS DURANTE O ESTÁGIO	5
4.1 DADOS GEOGRÁFICOS E GEOPROCESSAMENTO	6
4.2 REPRESENTAÇÃO DE DADOS GEOGRÁFICOS	7
4.2.1 REPRESENTAÇÃO VETORIAL OU <i>SHAPE</i>	7
4.2.1 REPRESENTAÇÃO MATRICIAL OU <i>RASTER</i>	8
4.2.2 GEOPROCESSAMENTO	9
4.2.3.1 ARCGIS 10	
5. DESCRIÇÃO DAS ATIVIDADES DESENVOLVIDAS.	12
5.1 CONSTRUÇÃO DE UMA PLATAFORMA PARA ORGANIZAÇÃO DE <i>SHAPES</i> E PROCESSAMENTO INTEGRADO.	12
5.1.1 CONCEITOS ENVOLVIDOS NO DESENVOLVIMENTO DA PLATAFORMA	12
5.1.2 ELABORAÇÃO DO BANCO DE DADOS DA PLATAFORMA.....	13
5.1.2.1 TECNOLOGIAS NO DESENVOLVIMENTO DA PLATAFORMA...	15
5.1.2.2 DIFICULDADES NO DESENVOLVIMENTO DA PLATAFORMA .	15
5.1.3 CONSTRUÇÃO DO WEBSITE DA PLATAFORMA.....	16
5.1.3.1 TECNOLOGIAS NO DESENVOLVIMENTO DO WEBSITE	18
5.1.3.2 DIFICULDADES NO DESENVOLVIMENTO DO WEBSITE	19
5.1.4 REFERENCIAL	19
5.1.4.1 BANCOS DE DADOS	19

5.1.4.2 USABILIDADE.....	21
5.2 <i>SCRIPT</i> PARA ANÁLISE DE ÁREAS DE PRESERVAÇÃO PERMANENTE – APP.	23
5.2.1 CONCEITOS ENVOLVIDOS	23
5.2.2 ÁREA DE APP	24
5.2.3 OBTENDO A ÁREA DE APP	24
5.2.4 CLASSIFICAÇÃO DE IMAGENS.....	26
5.2.5 FUNCIONAMENTO DO <i>SCRIPT</i> DE APP	27
5.2.6 TECNOLOGIAS NO DESENVOLVIMENTO DO <i>SCRIPT</i> DE APP.....	29
5.2.7 DIFICULDADES NO DESENVOLVIMENTO DO <i>SCRIPT</i> DE APP....	30
5.2.8 REFERENCIAL	31
5.2.8.1 UNIFIED MODELLING LANGUAGE – UML.....	31
5.3 <i>SCRIPT</i> PARA POSICIONAMENTO DE TORRES.....	35
5.3.1 CONCEITOS ENVOLVIDOS	35
5.3.2 FUNCIONAMENTO DO <i>SCRIPT</i> PARA POSICIONAMENTO DE TORRES	37
5.3.3 TECNOLOGIAS UTILIZADAS NO <i>SCRIPT</i> PARA POSICIONAMENTO DE TORRES	39
5.3.4 DIFICULDADES ENCONTRADAS NO <i>SCRIPT</i> PARA POSICIONAMENTO DE TORRES	39
6 CONCLUSÕES	43
6.1 CONSIDERAÇÕES FINAIS	43
6.2 TRABALHOS FUTUROS.	44
REFERÊNCIAS BIBLIOGRÁFICAS.....	45
APÊNDICE A – Interfaces da plataforma de organização de shapes.....	47

APÊNDICE B – Trechos de código.....	50
-------------------------------------	----

LISTA DE FIGURAS

Figura 1 Pontos, linhas e polígonos em uma representação vetorial.	8
Figura 2 <i>Raster</i> de um modelo de elevação.	9
Figura 3 Mapa gerado pelo Software ArcGis.	10
Figura 4 Diagrama entidade-relacionamento do Banco de Dados da plataforma de armazenamento de shapes e processamento integrado.	14
Figura 5. Diagrama de caso de uso.	17
Figura 6 Áreas de preservação permanente em verde.	24
Figura 7 Etapas para a obtenção da APP.	25
Figura 9 Fotografia de satélite.	26
Figura 8 Imagem Classificada.	26
Figura 10. Interface do Script de Análise de APP em versão funcional.	28
Figura 11. Relatório do algoritmo de Análise de APP.	29
Figura 12 Diagrama de atividades do algoritmo de APP.	31
Figura 13. <i>Shapes</i> das áreas de interesse e pontos de torres de visão.	36
Figura 14. <i>Viewshed</i> da torre em verde. As áreas rosadas são visíveis para a torre.	36
Figura 15. Distribuição de torres (pontos verdes) sobre um modelo de elevação.	38

LISTA DE TABELAS

Tabela 1. Elementos básicos do Diagrama de Atividades.	34
Tabela 2. Resultado do Algoritmo de Combinação de Torres.	39
Tabela 3. Número de combinações em função do número de torres n e agrupamento s	40
Tabela 4. Torres e seus rótulos.	41
Tabela 5. Número de combinações utilizando a estratégia de agrupamento.	41

LISTA DE EQUAÇÕES

Equação 1. Cálculo de combinações sem ordem e sem repetição de n elementos tomados em s	37
--	----

1. INTRODUÇÃO

Este trabalho descreve as tarefas realizadas durante o estágio em Tecnologia da Informação na área de geoprocessamento no Laboratório de estudo e projetos em Manejo Florestal - LEMAF, localizado no Campus da Universidade Federal de Lavras. A maioria dos produtos e processos desenvolvidos no local são providos e dependentes em grande parte de Sistemas de Informação, sendo, portanto a área de Tecnologia da Informação de extrema importância e atuação no LEMAF.

O trabalho realizado consiste na automatização de processos utilizados no geoprocessamento (área da ciência relacionada à extração de informações de dados geográficos). As atividades desenvolvidas foram desde a elicitação de requisitos até a implantação destes sistemas de automatização, atuando em todas as áreas do ciclo de desenvolvimento de software, com foco na implementação.

Os trabalhos realizados em geoprocessamento no laboratório, até o momento, consistiram em realizar uma sucessão de análises e transformações de imagens provindas de satélite, a fim de que sejam gerados mapas e relatórios contendo informações de interesse. O trabalho despendido desde a obtenção da imagem até os produtos específicos é grande, e muitas vezes consiste em tarefas repetitivas que podem se tornar automatizadas. O campo de geoprocessamento é vasto, quantifica e estuda objetos de diversas naturezas da superfície terrestre, gerando informações através da interpretação e análise das imagens da superfície do planeta.

2. ESTRUTURA DO DOCUMENTO

Na Seção 3, é descrito o local onde foi realizado o estágio, suas atividades e organização. Na Seção 4 são abordadas brevemente as tarefas desenvolvidas durante todo o estágio, contendo para cada atividade o número de horas despendidas, suas etapas e cronograma. A Seção 5 detalha as atividades apresentadas na Seção 4, descrevendo para cada atividade: visão geral, tecnologias, dificuldades, conceitos que auxiliaram na realização da tarefa e referencial. A Seção 6 traz a conclusão deste relatório de estágio, observando aprendizados extraídos, bem como observações e trabalhos futuros. O documento é finalizando com o referencial bibliográfico e apêndices.

3. DESCRIÇÃO DO LOCAL DO ESTÁGIO: LEMAF.

Fundando em Abril de 2004, o Laboratório de Estudos e Projetos em Manejo Florestal (LEMAF) pertencente ao Departamento de Ciências Florestais da Universidade Federal de Lavras objetiva principalmente realizar pesquisa, ensino e extensão nas áreas de Manejo e Inventário Florestais. Para que tal objetivo seja atendido é necessário conhecimentos sobre Ciências Florestais e ferramentas e técnicas que possibilitem a extração de informação e conhecimento. A tecnologia da informação atua como fonte das ferramentas necessárias para a realização dos trabalhos realizados, uma vez que provê meios para análise de imagens espaciais.

A coexistência da Tecnologia de informação dentro do laboratório se consolidou depois do sucesso do projeto “Inventário Florestal de Minas Gerais”, trabalho realizado para a secretaria do meio ambiente do Estado de Minas Gerais, onde sistemas de informações foram criados para auxiliar no desenvolvimento e divulgação dos dados do inventário.

O LEMAF conta atualmente com aproximadamente 85 funcionários, sendo que a tecnologia da Informação é distribuída nas equipes de Qualidade de software, desenvolvimento Java/Flex, Banco de dados e Geoprocessamento, dentre outras. Os sistemas desenvolvidos são grandes aliados à proteção ambiental, principalmente o Sistema Nacional de Cadastro Ambiental Rural (CAR), plataforma onde são cadastradas e analisadas áreas rurais de todo Brasil com o objetivo de verificar boas práticas ao código florestal e consequentemente a preservação de fauna e flora. O mercado alvo dos sistemas desenvolvidos é principalmente formado por entidades governamentais, como por exemplo os governos dos estados de Minas Gerais, Acre e Mato Grosso do Sul.

Por se tratar de um laboratório em uma instituição de ensino, o LEMAF não conta com um organograma. No entanto o coordenador de T.I é o principal

responsável no desenvolvimento dos sistemas do local, enquanto que as outras equipes de desenvolvimento possuem o mesmo nível hierárquico. A metodologia de desenvolvimento é o *Scrum*, que é uma metodologia de desenvolvimento ágil.

Para se desenvolver os sistemas, é gerada uma documentação prévia contendo as funcionalidades do mesmo, depois são gerados protótipos para validação. O co-orientador Fausto foi o idealizador dos sistemas desenvolvidos. Os colegas de trabalho Diego Sarmiento (desenvolvedor) e Cristiane Faleiros (Banco de Dados) auxiliaram nos sistemas desenvolvidos, além do Inácio Bueno que ajudou no entendimento dos processos do geoprocessamento.

4. ATIVIDADES DESENVOLVIDAS DURANTE O ESTÁGIO

A seguir, serão listadas as tarefas desenvolvidas durante o estágio, suas etapas e duração em ordem cronológica. Na próxima seção cada tarefa será detalhada e os conceitos computacionais desenvolvidos.

- A. Construção de uma plataforma para organização de *shapes* e processamento integrado.
 - Elaboração do banco de dados (40H - Agosto).
 - Construção do *website* (120H – Agosto, Setembro).

- B. Desenvolvimento de *Script* para Análise de Áreas de Preservação Permanente (APP).
 - Modelagem/Análise (10H - Agosto).
 - Construção do *Script* / Interface (40H – Agosto, Setembro).

- C. Desenvolvimento de *Script* para otimização em posicionamento de torres de visada.
 - Construção do *Script* (35H – Setembro, Outubro).
 - Análise de complexidade / Alterações (30H – Setembro, Outubro).

Para uma clara compreensão do conteúdo deste trabalho de conclusão de curso, serão descritos a seguir alguns conceitos importantes que dizem respeito a dados geográficos, geoprocessamento e sistemas de informação geográficos.

4.1 DADOS GEOGRÁFICOS E GEOPROCESSAMENTO

Um dado geográfico é a representação de um fenômeno ou objeto atrelado a sua ocorrência espacial (Ex. fábricas com suas respectivas coordenadas na superfície terrestre, mapa de ocorrência de determinada doença).

Os dados geográficos ou georreferenciados são dados que possuem uma dimensão espacial referente à sua localização na superfície da terra, num determinado instante ou período de tempo (CÂMARA, G; 1995). Dados geográficos são por exemplo: estradas, ferrovias ou uma imagem de satélite onde cada pixel está associado a um par de coordenadas.

Os dados geográficos possuem três características fundamentais: espaciais, não espaciais e temporais (DAGERMOND, 1990; MEDEIROS, 1994; LAURINI, 1992). As características espaciais definem o local de ocorrência do dado e sua geometria. As características temporais descrevem sobre o tempo de ocorrência e suas variações. As características não espaciais descrevem o fenômeno. Uma representação espacial de um dado geográfico é a descrição de sua forma geométrica associada à sua posição geográfica (BORGES, 1997).

Os dados georreferenciados possuem duas representações principais; a representação matricial e a representação vetorial que serão discutidas em breve. No que se refere à extração de informação, quando dados brutos são organizados e/ou relacionados em uma base de dados, é possível, através de técnicas computacionais como a mineração de dados, a obtenção de informações. Com dados geográficos também é possível a extração de informações, com a diferença que estes possuem a localização geográfica de sua ocorrência, sendo possível processá-los para se obter os seguintes tipos de informações:

- O centro de Lavras possui quantas casas?
- Existe algum cruzamento entre estradas e ferrovias no sul de Minas?

O potencial de um banco de dados geográficos se verifica quando esses dados são organizados e/ou relacionados produzindo informações de natureza geográfica, que podem ser plotadas em um gráfico ou mapa.

4.2 REPRESENTAÇÃO DE DADOS GEOGRÁFICOS

Os dados geográficos são representados principalmente por duas abordagens; a matricial e a vetorial.

4.2.1 REPRESENTAÇÃO VETORIAL OU *SHAPE*

A representação em formato vetorial define os dados georreferenciados através de pontos, linhas e polígonos e não como células em uma matriz como a representação matricial. Os pontos são representados por um par de coordenadas, enquanto que uma linha é representada por uma sequência de pares de coordenadas. O polígono é simplesmente uma sequência de linhas onde há um ciclo formado por coincidir o primeiro e último ponto destas linhas. Ruas, territórios políticos e paradas de ônibus podem ser respectivamente representados por linhas, polígonos e pontos.

Um arquivo que contém uma representação vetorial é comumente chamado de *shapefile* ou apenas *shape*. O armazenamento de um *shapefile* é mais econômico em termos de espaço do que um arquivo contendo uma representação matricial (*Raster*). Na Figura 1 é possível verificar que *Well*, *River* e *Lake* são respectivamente representados por pontos, linhas e polígonos.

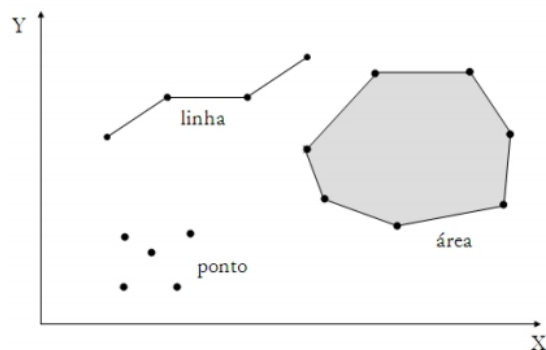


Figura 1 Pontos, linhas e polígonos em uma representação vetorial.

Fonte: <http://www.dpi.inpe.br/livros/bdados/cap1.pdf>

4.2.1 REPRESENTAÇÃO MATRICIAL OU *RASTER*

A representação em formato matricial, também chamada de *raster*, é uma matriz onde cada célula possui um conjunto de valores representando as características geográficas do local (BOTELHO, 1995). Um exemplo seria uma matriz que representa um modelo de elevação de uma determinada área. Células com valores mais altos representam uma altitude mais alta, enquanto que células com valores mais baixos representam baixa altitude. Sendo assim uma imagem em escala de cinza pode representar as diferentes altitudes em uma área, sendo portanto um modelo de elevação. Na Figura 2 é possível observar um *raster* que representa um modelo de elevação.

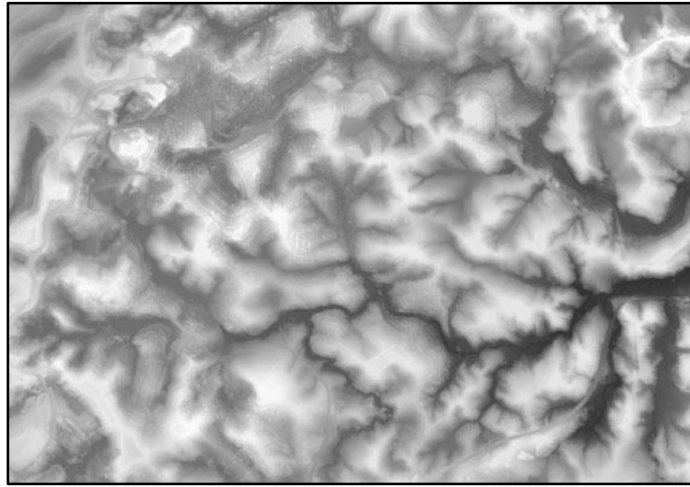


Figura 2 *Raster* de um modelo de elevação.

Fonte: <http://mundogeo.com/wp-content/uploads/2000/português/infogeo/55/pag38-2.jpg>

4.2.2 GEOPROCESSAMENTO

O geoprocessamento, disciplina do conhecimento que utiliza técnicas matemáticas e computacionais para o tratamento da informação geográfica, visa produzir informações a partir de dados geográficos (CÂMARA et al, 2001). Para que seus objetivos sejam cumpridos, são necessárias ferramentas computacionais, denominadas Sistemas de Informação Geográfica (SIG), estas ferramentas permitem a integração de vários dados georreferenciados em um único Banco de dados. E provê suporte para a extração de informações. Um GIS bastante conhecido, e que é utilizado neste trabalho, é o software ArcGis.

4.2.3.1 ARCGIS

O Software ArcGis, produto da ESRI (ArcGis, 2014), é capaz de realizar inúmeras atividades na extração de informações geográficas. O ArcGis Desktop é o principal aplicativo do conjunto ArcGis, contando com uma interface gráfica, onde os operadores manipulam dados georreferenciados como *rasters* e *shapes*.

No presente trabalho utilizou-se as funções do ArcGis para a programação, através da biblioteca ArcPy (ArcGis Resources, 2013).



Figura 3 Mapa gerado pelo Software ArcGis.

Fonte: Arquivo pessoal.

Por meio do software é possível gerar *layouts* de mapas prontos para o consumo dos usuários (Figura 3), contendo de fato as informações relevantes sobre a área de estudo. O mapa sumariza e demonstra de forma intuitiva os atributos de interesse. O software ArcGis conta com uma biblioteca de funções

chamada ArcPy, que oferece as funções internas do software, as mesmas disponíveis através da interface do usuário, porém com maior possibilidade de flexibilidade devido ao diferente uso de seus parâmetros. O ArcPy é disponibilizado para uso integrado com Python, fato este que justifica a utilização da Linguagem Python durante o trabalho.

5. DESCRIÇÃO DAS ATIVIDADES DESENVOLVIDAS.

5.1 CONSTRUÇÃO DE UMA PLATAFORMA PARA ORGANIZAÇÃO DE *SHAPES* E PROCESSAMENTO INTEGRADO.

Uma necessidade encontrada no local de trabalho foi a organização dos *shapes* usados diariamente e que são alvos de processamentos posteriores. Portanto, foi desenvolvida uma plataforma para a organização destes *shapes*, que consiste em um Banco de Dados e um *website*. Essa plataforma, além de organizar *shapes*, servirá futuramente como acesso para os algoritmos de automatização, que encontrarão em um único local os dados de que precisam. Para isto, serão estudadas e implantadas tecnologias de integração entre Python e PHP, para que o próprio *Website* da plataforma sirva como interface de acesso para os algoritmos de automatização. Além do acesso, a integração tornará possível a extração de dados esporadicamente, possibilitando a criação de um monitoramento de regiões geográficas de forma automática. Os algoritmos para análise de APP, e posicionamento de torres de visão que ainda serão discutidos são sistemas de automatização que serão integrados.

5.1.1 CONCEITOS ENVOLVIDOS NO DESENVOLVIMENTO DA PLATAFORMA

Para o desenvolvimento da plataforma foram utilizados os seguintes conceitos aprendidos durante o curso de Ciência da Computação.

- Diagrama Entidade Relacionamento (DER).
- Banco de Dados e Normalização de Banco de Dados Relacionais.

- Desenvolvimento Web.
- Administração de Servidores / Redes.
- Prototipagem.
- Usabilidade.

O desenvolvimento da plataforma de *shapes* foi dividido em duas etapas: Elaboração do banco de dados e o desenvolvimento do *website*.

5.1.2 ELABORAÇÃO DO BANCO DE DADOS DA PLATAFORMA

Para a elaboração do Banco de Dados da plataforma para armazenar arquivos de *shapes*, ou seja, os polígonos, linhas e pontos, que delimitam uma área espacial, foi primordial saber qual a natureza de pesquisa seria realizada sobre esse banco. Uma vez definidos os objetivos e as buscas esperadas foram determinados os atributos armazenados para cada *shape* no Banco de Dados. Os atributos são:

- Identificação única do *shape*.
- Id da pessoa que submeteu o *shape*.
- Data de submissão na forma mês-ano.
- Cena do satélite a qual pertence.
- Tipo de arquivo.

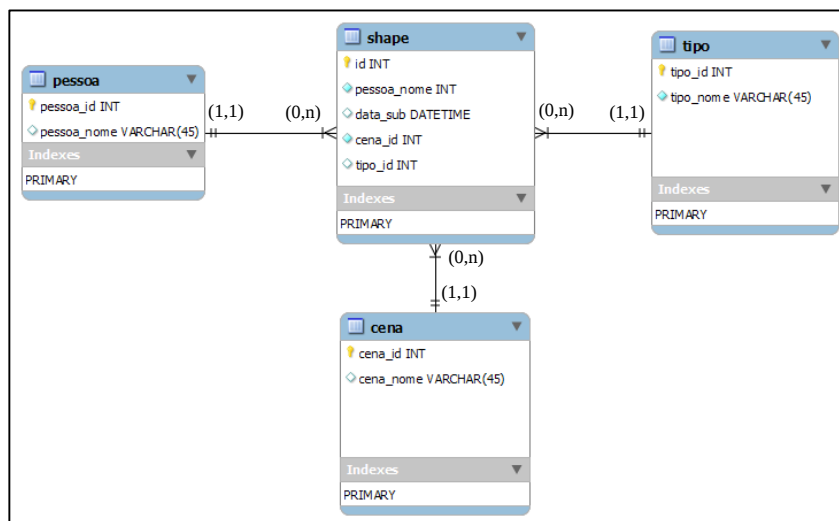


Figura 4 Diagrama entidade-relacionamento do Banco de Dados da plataforma de armazenamento de shapes e processamento integrado.

Fonte: Arquivo Pessoal

Desta forma, é possível recuperar arquivos a partir do nome da pessoa que o submeteu, data de submissão, cena e tipo. Para prover maior desempenho, confiabilidade, e eliminar redundâncias o Banco de Dados foi normalizado de acordo com a terceira forma normal – 3FN (KORTH et al, 2006; HEUSER, 2008). Para que um Banco de Dados esteja na 3FN deve estar também na 1FN e 2FN obedecendo as seguintes premissas:

- Os valores da coluna de uma tabela não se repetem e são atômicos (1FN).
- Todos os atributos não chaves dependem unicamente da chave primária (Depende de toda a chave primária, e não parte dela) (2FN).
- Nenhum atributo não chave depende de outro atributo não chave (3FN).

5.1.2.1 TECNOLOGIAS NO DESENVOLVIMENTO DA PLATAFORMA

Para o desenvolvimento da plataforma, o Sistema Gerenciador de Banco de Dados (SGBD) escolhido foi o MySQL¹, e o ambiente de desenvolvimento foi o MySQL Workbench² devido a simplicidade e funcionalidade.

5.1.2.2 DIFICULDADES NO DESENVOLVIMENTO DA PLATAFORMA

A dificuldade encontrada foi manter a integridade referencial do Banco de Dados, ou seja, quando há relacionamento entre as tabelas é preciso mantê-las consistentes mesmo quando alterações e exclusões são efetuadas.

A integridade referencial deve garantir que não será armazenado nenhum *shape* sem uma cena existente correspondente. Para resolver este problema o *website* que acessa o Banco de Dados automaticamente mostra apenas cenas já cadastrados no momento de inserir um novo *shape*. Assim como faz o mesmo para pessoa e tipo, garantindo a associação no momento da inserção.

Caso uma tupla seja excluída, o Banco de Dados excluirá todas as tuplas que contém o registro como chave estrangeira, fazendo uma exclusão em cascata. Desta forma se uma cena é excluída todos os *shapes* daquela cena também serão excluídos. Para evitar transtornos, quando o usuário escolhe excluir uma cena é exibida uma confirmação notificando a perda dos *shapes*.

¹ <http://www.mysql.com/>

² <http://www.mysql.com/products/workbench/>

5.1.3 CONSTRUÇÃO DO WEBSITE DA PLATAFORMA

Após concluída a elaboração do Banco de Dados, foi necessário um meio de acesso e armazenamento desses dados. Para isso, foi construído um *website* preocupado com usabilidade que permite ao usuário realizar as ações do seu dia-a-dia de forma eficiente. No Apêndice A é possível visualizar as interfaces do *website* em seu protótipo funcional.

Para garantir a usabilidade, que é detalhada adiante, foram tomadas as seguintes medidas propostas por Schneiderman (1998) e que fazem parte das suas oito regras de ouro.

- Consistência: Para garantir a consistência foram mantidos termos para ações semelhantes (Criar Tipo, Cria Cena.)
- *FeedBack* informativo: Durante as ações do usuário é dado com frequência recomendações de como prosseguir para atingir um objetivo (Como buscar *shapes* de uma cena ou data específica).
- Prevenção e tratamento de erros: Informar e manipular dados de entrada do usuário garante que os parâmetros sejam processados corretamente. (Preenchimento automático e máscaras nos formulários de datas e tipos).

A modelagem do *website* foi feita através de prototipação, que consiste em elaborar uma forma de representação simplificada da página *web*, um esboço com seus formulários, botões e funções principais. A prototipação, assim como os diagramas de caso de uso, são atividades da engenharia de requisitos que guiam o desenvolvimento de *software*. Ao utilizar protótipos é possível refinar o *website* tomando como base o *feedback* dos *stakeholders*¹, os trabalhadores do laboratório, de forma a obter um produto mais condizente com a realidade de utilização do mesmo.

¹ No contexto de engenharia de software *stakeholders* são as partes interessadas em um projeto de software (clientes, desenvolvedores, gerentes de projeto).

As tarefas de armazenar um *shape* e realizar buscas pelos usuários no Banco de Dados de forma confiável e acessível são os objetivos identificados e que o website deve corresponder. O sistema deve contar com acesso restrito, através de login, além das seguintes funcionalidades:

- Cadastrar/Excluir Cena.
- Cadastrar/Excluir Usuário.
- Cadastrar/Excluir Tipo de *Shape*.
- Cadastrar/Excluir/Alterar *Shape*.

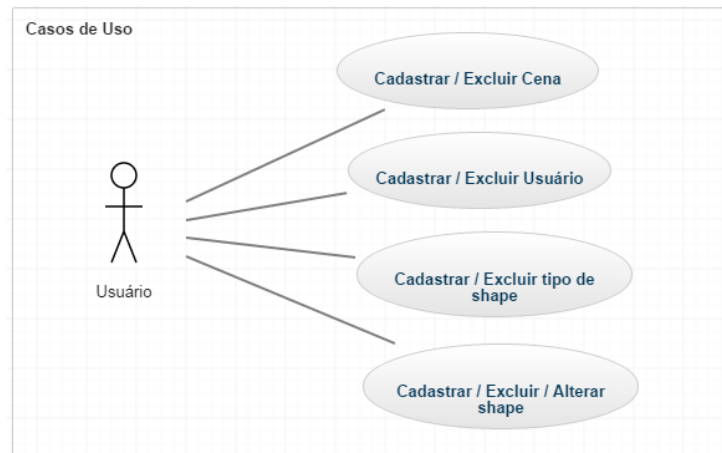


Figura 5. Diagrama de casos de uso.
Fonte: Arquivo Pessoal

5.1.3.1 TECNOLOGIAS NO DESENVOLVIMENTO DO WEBSITE

O *Website* foi desenvolvido em HTML (*HyperText Markup Language*)¹ que é uma linguagem de marcação para desenvolvimento de *websites*, PHP (*Hypertext Preprocessor*)² que é uma linguagem para gerar conteúdo dinâmico em páginas *web*. O software utilizado no desenvolvimento foi Adobe Dreamweaver CS6³, utilizando o framework Bootstrap⁴ que além de trazer um visual elegante à página também propicia recursos para usabilidade. As conexões com o Banco de Dados MySQL foram feitas em módulos PHP. A biblioteca jQuery⁵ foi utilizada para realizar máscaras na entradas de dados em formulários.

¹ <http://www.w3.org/MarkUp/>

² <http://php.net/>

³ <http://www.adobe.com/br/products/dreamweaver.html>

⁴ <http://getbootstrap.com/>

⁵ <http://jquery.com/>

5.1.3.2 DIFICULDADES NO DESENVOLVIMENTO DO WEBSITE

A dificuldade encontrada foi a alteração dos requisitos ao longo do desenvolvimento, embora o Banco de Dados tenha-se mantido o mesmo desde sua elaboração, o *website* foi reconstruído algumas vezes, para atender as seguintes modificações:

- Padronização de nomeação de *shapes*.
- Integração com o *website* do monitoramento de queimadas.

O Framework Bootstrap tornou a adaptação do *website* mais eficiente. Inserir novos elementos de formulários e itens ao menu de navegação são tarefas relativamente simples de serem realizadas.

5.1.4 REFERENCIAL

5.1.4.1 BANCOS DE DADOS

Os sistemas de bancos de dados são utilizados para gerenciar grandes volumes de informações, sendo uma coleção de dados inter-relacionados e programas que dão acesso aos usuários sobre estes dados.

O gerenciamento dos dados envolve a elaboração de estruturas para armazenar os dados, e mecanismos para a manipulação de informações, além de promover segurança. Durante o desenvolvimento de um Banco de Dados um modelo de dados fornece ferramentas conceituais para descrever os dados do banco, bem como suas restrições, sendo um aparato de “visualização” da organização dos dados.

Segundo Silberchatz (1995), os modelos de dados podem ser classificados em quatro categorias:

- Modelo Relacional.
- Modelo Entidade Relacionamento.
- Modelos de dados baseados em objetos.
- Modelos de dados semi-estruturados.

Durante o trabalho, foi utilizado o modelo relacional, esboçado pelo Diagrama Entidade Relacionamento ilustrado na Figura 4. O modelo relacional faz uso de tabelas nas representações dos dados e suas relações Silberchatz (1995).

Quanto à terminologia, as tabelas são chamadas de relações. As colunas de atributos, e as linhas de uma tabela de tuplas. Por exemplo, considere uma relação chamada *pessoa*. Esta tabela possui os atributos nome, idade e CPF. Uma tupla válida seria, neste caso; *José Silva, 47, 400.000.222-XX*. Desta forma, um Banco de Dados Relacionais é constituído de tuplas em diferentes tabelas. Cada tupla é unicamente identificada por um atributo especial, chamado de chave primária. CPF seria a chave primária do exemplo citado.

As tabelas relacionam-se umas com as outras através de chaves estrangeiras. Chaves estrangeiras são atributos que remetem à tuplas de outra tabela. Por exemplo, caso fosse criada uma relação para *filho*. Um atributo CPF seria suficiente para identificar o pai ou mãe da pessoa cadastrada na tabela *filho*, essa chave estrangeira seria uma referência à outra pessoa da tabela *pessoa*. Através do filho poderia se inferir o nome e a idade de seu pai ou mãe.

5.1.4.2 USABILIDADE

Entende-se por usabilidade a capacidade de um sistema oferecer meios efetivos para que o usuário realize com sucesso suas tarefas. Um sistema com alta usabilidade tornará a tarefa do usuários simples, intuitiva e direta. Ben Schneiderman (SCHNEIDERMAN, 2009) criou as 8 regras de ouro, que, quando aplicadas no desenvolvimento de uma interface aumentam a usabilidade de um sistema, são elas:

- **Consistência:** Sequência de ações similares para procedimentos parecidos. Manter padrões para as cores, Layout e fontes. Buscar o uso de mesmos termos em Menus.
- **Atalhos para usuários assíduos:** Teclas de atalho e navegação simples possibilitam que o usuário mais experiente com a interface realize suas tarefas rapidamente.
- **Feedback informativo:** Toda ação do usuário resultará em uma resposta. A resposta terá caráter informativo e/ou explicativo, conduzindo-o pela tarefa.
- **Diálogos que indiquem término da ação:** As sequências de ações do sistema devem ser organizadas de forma que o usuário entenda seus passos e saiba quando cada um deles for realizado, criando uma sequência lógica.
- **Prevenção e tratamento de erros:** A interface deve instruir o usuário a entrar com valores corretos, de modo a dificultar situações de erros

graves. A interface deve corrigi-los caso ocorram quando possível, ou então instruir o usuário a refazer uma operação nociva.

- **Reversão de ações:** Sempre que possível, as ações devem ser reversíveis, de forma que torne o usuário mais confiante para a exploração do sistema.
- **Controle:** Os usuários mais experientes devem ter a sensação de que eles dominam os processos do sistema e que ele apenas responde as suas ações.
- **Baixa carga de memorização:** O sistema deve conter uma interface simplificada para melhor memorização. Portanto é preciso uma boa estrutura e equilíbrio para relacionar elementos e facilitar a memorização subjetiva das telas.

5.2 SCRIPT PARA ANÁLISE DE ÁREAS DE PRESERVAÇÃO PERMANENTE – APP.

A plataforma de armazenamento e processamento integrado já mencionada, tem como papel armazenar arquivos e servirá como porta única de acesso a algoritmos de análises. A seguir será descrito sobre um destes algoritmos. No contexto da Engenharia Florestal, a tarefa de determinar áreas de preservação permanente, que são áreas protegidas de vegetação, constitui-se em parte por delimitar uma região de raio variável em torno de rios ou outras fontes hídricas. Esta região delimitada deve conter mata nativa segundo a lei, para proteger o manancial. No entanto é comum verificar que esta imposição muitas vezes não é atendida. Portanto, foi desenvolvido um *script* para realizar a tarefa de obtenção de APPs, que será integrado futuramente a plataforma de *shapes* e processamento integrado.

5.2.1 CONCEITOS ENVOLVIDOS

- Engenharia de requisitos.
- Algoritmos e Estruturas de Dados.
- Diagrama de atividades (UML).
- Desenvolvimento de interfaces.

5.2.2 ÁREA DE APP

Na Figura 6 é ilustrada em verde uma área de preservação permanente (APP), que são áreas protegidas localizadas no entorno de percursos hídricos.

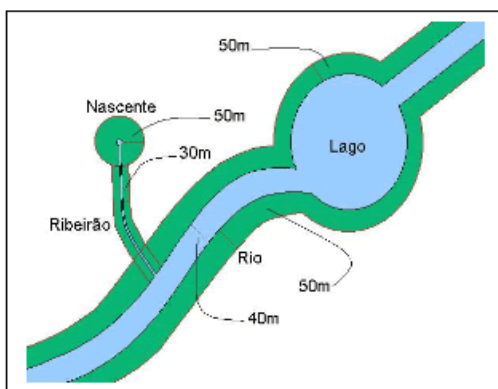


Figura 6 Áreas de preservação permanente em verde.

Fonte: http://www.atlasdasaguas.ufv.br/exemplos_aplicativos/barragens_figura26.jpg

5.2.3 OBTENDO A ÁREA DE APP

Na prática, para se obter a APP é preciso realizar uma sucessão de ações, como mostrado nos passos abaixo e ilustrado na Figura 7.

- Passo 1. Delimitar o percurso hídrico (Área azul delimitando um rio).
- Passo 2. Obter um contorno de raio x entorno do percurso delimitado. (Região amarela).

- Passo 3. Observar a ocorrência de vegetação dentro do contorno (classes dentro da região amarela delimitada).

O contorno obtido no Passo 2 corresponde a área de preservação permanente, e normalmente é de 100 metros a partir da margem do percurso hídrico, e por lei do novo código florestal essa área deve conter vegetação nativa.

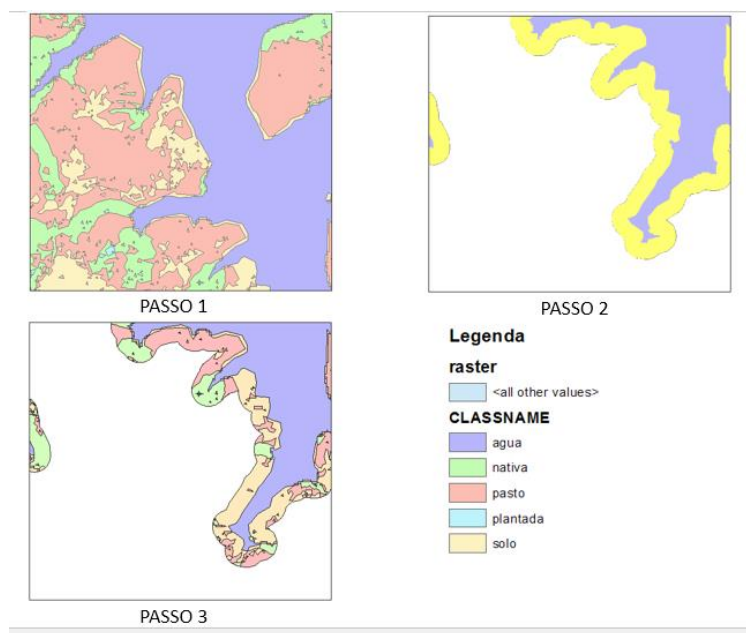


Figura 7 Etapas para a obtenção da APP.

Fonte: Arquivo Pessoal

Quando as etapas descritas são realizadas através do software ArcGis, o processo pode demorar vários minutos quando realizado pelo operador manualmente. Desta forma foi verificada a necessidade na automatização da sequência mostrada, e um *script* foi implementado. Antes de descrever o script

em detalhes, será discutido sobre o que é uma imagem classificada e classificação.

5.2.4 CLASSIFICAÇÃO DE IMAGENS

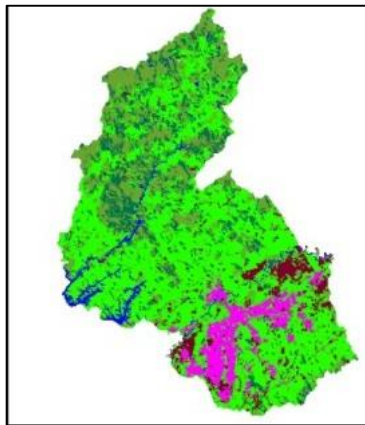


Figura 9 Imagem Classificada.
Fonte: Arquivo Pessoal

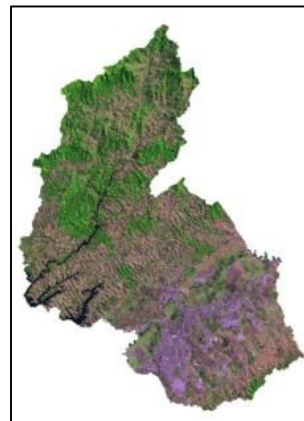


Figura 8 Fotografia de satélite.
Fonte: Arquivo Pessoal

Uma imagem classificada (Figura 8) é uma imagem *raster* onde cada pixel é classificado e identificado como um objeto da superfície de acordo com outra imagem (Figura 9). Para se realizar a classificação a partir da Figura 9, é preciso coletar polígonos de amostras que servirão de base para a classificação. A classificação pode ser realizada por um algoritmo de *clusterização* de máxima verossimilhança que agrupa um pixel de acordo com seu comportamento espectral. Resultando em uma imagem onde cada pixel corresponde a uma classe ou objeto da superfície terrestre (Ex. água, solo, floresta nativa – 3 classes).

5.2.5 FUNCIONAMENTO DO *SCRIPT* DE APP

Após a classificação, isola-se a classe água e aplica-se um contorno (Buffer) de 100 Metros obtendo a área de APP (Passo 2). Finalizando, é realizado o recorte (Clip) das outras classes que estão dentro da APP, para serem analisadas (Passo 3). Todo esse processo quando feito manualmente demanda maior tempo quanto maior o número de elementos da imagem, sendo necessário o recorte um a um destes elementos.

O Script que automatiza o processo descrito contém as seguintes sequências de funções:

- Recebe como parâmetro uma imagem classificada e a largura da área de APP.
- Obtém as classes da imagem e as armazena em uma lista.
- Realiza o Buffer na classe água.
- Faz sucessivos recortes das outras classes (Ex. solo, floresta) dentro do buffer.
- Geração de relatório contendo áreas de cada classe dentro da APP, (Figura 11).

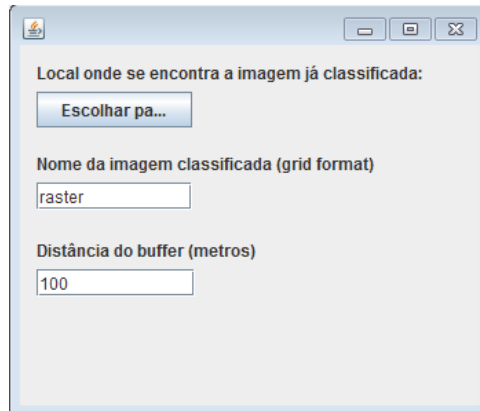


Figura 10. Interface do Script de Análise de APP em versão funcional.
Fonte: Arquivo pessoal.

A definição dos parâmetros é feita através de uma interface gráfica (Figura 10) feita em Java, e gera um arquivo de texto (.txt) contendo os seguintes parâmetros:

- Local da imagem classificada.
- Nome da imagem classificada.
- Distância da APP em metros.

Depois de ler os parâmetros, o Script cria três pastas no diretório da imagem; arquivos, *shapes* e *layers*, que armazenarão os arquivos intermediários no processamento, como o recorte das classes e a área de Buffer.

No Apêndice B, Código 1 é possível visualizar o pseudocódigo do algoritmo. Finalizando, um relatório é gerado como mostrado na Figura 11.

¹ <http://docs.oracle.com/javase/uiswing/>

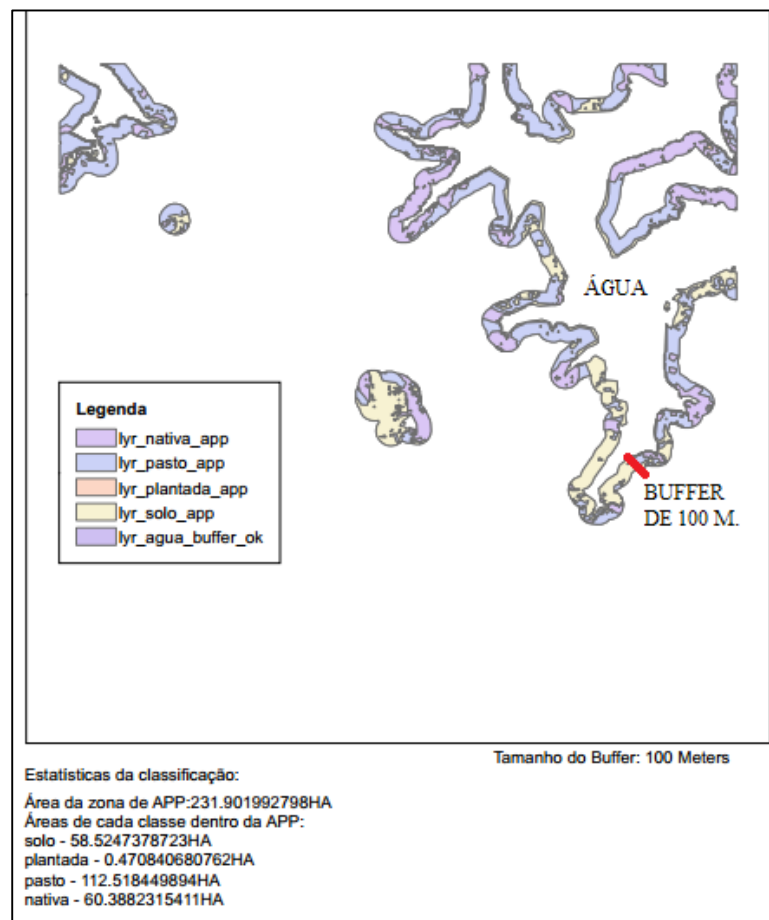


Figura 11. Relatório do algoritmo de Análise de APP.

Fonte: Arquivo Pessoal

5.2.6 TECNOLOGIAS NO DESENVOLVIMENTO DO *SCRIPT* DE APP.

Para o desenvolvimento do *Script* foram utilizadas as funções do ArcGis (ArcPy), a linguagem de programação Python, e Java para o desenvolvimento de uma simples interface gráfica – Biblioteca Swing¹.

5.2.7 DIFICULDADES NO DESENVOLVIMENTO DO *SCRIPT* DE APP.

Antes da implementação, durante a etapa de análise do script, as ferramentas UML, especificamente o diagrama de sequência, foi primordial para uma concepção lógica do processamento a ser implementado. Uma vez que o processo todo conta com a utilização de várias funções internas do ArcGis, e gera vários arquivos. O diagrama de atividades esboça a sequência de operações e provê a visualização necessária para um projeto eficiente. Além das funções internas do ArcGis, os arquivos gerados precisam ser organizados de forma lógica em diretórios para melhor atender o usuário em encontrar o que procura.

Na Figura 12 é ilustrado o diagrama de atividades do algoritmo para análise de APP. O círculo negro do lado superior esquerdo indica início da atividade, os retângulos que seguem são as atividades, e na sequência têm-se o último círculo que denota final da atividade.

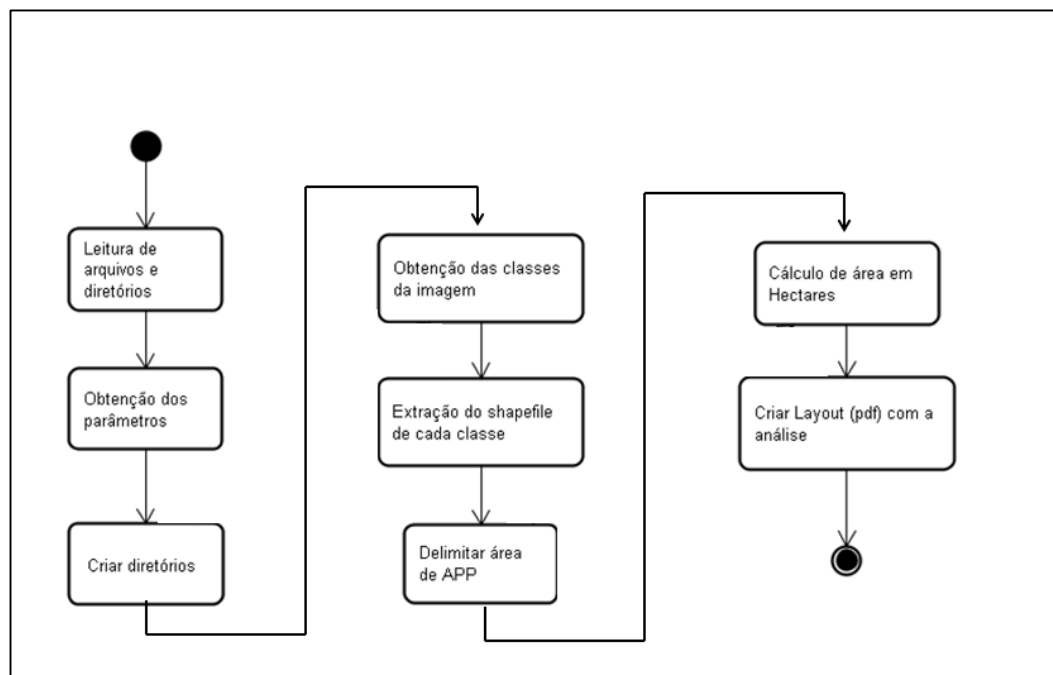


Figura 12 Diagrama de atividades do algoritmo de APP.

Fonte: Arquivo Pessoal

5.2.8 REFERENCIAL

5.2.8.1 UNIFIED MODELLING LANGUAGE – UML

Será apresentado a seguir a UML - *Unified Modeling Language*, que é um conjunto de ferramentas utilizadas durante o ciclo de vida de desenvolvimento dos software e provê meios padronizados de se representar funcionalidades dos sistemas, dentre outras características (BOOCH et al, 2006).

Durante o projeto de desenvolvimento de software, tal qual o desenvolvimento de um produto tangível, o primeiro passo é responder algumas

questões cruciais para garantir o que irá ser feito corresponderá a necessidade de seu uso. Se tratando de software, a engenharia de requisitos é quem fornece as técnicas de levantamento, documentação e análise das características do software, que também são chamadas de requisitos (PAULA FILHO, 2003).

- Qual problema o software se propõe a resolver?
- Quais características o software deverá apresentar?
- Haverá alteração no uso deste software?

Essas perguntas são alguns exemplos de questões que podem ser levantadas durante a etapa inicial de um projeto de software. E servirão como base para o planejamento e desenvolvimento de todo o projeto.

Os requisitos de um software são classificados em requisitos funcionais e requisitos não funcionais (PAULA FILHO, 2003).

- Requisitos Funcionais: Representam as características funcionais de um software (cadastrar produto, excluir cliente) e são representados pela UML através de diagramas de casos de uso.
- Requisitos Não Funcionais: Representam as características comportamentais de um sistema, não se limitando a funcionalidades individuais (rapidez, segurança).

A UML auxilia as tarefas da engenharia de requisitos, trazendo diagramas e provendo uma linguagem comum aos desenvolvedores e *stakeholders*, para que possam participar de forma conjunta durante o projeto de um novo software, desde sua concepção até sua entrega. A UML traz ferramentas para a visualização, especificação, construção e documentação de um sistema de informação. Surgida na década de 90, resultante da união de métodos já existentes (BOOCH, *Object Oriented Software Engineering e Object Modeling Technique*).

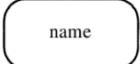
O uso da UML traz melhor entendimento de um sistema de forma geral, pois suas ferramentas criam diagramas que representam a sua estrutura, comportamento e limites.

Os principais diagramas da UML são:

- Diagrama de Classe: Utilizado para sistemas orientados a objetos, descrevem classes com seus métodos e atributos.
- Diagrama de caso de Uso: Descreve comportamentos do sistema e são utilizados para obter as funcionalidades desejadas pelo cliente (cadastrar produto, excluir cliente).
- Diagrama de estado: Descreve os estados de um sistema (ocupado, carregando, executando).
- Diagrama de sequência: Descreve como os componentes e objetos interagem entre si ao longo do tempo, representado por objetos e mensagens.
- Diagrama de atividade: Descreve a sequência de atividades em um único processo, mostrando suas dependências.

Durante a atividade de obter a área de APP foi utilizado o diagrama de atividades. O diagrama de atividades mostra o fluxo de controle e de dados, mostrando a sequência de ações realizadas no decorrer de um processo, é adequado para se representar o *workflow* e as lógicas do negócio. É formado principalmente pelos elementos mostrados na Tabela 1.

Tabela 1. Elementos básicos do Diagrama de Atividades.

	Nó de decisão ou Merge. Pode bifurcar o fluxo de controle de acordo com condições de guarda, ou representar um ponto final comum de diversos fluxos de controle.
	Nó de sincronização. Representa fluxos de controle que ocorrem em paralelo. Pode ser do tipo <i>fork</i> ou do tipo <i>join</i> . Especificando o início de fluxo de controle paralelos ou representar um ponto final comum de fluxos de controle paralelos.
	O nó inicial representa o início de uma atividade.
	O nó final representa o fim de uma atividade.
	Representação de uma ação, que é um bloco elementar de uma atividade.

5.3 SCRIPT PARA POSICIONAMENTO DE TORRES

Outro algoritmo de análise desenvolvido a ser integrado à plataforma é o *script* para posicionamento de torres. No contexto de Engenharia Florestal, quando uma área precisa ser monitorada (ex.: Controle de incêndio) podem ser utilizadas Torres de visada, que são torres distribuídas ao longo da área de interesse a ser monitorada. A definição da localização das torres é crucial para uma boa cobertura. Para se definir a localização de torres de visada deve-se considerar:

- A topografia da área.
- A altura da torre.
- Os ângulos de visão de cada torre.
- A curvatura da superfície terrestre.

5.3.1 CONCEITOS ENVOLVIDOS

- Projeto e Análise de Algoritmos.
- Algoritmos e Estruturas de Dados.

Na Figura 13 é possível visualizar em azul as áreas de interesse a serem monitoradas (fazendas), e os pontos escuros que são as torres, estas torres serão escolhidas de modo que maximize a cobertura de visão dentro da área de interesse.

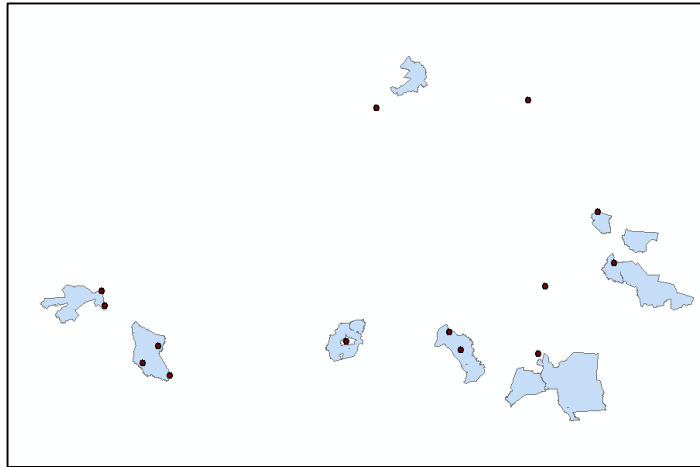


Figura 13. *Shapes* das áreas de interesse e pontos de torres de visão.
 Fonte: Arquivo Pessoal

A Figura 14 mostra o campo de visão da torre mais acima da Figura 13.
 Este campo de visão também é chamado de *viewshed*.

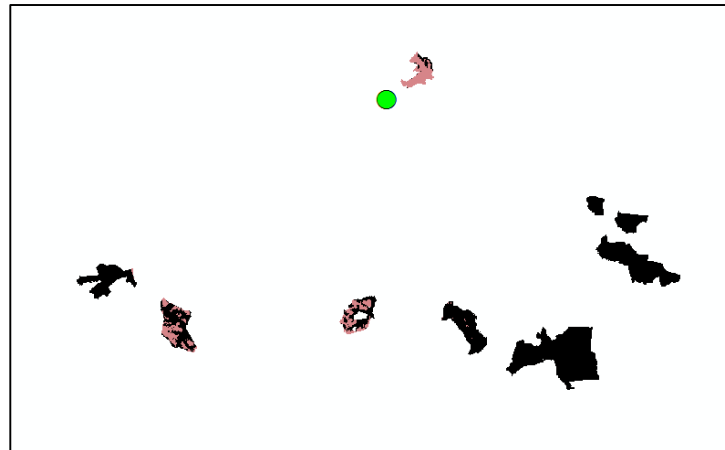


Figura 14. *Viewshed* da torre em verde. As áreas rosadas são visíveis para a torre.
 Fonte: Arquivo Pessoal

O *viewshed* de uma torre pode ser combinado com outros *viewsheds* de outras torres para se obter a visibilidade de um conjunto de torres. Logo existe uma combinação de torres tal que seu *viewshed* seja o maior possível para uma área de interesse.

Desta forma, foi desenvolvido um *Script* que encontra as melhores torres para cobrir determinada área de interesse. O problema em questão trata de encontrar uma combinação s de n torres que torne maior possível a área de visão para uma área delimitada. O recurso matemático que suporta a resolução do problema é o cálculo de combinações, sem considerar repetição ou ordem.

$$C_s^n = \binom{n}{s} = \frac{n!}{s! \times (n - s)!}$$

Equação 1. Cálculo de combinações sem ordem e sem repetição de n elementos tomados em s .

5.3.2 FUNCIONAMENTO DO *SCRIPT* PARA POSICIONAMENTO DE TORRES

O *Script* faz o *viewshed* de todas as combinações possíveis para um dado conjunto de torres e armazena em uma tabela, para cada combinação, a porcentagem de pixels visíveis dentro da área de interesse. O pseudocódigo pode ser visualizado no Apêndice B, código 2.

Na Figura 15 os círculos verdes representam a posição de 12 torres distribuídas sobre o modelo de elevação, que já foi apresentado na seção sobre dados geográficos. Uma torre posicionada em um local mais alto do terreno

naturalmente têm um campo de visão mais amplo, enquanto que uma torre posicionada em uma área de vale possui seu campo de visão reduzido.

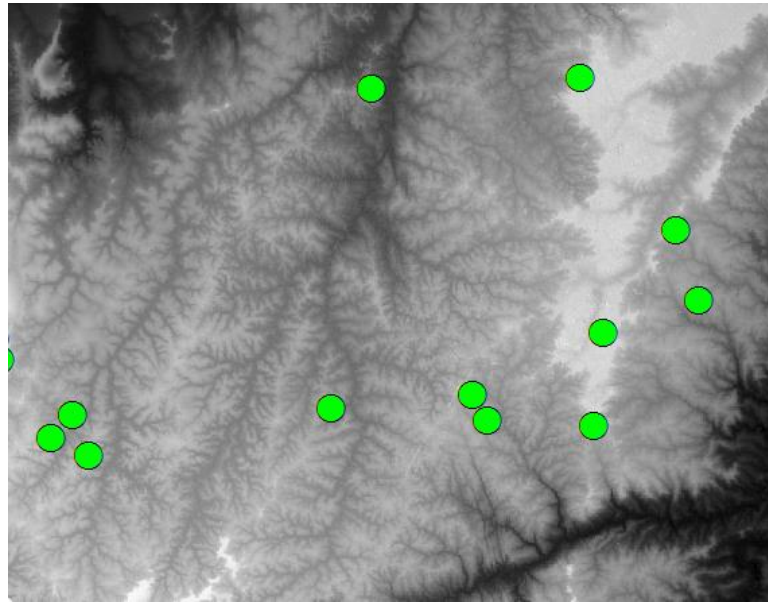


Figura 15. Distribuição de torres (pontos verdes) sobre um modelo de elevação.

Fonte: Arquivo pessoal.

O *script* recebe os parâmetros através do *prompt* do interpretador Python:

- Imagem do modelo de elevação (*raster*).
- Pontos que representam as posições das torres (*shape*).
- Áreas de interesse a ser monitorada (*shape*).

Ao final do processo de obtenção dos *viewsheds* é gerada uma tabela similar à Tabela 2, que contém um ranking de combinações de torres e sua porcentagem de visibilidade dentro da área de interesse.

Tabela 2. Resultado do Algoritmo de Combinação de Torres.

Resultado 4a4			
	OID	Combinação	Pixels inv
	269	[1, 9, 10, 14]	63,5179
	274	[1, 9, 12, 14]	63,4377
	779	[4, 9, 12, 14]	62
	774	[4, 9, 10, 14]	61,9979
	151	[1, 4, 9, 10]	61,9892
	106	[1, 3, 9, 10]	61,9415
	858	[5, 9, 10, 14]	61,8168
	863	[5, 9, 12, 14]	61,7925
	654	[3, 9, 10, 14]	61,7907
	659	[3, 9, 12, 14]	61,7727
	187	[1, 5, 9, 10]	61,6037

Na Tabela 2, que apresenta o resultado da combinação de 14 torres, tomadas 4 a 4. A coluna *Pixels_inv* mostra a porcentagem de cobertura dentro da área de interesse para cada combinação, em ordem decendente. As melhores torres foram 1, 9, 10 e 14.

5.3.3 TECNOLOGIAS UTILIZADAS NO SCRIPT PARA POSICIONAMENTO DE TORRES

Somente a biblioteca ArcPy foi utilizada na atividade.

5.3.4 DIFICULDADES ENCONTRADAS NO SCRIPT PARA POSICIONAMENTO DE TORRES

O número de combinações cresce exponencialmente quando uma nova torre é adicionada à análise. Como pode ser visualizado na Tabela 3.

Tabela 3. Número de combinações em função do número de torres n e agrupamento s .

n	s	Nº COMBINAÇÕES
15	3	455
15	4	1365
16	3	560
16	4	1820

Algoritmos combinatórios, que são algoritmos para investigar estruturas combinatórias, podem ser divididos em algoritmos de geração, enumeração e busca combinatória. (GODRICH et al, 2004). Os algoritmos combinatórios fazem parte da classe NP (GODRICH et al, 2004), ou seja, não se conhece ainda algum algoritmo combinatório que se resolva em tempo polinomial em relação a sua entrada.

Devido a este fato, quando o número de torres a ser analisado pelo *script* é grande, o processamento se torna proibitivo em termos de tempo e espaço em memória. Por exemplo, vinte torres combinadas de quatro a quatro ou cinco a cinco geram juntas o total de 20.349 combinações. Sabendo que cada combinação demora em média 2 segundos e gera um *raster* de 8 MB, temos um tempo de 11h, e 162GB de arquivos. Para amenizar esses gastos duas estratégias foram criadas:

- Descarte de arquivos intermediários.
- Agrupamento de torres.

A estratégia de agrupamento de torres rotula cada torre com uma letra, que representa o grupo ao qual aquela torre pertence. Como mostrado na Tabela 4.

Tabela 4. Torres e seus rótulos.

TORRE	RÓTULO
1	A
2	B
3	B
4	C

Se duas torres T1 e T2 estão próximas então, dificilmente estarão juntas em uma combinação que objetiva aumentar o campo de visão, e consequentemente qualquer combinação que contenha T1 e T2 pode ser descartada como opção.

O agrupamento serve para identificar as torres que dificilmente configuram uma boa opção, podendo fazer parte do mesmo grupo e sendo rotuladas com a mesma letra as torres próximas ou que possuem o campo de visão mais ou menos similar. Desta forma o número de combinações diminui de forma significativa, economizando recursos computacionais. Na Tabela 5 estão o número de combinações 2 a 2 possíveis com e sem a estratégia de agrupamento para as torres da Tabela 4.

Tabela 5. Número de combinações utilizando a estratégia de agrupamento.

COM AGRUPAMENTO	SEM AGRUPAMENTO	
3	6	Nº COMBINAÇÕES

Para se obter o número de combinações possíveis utilizando a estratégia de agrupamento basta, substituir o n da Fórmula 1 pelo número de grupos ou letras distintas de rótulos.

6 CONCLUSÕES

Esta seção é dividida em considerações finais e trabalhos futuros.

6.1 CONSIDERAÇÕES FINAIS

O estágio realizado no LEMAF foi particularmente rico em aprendizado, pois a área de geoprocessamento possui inúmeras possibilidades de aplicação de métodos computacionais.

Nos trabalhos realizados, diversos algoritmos e scripts foram desenvolvidos objetivando a extração de informações geográficas de forma automática e eficiente, integrando Sistemas Geográficos, Bancos de Dados, Projeto e Análise de Algoritmos, Engenharia de Requisitos e Tecnologias *web*. A plataforma para armazenamento de *shapes* e processamento integrado sintetiza os vários conceitos computacionais aprendidos durante a graduação.

A experiência organizacional adquirida trouxe maior maturidade em corresponder a prazos e compromissos, além de possibilitar a vivência de uma equipe. Dentro do LEMAF existem pessoas especializadas em Banco de Dados, desenvolvimento *web*, administração de servidores, o auxílio destas pessoas foi essencial a conclusão das atividades.

Além das atividades desenvolvidas, que perpassam as atividades da Engenharia de *software*, o estágio trouxe grande experiência organizacional e pessoal. Os trabalhos realizados tiveram grande participação dos *stakeholders*, a prototipação, o uso de diagramas UML e testes de validação trouxeram os usuários dos sistemas desenvolvidos a participar e contribuir para a qualidade dos sistemas gerados. O LEMAF, também obteve dos trabalhos realizados grande aproveitamento, por permitir maior eficiência e organização em seus trabalhos e processos.

6.2 TRABALHOS FUTUROS.

Para atividades futuras estão previstas alterações e modificações nas tecnologias geradas até o momento, neste sentido, desde o começo dos trabalhos tarefas de documentação (descrições, ilustrações conceituais e comentários em código-fonte) têm sido feitas para os sistemas e *scripts* desenvolvidos.

Outro trabalho a ser consolidado futuramente é intensificar as tarefas da Engenharia de Requisitos e participação dos *stakeholders*, que se mostraram efetiva ao gerar produtos considerados de boa qualidade ao laboratório. Para tal objetivo serão estudadas novas ferramentas e técnicas de testes de validação e levantamento de requisitos.

Outro ponto a ser trabalhado é a integração de programação Linear na resolução dos problemas em geoprocessamento através do software LP SOVE. Foi observado um possível ganho em desempenho ao se utilizar a programação linear, este fato foi observado ao se estudar o algoritmo de posicionamento de torres, que poderia ser modelado de outra maneira.

O algoritmo para análise de APP, gerou interesse aos desenvolvedores da plataforma CAR, que viram no algoritmo uma poderosa ferramenta aliada a proteção ambiental que poderá futuramente ser integrada ao sistema.

A plataforma para organização de *shapes* e processamento integrado será modificada a fim de permitir procedimentos de análises geoespaciais pela *web*, analogamente a um sistema OLAP. Para tal, será estudada a integração de Python e PHP.

REFERÊNCIAS BIBLIOGRÁFICAS

ArcGIS Resources. 2013. **ArcGIS Resources**. Disponível em <http://resources.arcgis.com/en/help/main/10.1/index.html#/Welcome_to_the_ArcGIS_Help_Library/00qn0000001p000000/>. Acesso em 07 de julho de 2014.

ArcGIS 2014, **ArcGIS Resources**. Disponível em <<https://www.arcgis.com/features/>>. Acesso em 11 de outubro de 2014.

BEZERRA, E. **Princípios de Análise e Projeto de Sistemas com UML**. 2007, Elsevier. p.5 – 15, Capítulo 5. Disponível em:< http://fabriciosousa.com/AMS_Cap5_Diagrama_Classe.pdf> Acesso em 01 de dezembro de 2014.

BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. **UML: guia do usuário**. Elsevier Brasil. 2006. p. 22.

BORGES, K. A. V. **Modelagem de dados geográficos uma extensão do modelo OMT para aplicações geográficas**. Dissertação (Mestrado) - Escola de Governo de Minas Gerais, Fundação João Pinheiro, Belo Horizonte. 1997.

BOTELHO, M. A. **Incorporação de facilidades espaço-temporais em banco de dados orientados a objetos**. Dissertação (Mestrado). Unicamp, SP. 1997.

CALDWELL, B.; COOPER, M.; REID, L. G.; VANDERHEIDEN, G. (2008) **'Web Content Accessibility Guidelines 2.0'**, Web Accessibility Initiative (WAI), World Wide Web Consortium (W3C) [online], Disponível em: <<http://www.w3.org/TR/WCAG20>> Acesso em 01 de dezembro de 2014.

CÂMARA, G. **Modelos, Linguagens e Arquiteturas para Bancos de Dados Geográficos**. São José dos Campos, SP: Tese (Doutorado). Instituto Nacional de Pesquisas Espaciais (INPE). 1995.

CÂMARA, G.; CLODOVEU, D. **Fundamentos de geoprocessamento**. Instituto Nacional De Pesquisas Especiais (INPE). São José dos Campos 2001. Cap. 1, p. 1.

DANGERMOND, J. **A classification of software components commonly used in Geographic Information Systems**. SPAD Systems, N.Y., 1983. Cap 3. p. 35.

GODRICH, M. T.; ROBERTO T. **Projeto de Algoritmos: Fundamentos, análise e exemplos da Internet**. Bookman, 2004. Cap. 13. p. 593.

HEUSER, C. A. **Projeto de Banco de Dados**. 6a Ed., Bookman, 2008. Cap 6. p. 125.

KORTH, H. F.; SILBERSCHATZ, A.; SUDARSHAN, S. **Sistema de Banco de Dados**. 5a ed. 2006. Cap. 7. p. 380.

LAURINI, R.; THOMPSON, D. **Fundamentals of Spatial Information Systems**. San Diego: Academic Press, 1992. Cap. 1, p. 52.

LOPES, ARTHUR VARGAS. **Estrutura de Dados**. Vol. 1. Editora da ULBRA. 1999. Cap. 3. p. 234.

MEDEIROS, C. B.; PIRES, F. **Databases for GIS**. SIGMOD Record, v.23, 1994. p. 107-115.

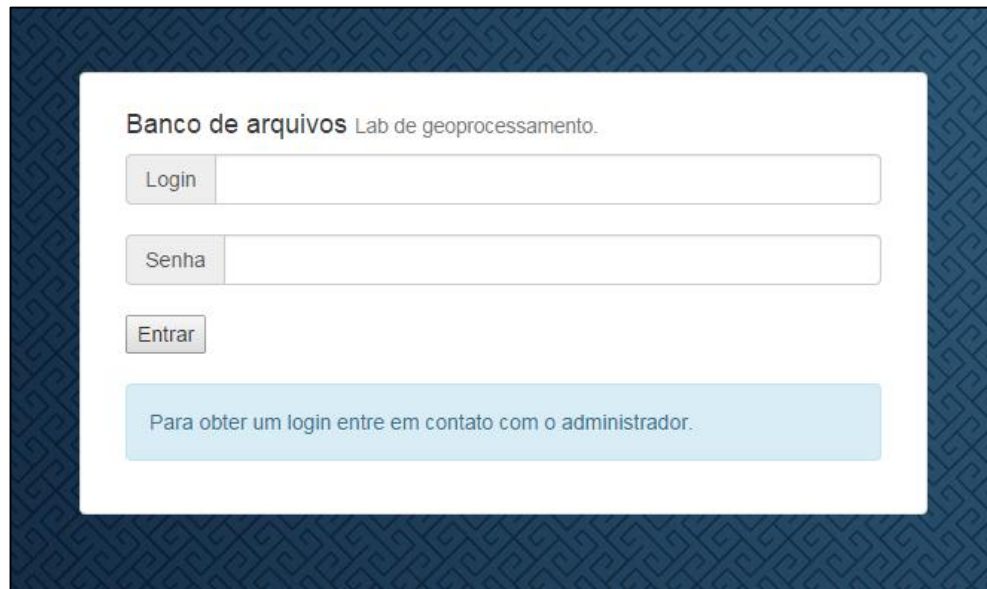
PAULA, F.; WILSON, P. **Engenharia de software**. Vol. 2. 2003. Cap 3. p. 150.

PEUQUET, D.J.; MARBLE, D.F. **Introductory readings in Geographic Information Systems**, 1990. Cap. 1. p. 62.

SHNEIDERMAN, B. **Designing the user interface** (Terceira edição). Addison-Wesley, 1998. Cap 2.

SILBERCHATZ, K.H.F. **Sistemas de Banco de Dados**, 1998. Cap. 2. p. 40.

APÊNDICE A – Interfaces da plataforma de organização de shapes.



The image shows a login form titled "Banco de arquivos" with the subtitle "Lab de geoprocessamento." The form is set against a dark blue background with a repeating geometric pattern. It contains two input fields: "Login" and "Senha" (Password). Below these fields is an "Entrar" (Enter) button. At the bottom of the form, there is a light blue rectangular box containing the text: "Para obter um login entre em contato com o administrador."

Figura 1. Formulário de Login.

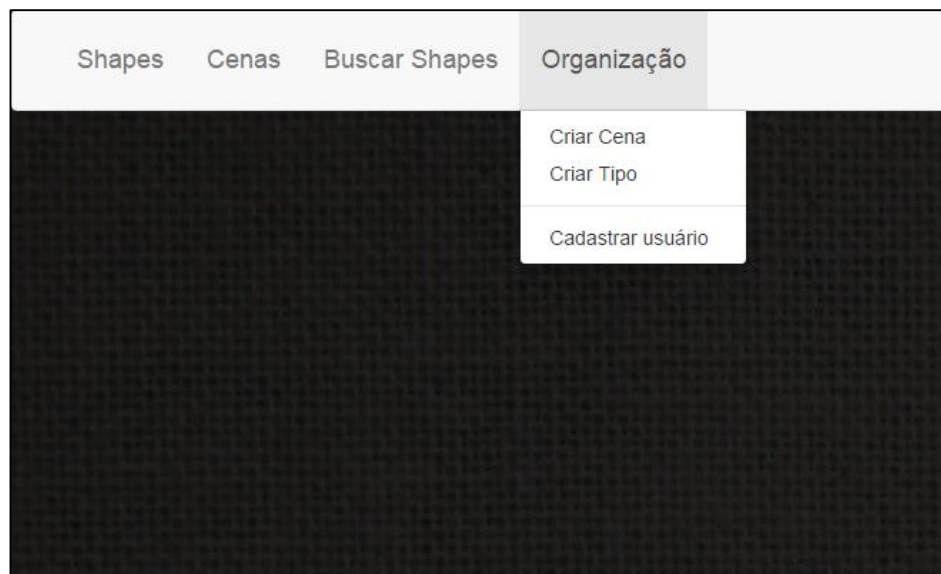


Figura 2. Barra de navegação principal

[Shapes](#) [Cenas](#) [Buscar Shapes](#) [Organização](#)

Para visualizar todos os shapes de uma cena vá até a guia Cenas.

Cena

Cena 1

▼

Nome do shape

arquivo

Usuário

Rodolfo

▼

Tipo

Queimada

▼

Buscar

Posterior à

09/01/2014

Anterior à

10/01/2014

Para buscar em um dia específico coloque as duas datas iguais ao dia.

Laboratório de geoprocessamento

X arquivos armazenados

Acesso como Mark Otto

Figura 3. Formulário de busca de shapes.

APÊNDICE B – Trechos de código.

```
Obtem_APP(distancia_app, classificacao_raster)

// Obtém as classes do raster classificado

classes = extrair_classes(classificacao_raster)

// Obtém a classe água e realiza o buffer

Para i=0 até comprimento(classes)

    se classe == 'água'

        buffer = extrai_buffer(classe, distancia_app)

// Retira buffer excedente a borda do mapa

buffer_corrigido = corta_buffer(buffer, classificacao_raster)

// Extrai as classes contidas dentro da APP

Para i=0 até comprimento(classes)

    se classe != 'água'

        classes_app <- classe_contida(buffer_corrigido, classe)

// Faz análise das classes contidas no APP e gera PDF

string analise = analise_app(classes_app)

monta_arquivo_pdf(analise, classes_app)
```

Código 1. Trecho de código para realização do *Buffer*.

```

Escolhe_torres(shape_torres, shape_area, n, s)

  // Extrai o shape contendo os pontos de cada torre e armazena em uma lista
  torres = extrair_torres(shape_torres)

  // Extrai o viewshed de cada torre
  Para i=0 até comprimento(torres)
    viewshed[i] = extrair_viewshed(torre[i])

  // Obtém as combinações
  combinacoes = obter_combinacoes(n,s)

  // Faz o viewshed das torres de cada combinação
  Para i=0 até comprimento(combinacoes)
    viewshed_combinacao =extrair_viewshed(combinacao[i], torre[i,i+1..])

    // Obtem a porcentagem da área de cobertura
    area_cobertura = extrair_cobertura(viewshed_combinacao, shape_area)

    // Insere na tabela as torres da combinação, e a área de cobertura
    Tabela_cobertura <- (combinacao, area_cobertura)

```

Código 2. Trecho de código para obtenção de *viewshed*.