

APLICATIVO DE RECONHECIMENTO DE IMAGENS EM DISPOSITIVOS MÓVEIS PARA AMBIENTES PREVIAMENTE MAPEADOS

Tiago Souza Tomasel¹

RESUMO

Este artigo apresenta os resultados da avaliação de uma ferramenta de software desenvolvida utilizando técnicas de processamento de imagens para reconhecimento e identificação de objetos. A ferramenta desenvolvida, nomeada SIROAM (Sistema Integrado de Reconhecimento de Objetos em Ambientes Mapeados) foi avaliada através de sua utilização, medindo sua confiabilidade através de feedback e análise de desempenho do seu algoritmo de identificação. Através de testes realizados de uma forma não intrusiva para reconhecimento de imagens, foi descoberto que algumas variáveis de ambiente influenciam na identificação de um objeto e que os índices de taxa de acertos são bons. Também como resultado da avaliação descobriu-se que o algoritmo de identificação utilizado foi considerado escalável devido ao seu desempenho que reduz de forma linear.

Palavras-chave: Identificação de Imagens. Recuperação de Imagens. Indexação de Imagens. Dispositivos Móveis. SIROAM.

1 INTRODUÇÃO

Vários sistemas que realizam a identificação de imagens em dispositivos móveis foram desenvolvidos, como, por exemplo, o Googles desenvolvido pela empresa Google. O desenvolvimento de um sistema deste tipo implica a utilização de recursos avançados de processamento de imagens. Um exemplo é permitir que a imagem de um objeto seja reconhecida em outra imagem captada por um dispositivo e assim se entenderá que o objeto procurado é aquele que teve mais semelhança dentro de um conjunto de comparações.

A motivação deste trabalho surge no contexto da educação patrimonial, onde muitas vezes não é possível ou desejável interferir no ambiente. Acredita-se no potencial benefício

¹ Discente do curso Ciência da Computação do Centro Universitário La Salle – Unilasalle, matriculado na disciplina de Trabalho de Conclusão II, sob orientação da Prof^ª. Dr^ª. Patrícia Kayser Vargas Mangan. E-mail kayser@unilasalle.edu.br.

para a sociedade que o resultado dessa pesquisa pode trazer, favorecendo a disseminação do conhecimento através do mapeamento e identificação de objetos, sem exigir nenhum tipo de sensor ou marcador, podendo ser aplicado em diversos ambientes, sem a necessidade de mudanças físicas nos locais para o funcionamento do mesmo. A solução proposta contribui para inúmeras situações de localização e reconhecimento de locais/objetos, pois, muitas vezes visita-se locais que possuem características ou objetos únicos em seu interior e que nem sempre sabe-se do que se trata e qual o seu significado, também pode-se citar sobre a localização dentro de um ambiente, pode-se chegar em um local que não sabe-se onde está e para onde se vai quando se está neste lugar. Pode-se dizer também que um indivíduo está em um local dotado de tecnologia e que o mesmo precisa ter recursos avançados para guiar as pessoas em seu interior. Baseando-se nessas situações, existem várias maneiras de resolver este problema colocando-se nesses locais, sensores, painéis digitais, auto-falantes entre outros. Porém, essas soluções geram custos e nem sempre são viáveis, pois, dependendo do local, pode não ser possível instalar, colar ou ligar qualquer tipo de equipamento, pois, esses objetos ou locais de relevância histórica e cultural podem ter restrições que impeçam alterações.

Assim, considerando-se esse contexto, o problema de pesquisa investigado pode ser definido do seguinte modo: como identificar determinado objeto em um ambiente da forma mais pervasiva possível, sem necessitar de recursos instalados no ambiente em que o mesmo se encontra para que as características originais deste ambiente sejam preservadas.

Considerando o problema de pesquisa levantado, este trabalho tem como objetivo geral investigar e implementar o reconhecimento de imagens através de técnicas de processamento de imagens para identificação de objetos em um determinado ambiente previamente mapeado sem a necessidade de alterações físicas nestes ambientes.

A constante evolução dos dispositivos móveis (*Tablet e Smartphone*) traz consigo uma gama de facilidades em nossas vidas. Estes aparelhos estão equipados com memória e processador de maiores capacidades e processamento do que alguns computadores lançados a algumas décadas atrás. Com esta característica importante, pode-se portar para estes dispositivos, softwares que exigem maior poder de processamento, como, por exemplo: jogos e editores de imagens e vídeos. Com base nas características dos dispositivos móveis de hoje, eles se tornam o principal aliado na execução da solução do problema encontrado.

Para efetuar a identificação de imagens, foi desenvolvido um software, nomeado SIROAM (Sistema Integrado de Reconhecimento de Objetos em Ambientes Mapeados) que implementa o mapeamento de locais e objetos e também a identificação de objetos,

utilizando-se de uma arquitetura mista (*Web e Mobile*), técnicas de processamento de imagens e tecnologias avançadas de desenvolvimento de software, bem como bibliotecas de código fonte aberto. Os resultados da avaliação indicam bons índices para taxa de acerto e para escalabilidade do sistema e apontam para a viabilidade de uso do protótipo em projetos reais de educação patrimonial.

Este trabalho está organizado com a seção II apresentando a metodologia de desenvolvimento do projeto. Já na seção III será apresentada a fundamentação teórica. Na seção IV serão apresentados alguns trabalhos relacionados com a seção V que trata do desenvolvimento do software. Na seção VI serão apresentados os resultados da avaliação de confiabilidade do software e de desempenho de seu algoritmo de reconhecimento de imagens, além da verificação dos objetivos. A seção VII finaliza esse trabalho apresentando as conclusões finais e os trabalhos futuros.

2 METODOLOGIA

Para solucionar o problema da identificação de objetos através do reconhecimento de imagens realizou-se estudo sobre Identificação de Imagens baseada em seus aspectos cromáticos, utilizando-se de técnicas de processamento de imagens.

Os conteúdos cromático e estrutural das imagens citados em [1] são características imediatamente percebidas pelo observador humano, e desta forma são elementos importantes para diferenciá-las. Ao invés de serem descritas por palavras-chaves, as imagens passaram a ser indexadas pelo seu próprio conteúdo visual, através de cor, textura, forma entre outras características [1].

Muitas são as propriedades que podem ser recuperadas a partir de uma imagem e estas podem ser características gerais ou de domínio específico. Conforme Hoshiro e Goldschmidt [2], as características gerais são aquelas que são encontradas facilmente em qualquer imagem, tais como cor, textura e forma, bem como uma combinação destas. As características de domínio específico, como o próprio nome sugere, são aquelas que dependem da área de aplicação do sistema em questão [2].

Considerando as afirmações acima, foi proposto um sistema para receber as imagens dos locais/objetos a serem identificados, onde é possível colocar as informações do mesmo como identificação, descrição, entre outras. Com isso o sistema extrairá as características destas imagens e guardará em um banco de dados. Este sistema rodará em um computador, ambiente

web, multiplataforma, pois com estas características não haverá necessidade de instalação de software no computador que será utilizado para mapear os objetos de um ambiente.

Os dispositivos móveis serão utilizados na segunda parte deste sistema, que será um aplicativo que captura foto do local onde a pessoa está, extrai as características da foto e procura no banco de dados qual a imagem mais semelhante com a que foi capturada. Uma vez encontrada, ele deverá exibir na tela todas as informações do local/objeto encontrado.

3 FUNDAMENTAÇÃO TEÓRICA

Com a necessidade de se trabalhar com reconhecimento de imagens, é preciso, para ter um conhecimento mais específico, saber como uma imagem é conhecida computacionalmente e também quais são os métodos de comparação entre imagens.

A execução do projeto depende muito das técnicas de processamento de imagens que serão utilizadas para efetuar a comparação das imagens. Comparar simplesmente pixel a pixel entre duas imagens não trará resultados bons, muito menos aproximados, pois, fotos capturadas com câmeras diferentes já trazem uma boa diferença visível a olho humano e uma comparação deste tipo, requer que cada pixel da imagem original seja igual ao pixel da imagem capturada e conforme citado em [17], se uma pessoa capturar a imagem de uma cena cem vezes, obterá cem imagens diferentes. Com base nos estudos, chegou-se a conclusão que existem outras maneiras de comparar imagens, extraindo propriedades que possuem dados que podem ser comparados matematicamente e então tornarem o sistema apto a fazer esta tarefa. Existem vários recursos utilizados para processamento de imagens, e isto tem se tornado cada vez mais comum, conforme ressalta [2] “são inúmeras as áreas onde a quantidade de imagens armazenadas vem aumentando significativamente nos últimos anos [...]”, com isso, aumentam as necessidades de indexação de imagens, recuperação e localização de imagens através de processos digitais.

3.1 Definições de Imagem

Conforme observação em [2] “Do ponto de vista matemático, uma imagem é considerada uma função bidimensional $f(x,y)$ onde x e y são coordenadas planas, e a amplitude de f em qualquer par de coordenadas (x,y) é chamada de intensidade ou nível de cinza da imagem no referido ponto. Quando (x,y) e a amplitude de f fazem parte de um conjunto de valores finitos, ou discretos, a imagem é chamada de imagem digital [...]” [2].

Conclui-se que “[...] uma imagem é uma matriz $n \times m$ onde cada elemento da matriz corresponde à intensidade ou nível de cinza $f(x,y)$ em um determinado ponto da imagem [...]” [2].

Contudo ainda é possível segmentar uma imagem em pedaços menores, para facilitar a análise. Conforme citado em [3] a segmentação de imagens é uma abordagem modular para obter valores de cor de imagem mais precisa. Segmentação de uma imagem significa dividir uma imagem em várias partes pequenas. O objetivo da segmentação é simplificar a representação da imagem facilitando o acesso e a análise. Os segmentos de imagens representam coletivamente toda a imagem principal. Por segmentação é possível obter diferentes características como: cor e textura de uma imagem e essas características podem ser manipuladas e calculadas facilmente.

A imagem colorida é representado por três matrizes de duas dimensões (2D) após a digitalização em RGB^2 do espaço de cor. Uma matriz é para a cor vermelha e outras duas são, respectivamente, para o verde e o azul. Assim, para uma imagem colorida temos três matrizes 2D do mesmo tamanho como o tamanho da imagem em pixels. Para cada imagem, temos três valores de cor após o processamento.

3.2 Métodos de Comparação de Imagens

Foi possível observar que várias propriedades podem ser extraídas das imagens para efetuar suas comparações e indexação e leva destaque os descritores de cores, mais especificamente os histogramas de cores.

Em [4] o histograma de cores descreve a ocorrência das cores em uma imagem. Para a obtenção de compactabilidade e desempenho, normalmente os histogramas apresentam a frequência de um intervalo de valores ao invés de um único valor de cor. Outras vezes, ocorre a fixação do número de cores utilizadas, seja pela maior frequência dessa cor na coleção de imagens, ou pelo seu poder de discriminação frente as demais.

Histograma de Cores, de acordo com Costa [1], “que é uma característica que não leva em consideração a distribuição espacial das cores, e é o descritor mais largamente utilizado em recuperação de imagens [...]” .

Aliado ao processamento de imagens, também é necessário apresentar algumas informações referentes ao geoprocessamento e ao georreferenciamento. Isso se deve ao fato da

² RGB é a sigla do sistema de cores aditivas formado pelas iniciais das cores em inglês Red, Green e Blue, que significa em português, respectivamente, Vermelho, Verde e Azul. Um píxel é formado por estas três cores.

ferramenta que foi desenvolvida utilizar esses recursos para refinar pesquisas no banco de dados e por sua vez, reduzir tempo de processamento de imagens.

3.3 Geoprocessamento e Georreferenciamento

Conforme citado em [16], georreferenciamento e geoprocessamento são formas de obter dados geográficos. O geoprocessamento é o processamento de dados georreferenciados.

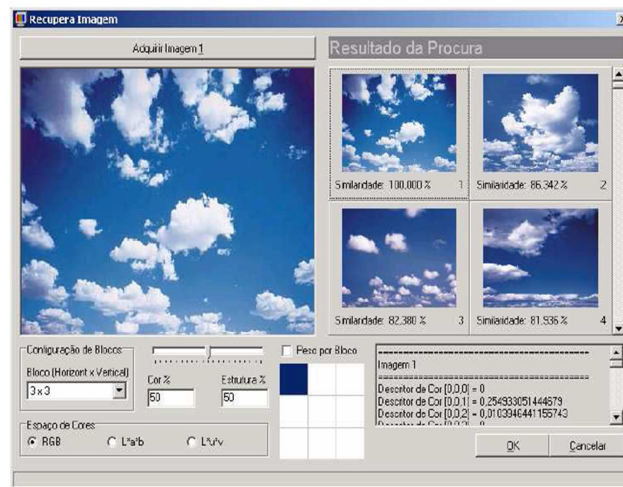
Também em [16] é apresentado uma definição sobre georreferenciamento “[...] georreferenciar uma imagem, mapa ou qualquer informação geográfica, é conseguir transformar suas coordenadas conhecidas em um determinado sistema de referência. Para conseguir fazer o georreferenciamento é preciso obter as coordenadas dos pontos de controle (locais que oferecem uma feição física perfeitamente identificável, como rios, represas, intersecções de estradas) do mapa a ser referenciado[...]” [16].

O projeto de software a ser descrito na seção V utiliza dados georreferenciados que são armazenados no banco de dados, onde, neste mesmo ambiente existem rotinas de geoprocessamento para calcular a localidade dos dispositivos que utilizarão o sistema.

4 TRABALHOS RELACIONADOS

Em Costa [1] foi desenvolvido um sistema para Windows em linguagem de programação Delphi 6 que faz a indexação de imagens. Primeiramente ele cadastra as imagens e o sistema faz a extração do histograma de cor de cada uma delas e salva em um banco de dados, neste sistema existe um menu onde é possível efetuar a comparação entre imagens para saber o quanto duas imagens são parecidas, através de percentuais. Esse sistema foi chamado de RICE, e a 0 ilustra o seu funcionamento.

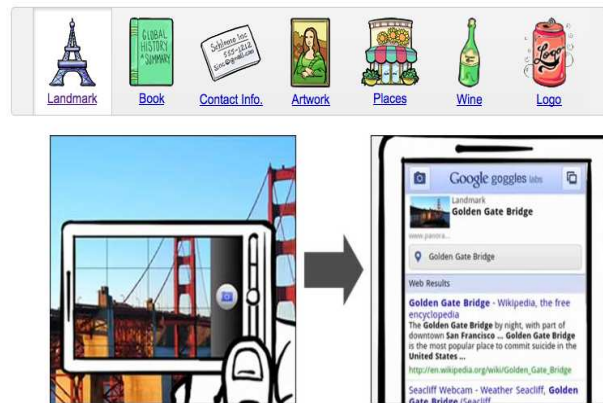
Figura 1 - Imagem do sistema RICE



Fonte: Costa, 2002

A empresa Google também desenvolveu uma aplicação, já voltada para dispositivos móveis com sistema operacional Android. Essa aplicação se chama Goggles, e sua funcionalidade está resumida em uma pessoa capturar uma imagem em um dispositivo móvel e o sistema procura em uma base de dados, que é mantida pelo fabricante, por imagens semelhantes, caso alguma imagem é encontrada, o sistema exibe o objeto encontrado, bem como os detalhes de onde este está localizado e algumas referências sobre o objeto. É um excelente sistema, responde rápido em sua utilização e encontra facilmente objetos bem reconhecidos no mundo todo, porém, muito abrangente, pois, procura qualquer imagem parecida em todo o mundo e os resultados já não se mostram satisfatórios para a solução do problema proposto neste projeto, conclusão esta obtida em testes efetuados, onde foi pego um objeto qualquer de um ambiente e o sistema apontou ser outro objeto nada parecido de um outro local do mundo. Conforme 0 é possível observar o funcionamento do sistema Google Goggles:

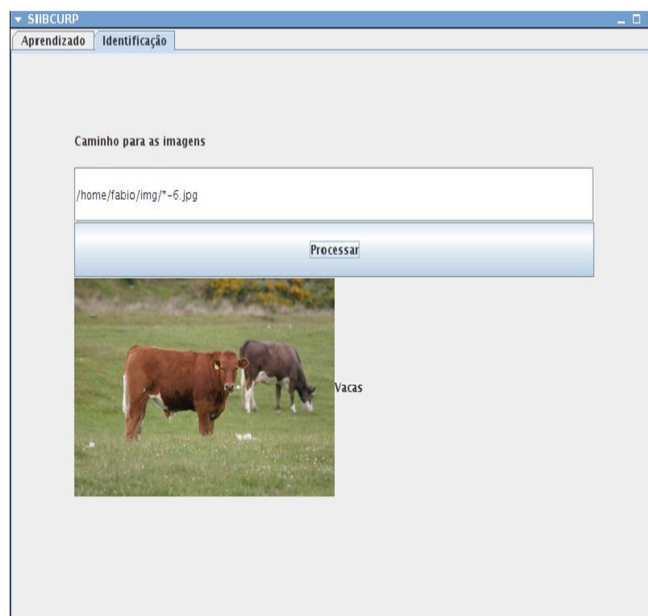
Figura 2 - Exemplo de funcionamento do sistema Google Goggles



Fonte: Clark, 2010

Outro sistema que pode ser citado é o que foi desenvolvido em Andrijauskas [5] para reconhecimento de padrões de imagens, chamado SIIBCURP. O SIIBCURP tem comportamento similar ao proposto neste projeto, porém, com utilização de redes neurais para identificar os padrões que devem ser reconhecidos dentro das imagens. Na 0 observa-se uma tela do sistema.

Figura 3 - SIIBCURP, sistema desenvolvido em Andrijauskas

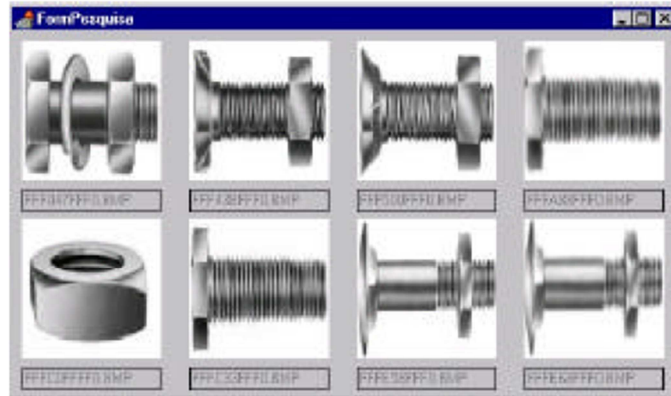


Fonte: Andrijauskas, 2007.

Em Rudek [7] também foi desenvolvido um sistema para indexação e reconhecimento de imagens com padrões conforme demonstração da 0, este sistema procura por imagens semelhantes através de vetores de comportamento, onde o autor selecionou uma imagem de

um parafuso e o sistema procurou na base de imagens por imagens que continham as mesmas características de vetor de comportamento da imagem original.

Figura 4 - Sistema desenvolvido em Rudek



Fonte: Rudek, 2001.

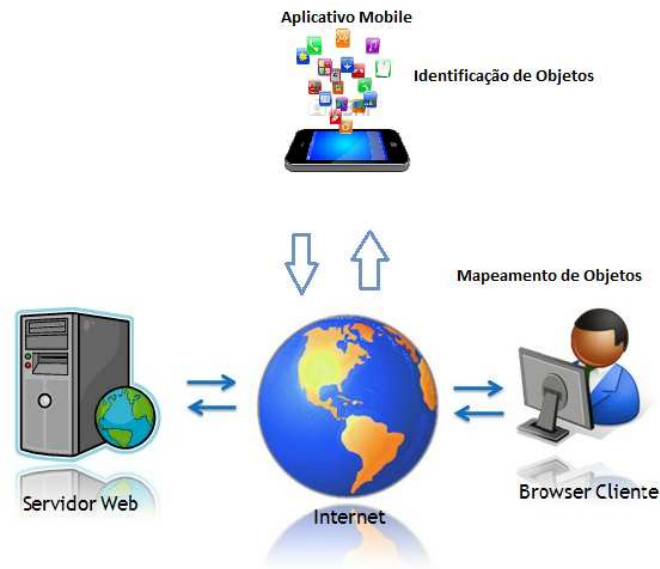
5 PESQUISA E DESENVOLVIMENTO: SIROAM

Levando em conta os projetos apresentados nos trabalhos relacionados este trabalho considera a possibilidade de contribuir com a disseminação da informação através da identificação de objetos em ambientes previamente mapeados utilizando o reconhecimento de imagens com técnicas de processamento de imagens. Este tipo de solução não é facilmente encontrado e, conforme as pesquisas e testes efetuados com os trabalhos relacionados, estes não possuem um resultado satisfatório. Isso se dá, por serem muito abrangentes, isto é, não possuem um mapeamento de ambiente específico ou por utilizarem algoritmos de comparação que não se encaixam nos modelo de comparação proposto neste trabalho, pois utilizam reconhecimento de padrões em imagens, como por exemplo, reconhecimento facial que detecta a face, porém, não a face específica de uma pessoa.

A identificação de objetos em ambientes previamente mapeados deve seguir um fluxo para a alimentação da base de dados e posteriormente o processamento de imagem propriamente dito, para seguir esse padrão o projeto de software será dividido em duas partes (plataformas) diferentes, com objetivos e características distintas e totalmente dependente uma da outra. Na primeira parte, teremos o ambiente Web onde será feito todo o mapeamento dos locais e objetos que serão a base de pesquisa para a identificação dos objetos, esta parte também será responsável pelo processamento de imagens, ou seja, a comparação de imagens. Na segunda parte, teremos o aplicativo final, utilizado pelos usuários para capturar a imagem

de um objeto, enviar ao ambiente Web e obter uma resposta sobre aquilo que está tentando identificar, a 0 ilustra estes ambientes e a comunicação entre eles.

Figura 5 - Ambiente web e aplicativo mobile e a comunicação entre eles



Fonte: Autoria própria, 2014.

Um aplicativo de reconhecimento de objetos — chamado de SIROAM (Sistema Integrado de Reconhecimento de Objetos em Ambientes Mapeados) — foi desenvolvido com objetivo de realizar a tarefa de reconhecimento de objetos em qualquer ambiente mapeado, onde, em primeiro lugar, um usuário do ambiente Web, que tem as devidas permissões de acesso, utilizará esta ferramenta para cadastrar o local dos objetos e posteriormente cada objeto que está contido naquele local. A inclusão dos locais é cadastro onde serão necessárias informações mais detalhadas e complexas de serem geradas e que serão descritas nas seções seguintes deste artigo. Os objetos também precisarão de informações detalhadas e que contribuirão para sua devida identificação. Em segundo lugar, na plataforma mobile, mais precisamente no sistema operacional Android, um usuário com o aplicativo instalado, poderá capturar uma foto de um objeto localizado em um ambiente que foi mapeado e descobrir informações completas sobre este objeto, também será possível obter informações do local onde se está, bem como, informações de objetos relacionados com o objeto localizado. O reconhecimento de objetos não será efetuado em locais que não foram mapeados, este tipo de ação não será possível realizar, pois, todo o reconhecimento de imagens é feito a partir de uma base específica contendo imagens georeferenciadas.

É importante salientar que para o desenvolvimento do projeto SIROAM foi necessário utilizar um Processo de Desenvolvimento de Software (PDS). A agilidade no desenvolvimento, arquiteturas utilizadas e tempo disponível foram os itens mais importantes para escolher um processo de desenvolvimento. Levando em consideração estes itens, foi selecionado o modelo de desenvolvimento ágil SCRUM. De acordo com [8], todos os itens para a escolha de um Processo de Desenvolvimento de Software e os demais que serão elencados ainda neste artigo devem ser escolhidos no início do projeto, evitando assim, problemas em seu desenvolvimento.

5.1 Processo de Desenvolvimento de Software com SCRUM

O SCRUM é uma metodologia de desenvolvimento ágil e flexível. Tem por objetivo definir um processo iterativo e incremental de desenvolvimento de produtos ou gestão de projetos. Produz um conjunto de funcionalidades potencialmente mais próximas do objetivo final no terminar de cada iteração (normalmente com uma duração de 30 dias). Centrado no trabalho em equipe, melhora a comunicação e maximiza a cooperação, permitindo que cada um faça o seu melhor e se sinta bem com o que faz, o que mais tarde se reflete em aumento de produtividade [9].

Conforme [9], SCRUM aplica-se a projetos tanto pequenos como grandes, esforçando-se para liberar o processo de quaisquer barreiras, o seu principal objetivo é conseguir uma avaliação correta do ambiente em evolução, adaptando-se constantemente ao “caos” de interesses e necessidades, indicado e utilizado para o desenvolvimento de softwares em ambientes complexos, onde os requisitos mudam com certa frequência, sendo o caminho utilizado para aumentar produtividade nesses tipos de sistemas.

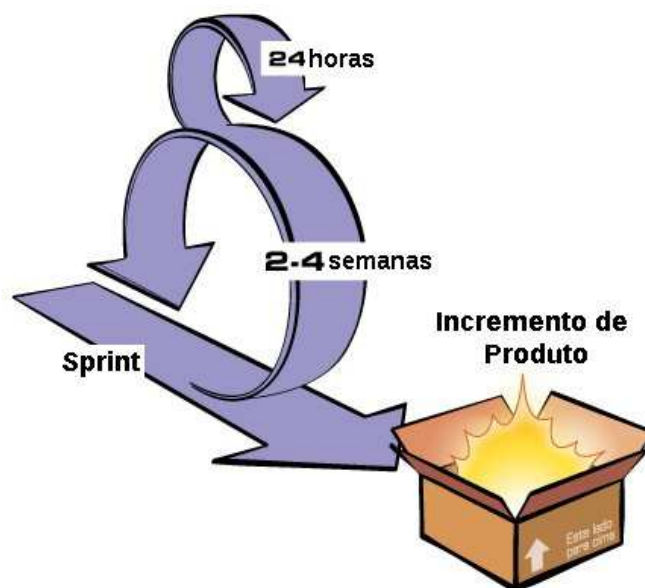
O processo SCRUM possui um vocabulário específico que precisa ser conhecido para entender como foi dado o desenvolvimento do SIROAM. Assim serão elencados os itens deste vocabulário [9], junto com uma explicação de sua utilização:

- BackLog – Lista de todas as funcionalidades a serem desenvolvidas durante o projeto completo, sendo bem definido e detalhado no início do trabalho, deve ser listado e ordenado por prioridade de execução;
- Sprint – Período não superior a 30 dias, onde o projeto (ou apenas algumas funcionalidades) é desenvolvido;
- Sprint Planning Meeting – Reunião de planejamento;

- Sprint Goal – Disparo dos objetivos/metast;
- Sprint Review – Reunião de revisão.

Seguindo a linha de utilização do SCRUM, na figura abaixo é possível observar o ciclo do Sprint, evento do SCRUM que foi o mais utilizado durante o desenvolvimento do SIROAM.

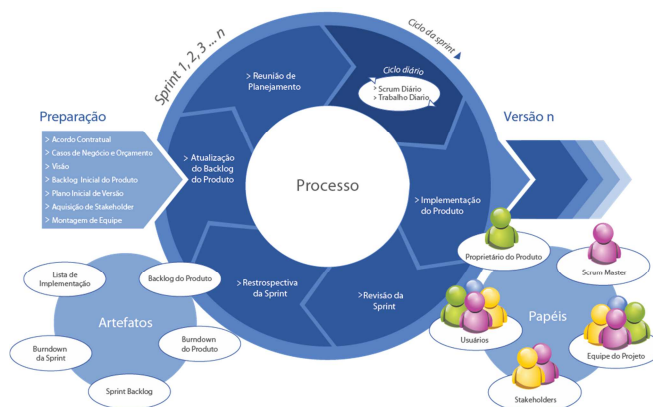
Figura 6 - Ciclo do Sprint



Fonte: Devin, 2009

Para ilustrar melhor todo o processo SCRUM, é apresentada a imagem abaixo que simboliza todo o ciclo do processo:

Figura 7 - Processo Scrum



Fonte: 2Gather Tecnologia, 2014

Foram adotados Sprint's de duas semanas, onde ao fim de cada Sprint foi entregue uma parte do desenvolvimento do SIROAM.

O SCRUM é bem amplo, porém, como este trabalho foi desenvolvido por uma pessoa só, algumas cerimônias não foram necessárias e no lugar delas, foram realizadas reuniões com a orientadora do trabalho.

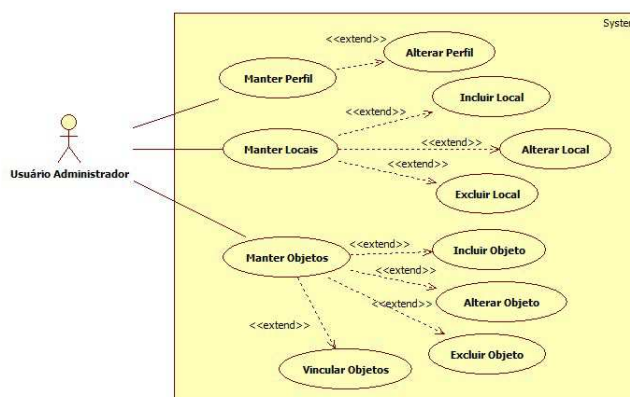
O SIROAM é uma ferramenta de software para a Web e Mobile (Android), desenvolvido com padrões de Análise e Projeto Orientado a Objetos, utilizando-se a metodologia SCRUM. Não foi necessário desenvolvimento de uma documentação específica, conforme necessário em outros padrões, porém, para definir o BackLog de uma maneira bem clara, foram elencados todos os requisitos necessários e também todos os casos de uso, juntamente com seus diagramas. Por se tratar de orientação a objetos, onde é possível representar todos os atores e artefatos do sistema, foi utilizada a Linguagem de Modelagem Unificada, conhecida como UML. Após a escolha das metodologias e ferramentas, foi iniciado o desenvolvimento do SIROAM, organizados em Sprint de duas semanas, para o seu desenvolvimento foi necessário sete Sprint's, totalizando mais de três meses de pesquisas e estudos, contando com apenas um desenvolvedor para analisar, codificar e testar todos os módulos desenvolvidos.

Para o desenvolvimento do SIROAM foram levantados uma série de requisitos. Em primeiro lugar, no Sprint1, elencou-se os requisitos funcionais necessários para o desenvolvimento da solução no ambiente Web, que compreende os cadastros:

- R001 – Alterar perfil;
- R002 – Incluir local;
- R003 – Alterar local;
- R004 – Excluir local;
- R005 – Incluir objeto;
- R006 – Alterar objeto;
- R007 – Vincular objetos;
- R008 – Excluir objetos;

Baseado nesses requisitos, foram criadas três classes, sendo elas: Usuario, Local e Objeto. Cada uma dessas classes representa fielmente a modelagem de banco de dados que é apresentada nas próximas seções deste artigo. Os requisitos funcionais apresentados acima estão representados graficamente por diagrama de caso de uso, conforme a figura abaixo:

Figura 8 - Diagrama de Caso de Uso: Cadastros do ambiente web



Fonte: Autoria própria, 2014.

Dando continuidade ao projeto, foram criados os cadastros correspondentes ao diagrama, esses foram divididos na seguinte forma:

- Cadastro de locais – O usuário administrador inclui, altera e exclui locais (coordenadas geográficas);
- Cadastro de objetos – O usuário administrador inclui, altera, vincula e exclui objetos;
- Perfil – O usuário administrador do sistema altera seu perfil, no ambiente web só existe o usuário administrador.

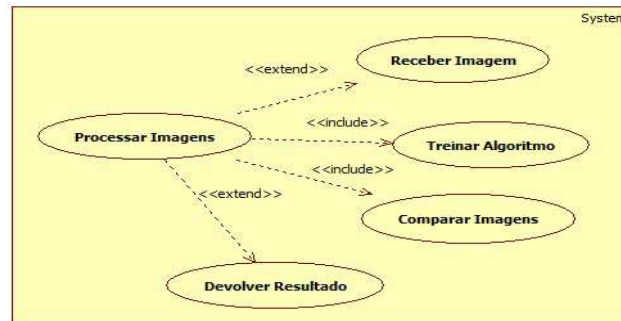
Em sequência, concluídos e testados os cadastros, finalizou-se o Sprint1.

No Sprint2 foram elencados os requisitos funcionais das tarefas a serem executadas pelo sistema, na mesma forma que os cadastros, essas tarefas foram as seguintes:

- R009 – Receber imagem;
- R010 – Treinar algoritmo;
- R011 – Comparar imagens;
- R012 – Devolver resultado;

Como pode-se perceber, esse Sprint 2 foi destinado ao desenvolvimento da atividade principal do projeto, que é receber uma imagem e compará-la com as imagens existentes e devolver um resultado com a imagem mais semelhante à imagem enviada.

Figura 9 - Diagrama de Caso de Uso: Processamento de imagens



Fonte: Autoria própria, 2014.

A Sprint2 finalizou-se com o processamento de imagens funcionando e pronto para ser integrado ao sistema Web, ação essa que, será detalhada nas próximas seções deste artigo.

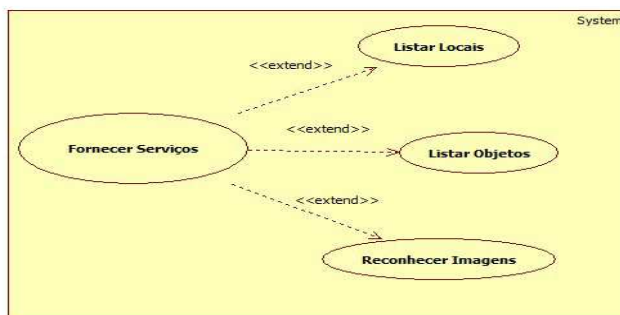
Continuando com o fluxo de trabalho, deu-se início ao Sprint 3. Esse Sprint foi destinado à integração do serviço de processamento de imagens com o ambiente Web. Consistiu-se basicamente em criar no ambiente Web um webservice que recebe a imagem de qualquer outro aplicativo, independente de linguagem, plataforma e recurso de hardware e em fazer a comunicação desse webservice com o caso de uso Processar Imagens. Também foram criados nesse serviço métodos que retornam o(s) local(is) que um dispositivo se encontra, baseando-se em coordenadas geográficas enviadas ao serviço.

A seguir será detalhado cada método que foi criado e um resumo sobre seu objetivo no projeto:

- Método: List<LocalModel> ListaLocaisLL(String latitude, String longitude) – Método que recebe a latitude e longitude atual de um dispositivo e retorna uma lista de locais onde ele está, é uma lista pois, pode haver uma interseção de polígonos.
- Método: List<ObjetoModel> ListaObjetos(String objid) – Método que lista os objetos relacionados de um determinado objeto, recebe por parâmetro o id do objeto que se quer pesquisar seus objetos relacionados.
- Página: ReconheceImagem.aspx, recebe por parâmetro o Stream de uma foto e também a latitude e longitude atual do dispositivo. Esta página se comunica com o caso de uso Processar Imagens e retorna, se encontrado, o objeto localizado.

O diagrama de caso de uso a seguir exemplifica graficamente o que foi desenvolvido:

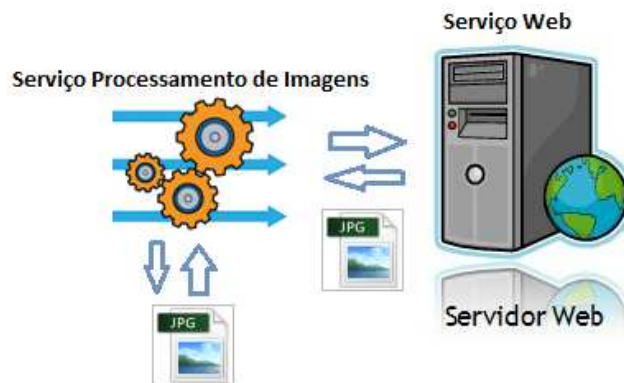
Figura 10 - Diagrama de Caso de Uso: Fornecer serviços



Fonte: Autoria própria, 2014.

Estas foram as implementações realizadas no Sprint 3. Nesta fase do projeto, não foi possível fazer testes precisos, pois, dependia das próximas implementações para que os serviços recebam os dados de forma correta para efetuar todo o processamento das requisições web. Para demonstrar graficamente como foi efetuada a integração entre o ambiente web e o serviço de processamento, foi desenvolvida a ilustração abaixo:

Figura 11 - Integração entre ambiente web e serviço de processamento de imagem



Fonte: Autoria própria, 2014.

A 0 mostra que há uma separação entre os ambientes, pois, o ambiente de processamento de imagens é separado do ambiente web e a comunicação entre ambos ocorre via serviços que o ambiente de processamento de imagens fornece.

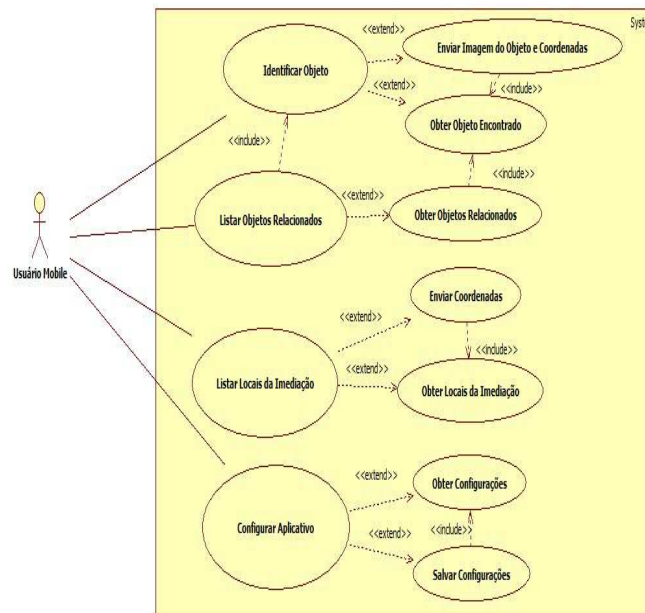
O Sprint 4 foi destinado para o desenvolvimento da aplicação mobile, integração com o ambiente web e para os testes iniciais das ferramentas. Para realizar o Sprint 4, foram elencados os seguintes requisitos funcionais:

- R013 – Identificar objetos (identificação);
- R014 – Listar objetos relacionados (objetos relacionados);

- R015 – Listar locais (imediações);
- R016 – Configurar aplicação (configurações);

Cada requisito funcional levantado nesse Sprint se tornou diretamente um menu no sistema mobile, cujo cada recurso será apresentado e descrito nas próximas seções deste trabalho. Para auxiliar no desenvolvimento destes requisitos, foi criado o diagrama de caso de uso representado abaixo:

Figura 12 - Diagrama de Caso de Uso: Aplicativo Mobile



Fonte: Autoria própria, 2014.

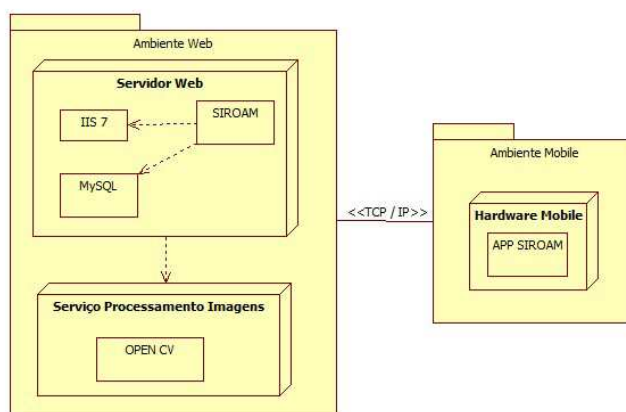
Como indicado no Sprint 3, a comunicação entre o ambiente móvel e o ambiente Web se dará através de serviços. Cada classe criada no ambiente móvel será instanciada e seus dados serão provenientes das chamadas aos métodos do Web Service. Para atender os casos de uso acima, foram geradas três classes que representam a informação que trafega entre os ambientes - LatLng, ObjetoModel e LocalModel - cada classe possui os atributos dos objetos reais, como, por exemplo, descrição, foto e nome.

Concluindo o projeto entrou-se na Sprint 5, que foi destinada a ajustes e diversos testes com o aplicativo. Durante este último Sprint foi encontrado uma série de erros que foram corrigidos para que os testes fossem realizados corretamente. O Sprint 5 se encerrou com a aplicação pronta para ser testada e avaliada, sem ocorrência de erros de programação ou falhas nos algoritmos.

5.2 Arquitetura

O SIROAM, como já foi visto, é um projeto que integra várias plataformas, possui código fonte gerado para duas plataformas distintas em duas linguagens de programação diferentes. Para exemplificar graficamente esta integração, foi necessário desenvolver o diagrama de implementação, apresentado na figura a seguir:

Figura 13 - Diagrama de Implementação



Fonte: Autoria própria, 2014.

Com a visão da 0, é possível ver a relação entre os ambientes e também a comunicação que há entre eles. Nesta seção do artigo, serão apresentados todos os ambientes, suas características técnicas e ferramentas e tecnologias utilizadas para a construção do projeto. Esta apresentação se dará na seguinte sequência: Serviço Processamento de Imagens, Servidor Web e Aplicativo Mobile. Esta sequência é a mesma em que todo o projeto foi desenvolvido, pois, uma implementação depende fortemente da outra para funcionar.

5.3 Arquitetura do Serviço de Processamento de Imagens

Para o Serviço de Processamento de Imagens foi escolhida a ferramenta Microsoft Visual Studio 2012, por se tratar de uma ferramenta amplamente utilizada no mercado e por ser fornecida gratuitamente para estudantes. A linguagem de programação padrão escolhida foi C#, uma das linguagens mais utilizadas no mercado atualmente conforme [10]: “Java, C++ e Ruby seguem a lista, embora C# e C são os que mais estão crescendo”. Após a escolha das linguagens e ferramentas foi feito uma pesquisa por componentes de processamento de

imagens, mais especificamente, componentes que efetuam a comparação de imagens. No mercado existe uma série de componentes que fazem esta tarefa, porém, era preciso uma solução de software livre e sem custos, seguindo essas premissas, foi encontrada a biblioteca OpenCV. Conforme [11] a OpenCV é uma biblioteca de software liberada para uso acadêmico e comercial, possui interfaces de C++, C, Python e Java e suporta Windows, Linux, Mac OS, iOS e Android. OpenCV foi projetado para a eficiência computacional e com um forte foco em aplicações de tempo real. Escrito e otimizado para C/C++, a biblioteca pode tirar proveito do processamento multi-core. Habilitado com OpenCL, ele pode tirar proveito da aceleração da plataforma de computação heterogênea subjacente de hardware. Adotado em todo o mundo, OpenCV tem mais de 47 mil pessoas em sua comunidade de usuários e um número estimado de download superior a 7 milhões.

Para haver uma integração da biblioteca OpenCV com a arquitetura de desenvolvimento escolhida, seria necessário um grande tempo de desenvolvimento para criar uma nova biblioteca com a função de adaptar as duas plataformas, porém, isto não foi necessário, pois, já existe no mercado esta solução, que por sua vez, também é de código aberto e é conhecida por EmguCV, conforme [12] a EmguCV é uma biblioteca adaptadora de .NET para a biblioteca OpenCV, permitindo que as funções da OpenCV sejam chamadas na plataforma .NET e é compatível com diversas linguagens, por exemplo: C#, VB.NET, VC++ e IronPython. Foram realizados alguns testes de utilização da biblioteca EmguCV e todos os resultados foram positivos. Com base nos testes positivos, foi criado o projeto do serviço de reconhecimento de imagens, este serviço foi criado utilizando projeto com arquitetura WCF (Windows Communication Foundation) que por sua vez, isola o processamento e as dependências de bibliotecas que esse serviço utiliza das plataformas que vão consumi-lo, neste caso a plataforma web. Este serviço possui apenas um método, que foi chamado de `ComparaImagens`.

O método `ComparaImagens` recebe um objeto chamado de `ObjetoComparacaoModel` que possui encapsulado a imagem do objeto que se quer encontrar e uma lista de imagens que se quer comparar. Este método instancia a biblioteca EmguCV e chama o método que treina o algoritmo para, em sequência, efetuar a comparação. Após efetuada a comparação, ele retorna a lista de imagens que foram comparadas com a quantidade de pontos de semelhança entre a imagem do objeto e a imagem que foi comparada. Abaixo será demonstrado o trecho de código correspondente ao que foi explicado.

Figura 14 - Código fonte do serviço de processamento de imagens

```

public ObjetoComparacaoModel ComparaImagens(ObjetoComparacaoModel obj)
{
    return Match(obj);
}

public ObjetoComparacaoModel Match(ObjetoComparacaoModel obj)
{
    string[] dbImages = new string[obj.Imagens_Servidor.Count];

    for (int i = 0; i < obj.Imagens_Servidor.Count; i++)
    {
        dbImages[i] = obj.Imagens_Servidor[i].Imagem_Servidor;
    }

    string queryImage = obj.Imagem_Objeto;

    IList<IndicesMapping> imap;

    // computa os descritores de cada imagem
    var dbDescsList = ComputeMultipleDescriptors(dbImages, out imap);

    // concatena todas as imagens do banco de dados dentro de uma unica matriz
    Matrix<float> dbDescs = ConcatDescriptors(dbDescsList);

    // computa os descritores da imagem que será comparada
    Matrix<float> queryDescriptors = ComputeSingleDescriptors(queryImage);

    FindMatches(dbDescs, queryDescriptors, ref imap);

    foreach (IndicesMapping map in imap)
    {
        foreach (ObjetoModel oi in obj.Imagens_Servidor)
        {
            if (oi.Imagem_Servidor == map.fileName)
            {
                oi.Percentual_Igualdade = map.Similarity;
            }
        }
    }

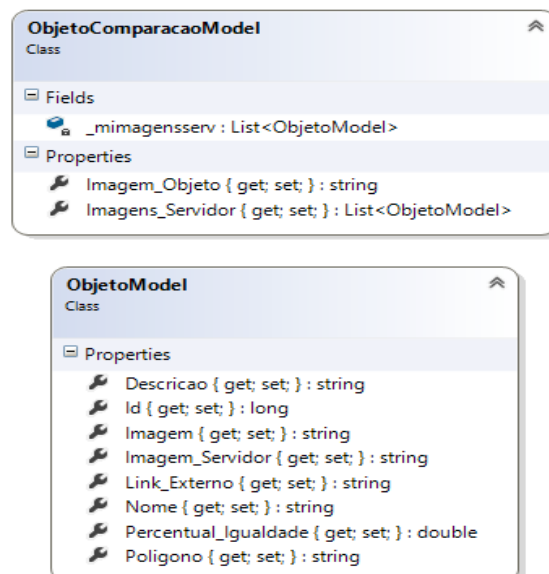
    return obj;
}

```

Fonte: Autoria própria, 2014.

As classes que são utilizadas na comunicação entre os serviços possuem a seguinte representação gráfica:

Figura 15 - Classes utilizadas no serviço de processamento de imagens



Fonte: Autoria própria, 2014.

O algoritmo da EmguCV que faz a comparação das imagens é conhecido por SURF (Speeded Up Robust Features) que faz buscas em uma imagem por um determinado objeto utilizando as técnicas de histograma de cores. O algoritmo SURF, efetua a comparação e pode, se necessário, retornar uma imagem demonstrando como a comparação foi feita. Para exemplificar esta funcionalidade, observe a 0 onde é exibida a imagem normal e o objeto de procura e qual foi o resultado obtido.

Figura 16 - Exemplo de comparação de imagens efetuada pelo SURF em um objeto na geladeira



Fonte: Aatoria própria, 2014.

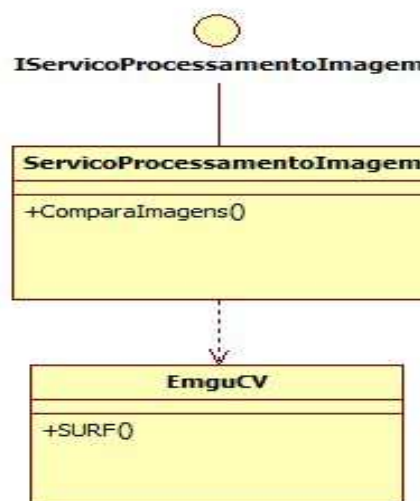
A imagem da 0 foi obtida através de testes preliminares para a escolha da biblioteca e o retângulo vermelho representa o objeto encontrado, a foto do lado direito é a imagem do objeto capturada previamente e catalogada no sistema e a imagem da esquerda é a foto capturada por um dispositivo móvel para ser comparada com a imagem catalogada. Contudo a implementação do serviço de processamento de imagens não necessita dar um retorno visual de como a comparação foi feita e sim o retorno em números, conforme já citado anteriormente, cada imagem comparada receberá um número de pontos de semelhança, que, significa, olhando a Figura 15 cada linha de semelhança entre as duas imagens. A partir deste recurso, pode-se desenvolver diversos métodos de comparação, inclusive, definindo um número mínimo de pontos de semelhança para que a comparação efetuada seja aceita.

Para tornar estas comparações mais rápidas, a biblioteca OpenCV utiliza, quando disponíveis, os recursos da GPU da placa de vídeo para efetuar o processamento das imagens, conforme citado em [13] no ano de 2010 foi adicionado na OpenVC um módulo que utiliza a aceleração de hardware da GPU, sem a necessidade dos usuários, neste caso, programadores, fazer treinamentos sobre o funcionamento da mesma. Segundo pesquisa efetuada em [13], utilizando GPU a OpenCV tem um ganho de trinta vezes mais rápido no processamento de imagens, e, para o algoritmo SURF, doze vezes mais rápido.

Com a utilização da arquitetura WCF da Microsoft, este projeto ficou isolado do processo web, permitindo assim, se disponível, a utilização da GPU para o processamento no algoritmo SURF, lembrando que, conforme documentação disponível em [12] e também citado em [13], esta utilização é transparente ao programador.

Para desenvolver o serviço foi desenhado o diagrama de classes representado na 0, que mostra, de uma forma simplificada a implementação desta parte do projeto.

Figura 17 - Implementação do serviço de processamento de imagens



Fonte: Autoria própria, 2014.

Esta é a estrutura do serviço, olhando de um ponto mais abrangente, as operações executadas por ele são:

- Receber a imagem de comparação e a lista de imagens para comparar;
- abrir binariamente cada imagem de comparação para treinar o algoritmo SURF;

- efetua a comparação, dando a cada imagem uma quantidade de pontos de semelhança;
- retorna os resultados obtidos para que as outras aplicações utilizem.

O próximo passo será o ambiente Web consumir este serviço.

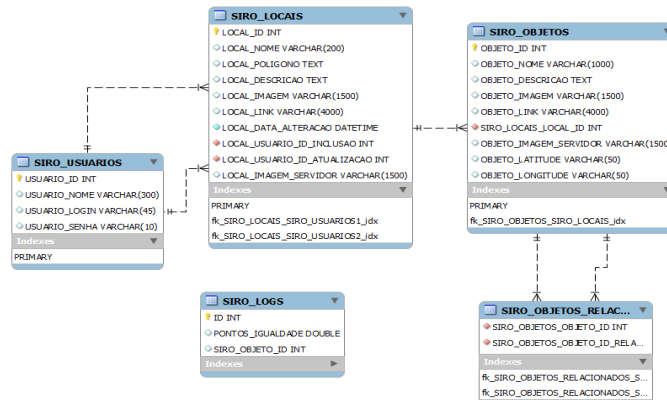
5.4 Arquitetura do Ambiente Web / Servidor Web

Para o ambiente Web foram utilizadas as mesmas ferramentas de desenvolvimento do serviço de processamento de imagens, o que difere um do outro é o tipo de projeto. Para atender esse requisito foi criado um novo projeto do tipo WebSite no Microsoft Visual Studio 2012. Como é um ambiente Web, foi escolhido o Microsoft ASP.NET como arquitetura principal de desenvolvimento e a linguagem de programação C#. Para este ambiente, também foi elencado um padrão de layout, para que todos os módulos desenvolvidos estejam em conformidade e que possam ser acessados tanto por computadores quanto por dispositivos móveis.

Atendendo a necessidade levantada, escolheu-se o framework de desenvolvimento de layout bootstrap, fundado pela empresa Twitter, conforme em [14] é uma coleção de ferramentas, padrões e temas de desenvolvimento em HTML5 para facilitar o desenvolvimento de aplicações Web, aumentando a produtividade e permitindo ao desenvolvedor criar aplicações responsivas, em suma, que se adaptam a qualquer dispositivo que acessar, permitindo o redimensionamento dos componentes de tela.

Em continuidade, foi elencado como banco de dados relacional o MySQL para ser utilizado com o objetivo de guardar as informações necessárias do projeto. Estas informações são espelho do levantamento de requisitos efetuado anteriormente neste artigo. Cada tabela deste banco, representa uma classe do sistema, obedecendo as características de orientação à objetos do mesmo. Será apresentado em sequência o diagrama de banco de dados e a explicação de cada tabela e qual o seu objetivo no sistema.

Figura 18 - Diagrama de banco de dados do siroam



Fonte: Autoria própria, 2014.

Conforme apresentado na 0 o SIROAM possui em seu modelo de dados cinco tabelas, cuja suas aplicações são descritas:

- SIRO_USUARIOS – Tabela responsável por armazenar as informações dos usuários que terão acesso ao sistema.
- SIRO_LOCAIS – Tabela responsável por armazenar os dados de cada local mapeado, no fluxo de cadastro do sistema essa é a primeira tabela que deve ser preenchida após fazer login.
- SIRO_OBJETOS – Tabela que armazena as informações de todos os objetos mapeados de um local, possui dependência lógica de registros existentes na tabela SIRO_LOCAIS.
- SIRO_OBJETOS_RELACIONADOS – Tabela que armazena quais objetos estão relacionados com um determinado objeto, característica essa que será apresentada na demonstração de funcionalidades do projeto neste artigo, possui dependência lógica de registros existentes na tabela SIRO_OBJETOS.
- SIRO_LOGS – Tabela que armazena informações de cada execução de reconhecimento de imagem efetuada.

Como é notável, a estrutura necessária para atender as necessidades do projeto não são complexas, mas, de suma importância para se realizar todo o mapeamento de um local.

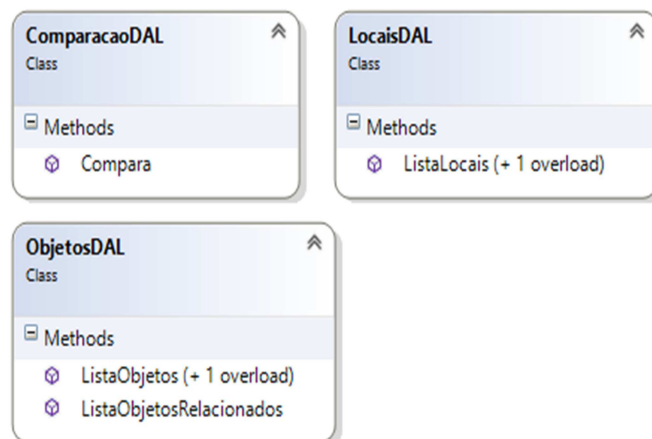
Foi convencionado que, para mapear um local, é necessário, ao incluir um local no SIROAM, enviar juntamente com seus dados um arquivo kml (Keyhole Markup Language) que possui em seu conteúdo o polígono de um determinado local em formato xml padronizado pelas ferramentas de geoprocessamento, com isso, fez-se necessário escolher uma biblioteca que recebe o conteúdo deste arquivo e extrai, em formato de string as

coordenadas geográficas, que por sua vez, serão armazenadas na tabela SIRO_LOCAIS para futuras consultas geoespaciais. Para efetuar tal operação houve a necessidade de localizar uma biblioteca que já possui esta implementação por ser algo trivial, não haveria necessidade de desenvolver tais métodos. Foram efetuados alguns testes de bibliotecas open source e dentre as encontradas, a escolhida foi a SharpKML, biblioteca essa desenvolvida em C#, de código aberto e que possui métodos que manipulam arquivos kml facilmente. Com o intuito de utilizar esta biblioteca foi criada uma classe chamada Geoprocessamento que tem um método nomeado GetPontosPoligono que recebe por parâmetro todo o xml do arquivo kml e retorna para a aplicação apenas uma string com todas as coordenadas geográficas de um polígono contido neste xml. Com isto a necessidade de se trabalhar com polígonos foi atendida satisfatoriamente.

Continuando com a arquitetura do ambiente Web, foi utilizado com servidor de aplicação o Microsoft IIS7, que, por sua vez, pode ser instalado em qualquer computador com Microsoft Windows sem a necessidade de pagar por licença ou direitos de uso.

Em nível de projeto o ambiente Web foi dividido em três camadas, a primeira delas é responsável pelo modelo de objetos que são manipulados pelo sistema, apresenta a estrutura básica das classes do projeto, como por exemplo: Locais, Objetos. A segunda camada trata das regras de negócio, faz a persistência entre o banco de dados e o objeto instanciado, permitindo que cada objeto seja carregado com os dados que estão armazenados no banco de dados, a terceira camada, chamada camada de apresentação, é a aplicação em si que será apresentada nas próximas sessões deste artigo.

Figura 19 - Diagrama de classes da segunda camada da aplicação web



Fonte: Autoria própria, 2014.

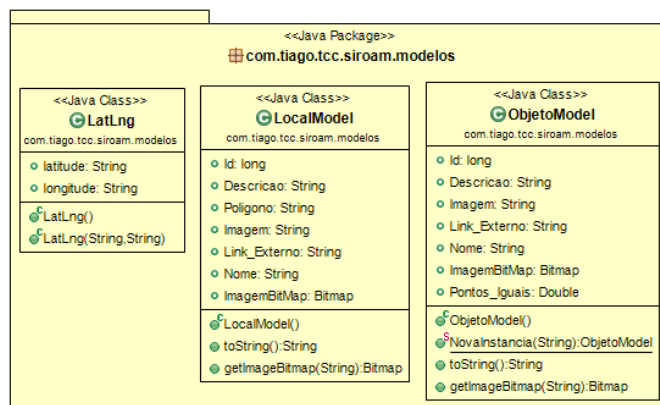
A 0 representa o diagrama de classes da segunda camada implementada no SIROAM e com ela pode-se notar que cada classe manipula um objeto específico.

5.5 Arquitetura do Ambiente Mobile / Android

Para o desenvolvimento do sistema mobile foi necessário a escolha de uma ferramenta diferente, pois, o Microsoft Visual Studio não foi projetado para desenvolvimento de sistemas para Android. O que motivou a escolha do Android como plataforma para o ambiente mobile foi a fácil localização de códigos de exemplo, ferramentas de desenvolvimento open source e também a constante evolução que este sistema operacional vem sofrendo. Levando em conta as características levantadas acima, foi escolhido como plataforma de desenvolvimento o Eclipse e linguagem padrão para desenvolvimento Android Java.

Foi criado no eclipse um novo projeto, também nomeado SIROAM, e, para atender a comunicação deste ambiente com o ambiente Web, foram criadas classes específicas, conforme citadas anteriormente nos requisitos levantados, essas classes representam as propriedades essenciais para locais e objetos. Também foram criadas classes adaptadoras, que se comunicam com os serviços Web criados e instanciam essas classes com os dados retornados das operações efetuadas no servidor. Para representar as classes criadas neste ambiente, foi criado o diagrama de classes para cada pacote desenvolvido, conforme a 0 é apresentado o diagrama de classes das classes modelos.

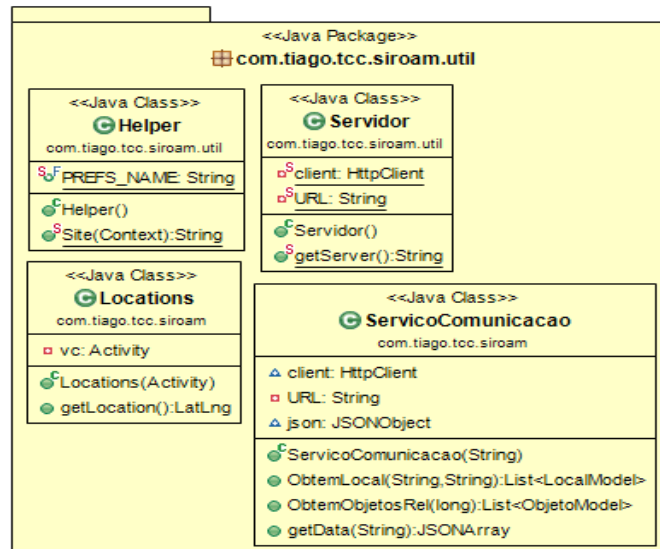
Figura 20 - Diagrama de classes do pacote de modelos do aplicativo mobile



Fonte: Autoria própria, 2014.

As classes adaptadoras possuem o seguinte diagrama de classes:

Figura 21 - Diagrama de classes do pacote de classes adaptadoras aplicativo mobile



Fonte: Autoria própria, 2014.

A 0 apresenta o diagrama de classes das classes adaptadoras que são utilizadas em diversas partes do sistema mobile e que basicamente fazem as comunicações entre os serviços. O sistema mobile também foi dividido em camadas, sendo a primeira camada o pacote que possui todas as classes modelo apresentado pela 0. A segunda camada é representada pelas classes adaptadoras, conforme diagrama de classe exibido na 0 e a terceira camada é a de apresentação, sendo esta a parte visível para o usuário final e que será apresentada nas próximas seções deste artigo.

5.6 SIROAM – Web

O software SIROAM desenvolvido para este trabalho tem caráter inicial de experimento para que os testes de avaliação do reconhecimento de imagens possam ser efetuados. Contudo, desde sua concepção, foram pensadas medidas de segurança no seu acesso, com isso, o SIROAM, possui, conforme citado anteriormente um controle de usuários que podem acessá-lo, permitindo assim, que o mesmo possua uma tela de acesso inicial, onde somente pessoas com um usuário e senha válidos consigam entrar no sistema.

Figura 22 - Janela de autenticação do SIROAM

Bem vindo ao SIROAM

Entre com seu usuário e senha

[Login](#)

Fonte: Autoria própria, 2014.

Na 0 o usuário faz o acesso ao sistema colocando o seu usuário e senha. Este acesso é simples e não tem diferenciação de perfil para cada usuário, ou seja, o menu principal carregado após o acesso é o mesmo para qualquer usuário.

Figura 23 - Janela principal do SIROAM

SIROAM

SIROAM / Página Inicial

A SIROAM

SIROAM Entenda o projeto

Cadastre os locais/ambientes

Primeiramente inclua todos os locais (ambientes) possíveis, antes de incluí-los, você deve criar o polígono do local, para isto, pode ser utilizado o Google Earth. É importante incluir o polígono do local para que as consultas geoespaciais consigam identificar onde você está em determinado momento. É interessante também colocar uma foto do local e uma descrição completa do mesmo, para que todos possam conhecer melhor o ambiente.

Cadastre os objetos

Após cadastrar os locais, cadastre os objetos. Se possível coloque a coordenada onde aquele objeto se encontra e também, cadastre obrigatoriamente uma foto de boa qualidade deste objeto. As identificações do sistema ocorrerão com base na foto cadastrada. Não esqueça também de colocar uma boa descrição sobre o objeto. Quanto mais informações forem preenchidas, melhor será para o usuário final do sistema.

Fonte: Autoria própria, 2014.

Acessando o SIROAM o usuário receberá uma janela com algumas dicas e explicações sobre o projeto, bem como um menu inicial, apresentado na 0. O usuário terá a opção de alterar o seu nome e senha clicando no item perfil. Este ambiente do projeto, tem por objetivo principal, permitir que o usuário capture fotos de locais e objetos e posteriormente acesse esse recurso para alimentar a base de dados com as informações pertinentes, então, existem dois módulos desenvolvidos que são os principais e são fruto do levantamento de requisitos efetuado no início da análise deste projeto. Esses módulos foram incluídos no sistema Web, no menu cadastros e são chamados de Locais e Objetos.

O módulo de cadastro de locais é apresentado ao usuário assim que clicado no item Locais do menu e em seguida o sistema apresenta a tela, solicitando os dados de um novo local e abaixo uma lista dos locais já cadastrados, permitindo assim a alteração ou exclusão de qualquer um deles.

Figura 24 - Cadastro de Locais no SIROAM

The screenshot shows a web interface for managing locations. The main form has the following fields:

- Nome:** A text input field.
- Polígono:** A field with a 'Nenhum arqui...' placeholder and a 'Procurar' button.
- Foto:** A field with a 'Nenhum arqui...' placeholder and a 'Procurar' button.
- Link Externo:** A text input field.
- Descrição:** A large text area.

 Below the form are two buttons: 'Novo Local' (blue) and 'Salvar' (blue). At the bottom, there is a table with the following structure:

Local	Ações
Casa	Alterar Excluir

Fonte: Autoria própria, 2014.

Conforme pode-se ver na 0 pode ser inserido para um local uma foto e o arquivo kml que representa o polígono deste local, é obrigatório a inclusão deste polígono, pois, este favorece no refinamento das pesquisas por imagens, onde o sistema pesquisará por objetos que estejam contidos no polígono do local em que a coordenada geográfica do dispositivo móvel se encontra.

Após incluído um local, o usuário está apto a incluir vários objetos para este local, para fazer isto, basta ele clicar no menu Objetos do sistema, que exibirá a tela solicitando os dados do objeto que vai ser incluído.

Figura 25 - Cadastro de Objetos no SIROAM

The image shows a web form titled 'Objetos'. It contains the following elements: a dropdown menu for 'Local' with 'Unilasalle' selected; text input fields for 'Nome', 'Latitude', and 'Longitude'; a 'Foto' section with a text input containing 'Nenhuma imagem...' and a 'Procurar' button; a 'Link Externo' text input; a large text area for 'Descrição'; and a 'Objetos Relacionados' dropdown menu with the text 'Selecione algumas opções'. At the bottom of the form are two buttons: 'Novo Objeto' and 'Salvar'.

Fonte: A autoria própria, 2014.

Para o cadastro de um objeto é diferente. É preciso selecionar o local e preencher a latitude e longitude deste objeto. Este dado de geolocalização embora não seja, obrigatório, é importante para ajudar nas pesquisas. O item do cadastro que é indispensável e não pode ser esquecido é a imagem do objeto, esta é a base para todo o funcionamento da identificação de imagem. Cadastrar um objeto sem por a sua imagem é uma opção que não deve ser considerada.

Os passos apresentados acima são os essenciais para que um ambiente seja previamente mapeado e são requisitos obrigatórios para o bom funcionamento do ambiente mobile que será apresentado em seguida.

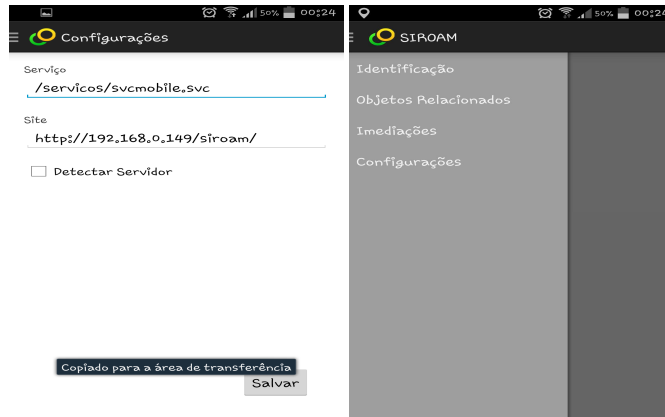
5.7 SIROAM – Mobile

O SIROAM Mobile também possui caráter experimental e foi desenvolvido para rodar em dispositivos móveis com sistema operacional Android 2.3 ou superior e que possuam GPS e Câmera. Este aplicativo é a camada de apresentação final do projeto e que comprova o funcionamento do algoritmo e que permite a avaliação do mesmo.

Ao acessar o aplicativo, o mesmo já abre ao usuário a janela da câmera para que o mesmo possa identificar algum objeto. Antes de apresentar o funcionamento da identificação serão apresentadas algumas janelas que funcionam e precisam ser acessadas antes de utilizá-lo.

As primeiras características a serem apresentadas são o menu do sistema e o menu configurações:

Figura 26 - Na esquerda o menu do sistema e na direita as configurações



Fonte: Autoria própria, 2014.

Acessando o menu configurações o usuário pode configurar o endereço dos serviços, estes endereços vem pré-configurados, porém, se necessário, eles podem ser alterados.

O segundo menu a ser apresentado é o imediações, este módulo obtém as coordenadas geográficas atuais do dispositivo e localiza em qual ou quais locais este dispositivo se encontra, vindo de encontro com a funcionalidade do cadastro de incluir o polígono do local, só assim o sistema poderá descobrir onde se encontra o dispositivo.

Figura 27 - Na esquerda o menu imediações e na direita o item casa acessado

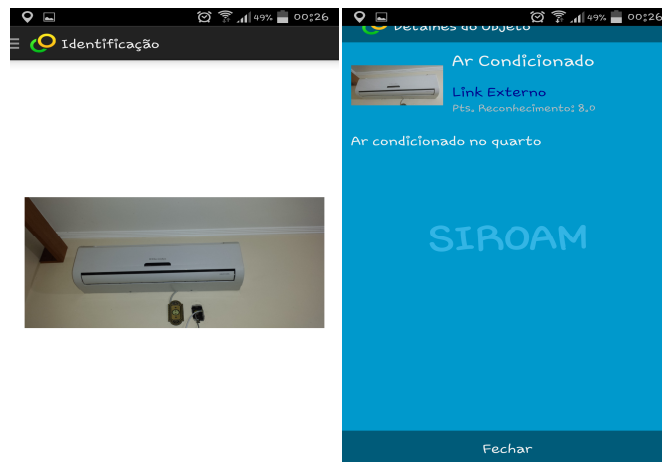


Fonte: Autoria própria, 2014.

Quando um ou mais locais onde o dispositivo se encontra é localizado os mesmos são listados e o usuário pode clicar sobre o local e obter as informações deste local conforme lado direito da 0.

Finalizando a apresentação do SIROAM – Mobile, apresenta-se o módulo principal que efetua o reconhecimento dos objetos, clicando no menu identificação o usuário recebe a tela da câmera, captura uma foto e aguarda o sistema responder, neste exemplo foi capturada a foto de um objeto previamente cadastrado e após o processamento o resultado é obtido.

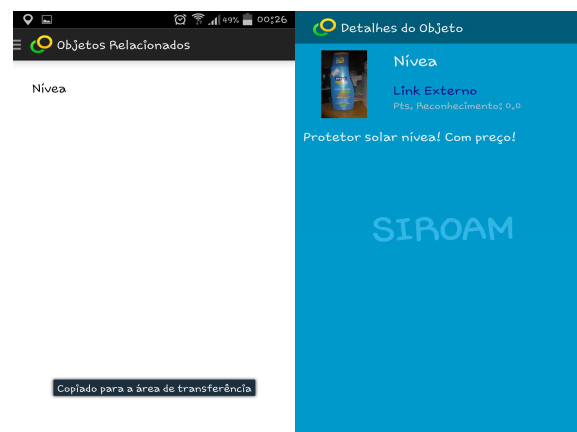
Figura 28 - Na esquerda a foto do objeto capturada pelo dispositivo e na direita a identificação do objeto realizada



Fonte: Autorial própria, 2014.

Após a identificação o usuário pode ver as informações do objeto que foi identificado e também pode acessar o menu objetos relacionados, onde serão listados os objetos relacionados com a identificação de objeto que foi feita e o usuário poderá clicar sobre um desses objetos e ver as informações detalhadas.

Figura 29 - Na esquerda a foto do objeto relacionado e na direita a informação detalhada do objeto clicado



Fonte: Autorial própria, 2014.

Como pode ser observado, o objeto exibido no lado direito da 0 é objeto relacionado com o objeto de identificação da 0. Com esta característica de exibir objetos relacionados o sistema traz vantagens para quem está utilizando-o, evitando até uma identificação desnecessária, visto que, por serem objetos relacionados, podem estar perto um do outro.

Ao encontro do objetivo desta pesquisa que visa avaliar o desempenho e qualidade da identificação dos objetos pelo algoritmo utilizado, as próximas seções apresentarão os resultados obtidos a partir de testes efetuados, levando em consideração algumas variáveis de suma importância. Estes testes foram realizados em um ambiente normal, sem impedimentos físicos de acesso.

6 TESTES E AVALIAÇÃO DO ALGORITMO

No projeto de pesquisa que deu origem a esse trabalho foi considerado como premissa identificar um objeto ou ambiente apenas com reconhecimento de imagens utilizando técnicas de processamento de imagens. Para validar essa premissa e realizar a avaliação do algoritmo de reconhecimento de imagens, o SIROAM foi devidamente instalado e configurado no computador servidor e sua base foi alimentada com o polígono do local onde foram realizados os testes. Também foram incluídos no sistema alguns objetos presentes neste ambiente. Foram escolhidos dois tipos de objetos, o primeiro tipo escolhido foi um objeto de duas dimensões (2D), mais precisamente um quadro e o segundo tipo de objeto escolhido foi um objeto de três dimensões (3D), que, neste caso foi um recipiente de protetor solar.

Foram escolhidos dois tipos distintos de objetos, pois, objetos 2D e 3D não possuem as mesmas representações, objetos 2D são planos e tem apenas uma superfície, sendo assim, esse objeto pode ter alterações em fotografias por diferenças em variáveis de ambientes que serão citadas nas avaliações dos resultados deste projeto. Os objetos 3D possuem mais de uma superfície, por conta desta característica, essas superfícies podem possuir ângulos que refletem a luz do ambiente de forma diferente dos objetos 2D, influenciando diretamente da fotografia deste objeto e por estes critérios levantados é que foram escolhidos ambos os tipos de objetos.

Em seguida, instalou-se o SIROAM em um dispositivo móvel, para os testes foi utilizado um Smartphone Galaxy S4. Todos os recursos do celular foram ativados (WiFi, GPS e Localização Inteligente do Google), estes recursos foram ativados pois, são necessários para o funcionamento correto do SIROAM.

Neste trabalho foram feitos dois testes diferentes, o primeiro teste visa avaliar a qualidade e confiabilidade do reconhecimento de imagens levando em consideração algumas variáveis de ambiente, que, neste caso são: fotos com ângulos e luminosidade diferentes, também foram efetuados testes com o flash ligado e desligado. O segundo teste, visa avaliar o desempenho do algoritmo, para isto, foram avaliados tempo de processamento, tendo como variável, o número de objetos que um determinado local possui. Foi possível efetuar estes testes porque o serviço de identificação salva automaticamente o momento em que o processamento iniciou e finalizou, salvando no banco de dados o número de ponto de semelhança entre os objetos e também o tempo em milissegundos que o processamento levou para executar. A seguir será apresentado cada método de avaliação, bem como os resultados obtidos, seguido de uma conclusão.

6.1 Qualidade e Confiabilidade

Como o algoritmo de reconhecimento de imagens retorna a quantidade de pontos de semelhança entre os objetos comparados, esse será o valor a ser medido. Para efetuar os testes, foram elencadas as seguintes variáveis, bem como sua possível influência sobre uma fotografia:

- Ângulo: em [17] é citado o ângulo como ponto de vista, segundo o autor, se cem pessoas fotografarem uma mesma cena, existirão cem fotos diferentes, pois, cada pessoa destacará algo que o atrai nesta cena, uns fotografarão antes, outros depois, uns chegarão mais perto e outros ficarão mais distantes, uns ficarão de pé, outros se agacharão;
- Luminosidade: conforme [17], sem a luz uma fotografia sequer existiria, a luminosidade é o que torna esta arte possível, é a essência da fotografia e todos seus princípios básicos.
- Flash: também levantado por [17], saber utilizar a luminosidade, tanto a luz ambiente quanto luminosidade artificial é uma técnica de composição de imagem e a disposição e a qualidade da luz pode criar efeitos completamente diversos em uma imagem.

Analisando as variáveis citadas acima, percebe-se que cada uma tem muita influência sobre uma fotografia e dependendo das condições do ambiente, podem ser geradas imagens com características muito diferentes da imagem original do objeto escolhido.

Estas variáveis foram aplicadas, inicialmente, para o objeto 2D, gerando, como resultado a Tabela I que apresenta os resultados obtidos para cada valor de variável, considerando também que esses valores são de identificações efetuadas com sucesso e que para cada teste de valor de variável, foram feitas dez identificações e foi pego a média dos valores obtidos como resultado médio.

Tabela 1 - Avaliação de Objeto 2D

VARIÁVEL	Objeto 2D			
	Valor	Média	Desvio Padrão	%
Ângulo	Igual 0°	84,8	2,14	100%
	Ângulo 25°	53,6	3,73	63,2%
	Ângulo 45°	36,5	2,12	43,0%
	Ângulo 65°	27,3	1,76	32,2%
Luminosidade	Igual 100%	93,8	2,04	100%
	120 %	48,3	3,09	51,5%
	30%	23,4	1,83	24,9%
	50%	25,9	4,06	27,6%
Flash	Nula 0%	0	0	0,0%
	Ligado	89,2	2,20	100%
	Desligado	87,1	1,19	97,6%

Fonte: Autoria própria, 2014.

Os resultados obtidos na Tabela I mostram que variáveis de ambiente influenciam diretamente na identificação dos objetos 2D, considerando os valores da coluna *média*, nitidamente essa influência tem tendência para menos, o que pode, dependendo do objeto que se está identificando, prejudicar a sua localização, reforçando essa conclusão, pode-se ver o desvio padrão da variável luminosidade que é superior e espelha a sensibilidade que a luz do ambiente pode representar em uma identificação de objeto.

Com os resultados obtidos pode-se estabelecer um padrão de aceitação mínimo de pontos de semelhança para que a identificação de um objeto 2D seja aceita, portanto, neste caso, pode-se considerar que um valor aceitável fica em torno de 23 (vinte e três) pontos de semelhança entre os objetos, ou então, 24,9% da pontuação máxima obtida.

Ainda observando a Tabela I, nota-se que a variável flash teve pouca variação sobre o objeto utilizado, por este motivo, foi realizado um segundo teste utilizando um objeto 2D com

uma superfície mais refletora e assim pode-se observar a Tabela II que apresenta os resultados.

Tabela 2 - Avaliação da Variável Flash em Objeto 2d

VARIÁVEL	Objeto 2D			
	<i>Valor</i>	<i>Média</i>	<i>Desvio Padrão</i>	<i>%</i>
Flash	Ligado	77,9	3,90	100%
	Desligado	52,8	3,93	67,7%

Fonte: Autoria própria, 2014.

Avaliando-se a Tabela II, conclui-se que a superfície de um objeto 2D pode também influenciar no reconhecimento de imagem, levando-se em conta que uma superfície mais refletora tem maior influência do que uma superfície fosca.

Para o objeto 3D, foram utilizadas as mesmas variáveis: Ângulo, Luminosidade e Flash, a quantidade de identificações realizadas foram dez para cada valor de variável e foi pego a média dos valores obtidos como resultado médio. Conforme Tabela III, os resultados obtidos foram esses:

Tabela 3 - Avaliação de Objeto 3D

VARIÁVEL	Objeto 3D			
	<i>Valor</i>	<i>Média</i>	<i>Desvio Padrão</i>	<i>%</i>
Ângulo	Igual 0°	102,3	1,82	100%
	Ângulo 25°	35,7	2,16	34,9%
	Ângulo 45°	13,3	1,41	13,0%
	Ângulo 65°	10,2	1,61	9,9%
Luminosidade	Igual 100%	99,1	1,79	100%
	120%	85,5	2,17	86,3%
	30%	7,2	0,91	7,2%
Flash	50%	44,7	2,94	45,1%
	Nula 0%	0	0	0,0%
	Ligado	93,3	2,31	100%
	Desligado	74,6	2,63	79,9%

Fonte: Autoria própria, 2014.

Analisando a Tabela III fica visível que a variável ângulo também influencia a identificação consideravelmente, pois, em um ângulo de 65° do objeto 3D obteve-se uma média de apenas 10,2 pontos de semelhança, valor esse que representa 9,9% dos pontos de semelhança em uma condição ideal de comparação. A luminosidade também influencia com diferenças consideráveis quando a mesma é nula ou reduzida, padrão este que foi observado na avaliação do objeto 2D.

Para um objeto 3D, pode-se assumir um padrão mínimo de pontos de semelhança entre os objetos através dos resultados da Tabela III, com isso, conclui-se que um objeto 3D pode ser identificado com no mínimo 10 (dez) pontos de semelhança ou então 9,9% da pontuação máxima obtida.

Efetuada uma comparação entre as tabelas I e III, observa-se que os padrões de resultados obtidos entre as tabelas são semelhantes, ficando apenas, uma diferença considerável na variável de ângulo, que incide mais sobre objetos 3D no algoritmo. Já os resultados do desvio padrão de ambas tabelas mostram que para objetos 3D a variação é maior, ficando claro que a identificação de objetos 3D encontra menos pontos de semelhança conforme a influência das variáveis citadas na Tabela III. O ponto positivo entre as comparações é que a identificação do objeto não conflitou com outro objeto, ou seja, mesmo com uma baixa taxa de pontos de semelhança o objeto foi reconhecido corretamente, não trazendo para o usuário final um resultado errado.

6.2 Escalabilidade

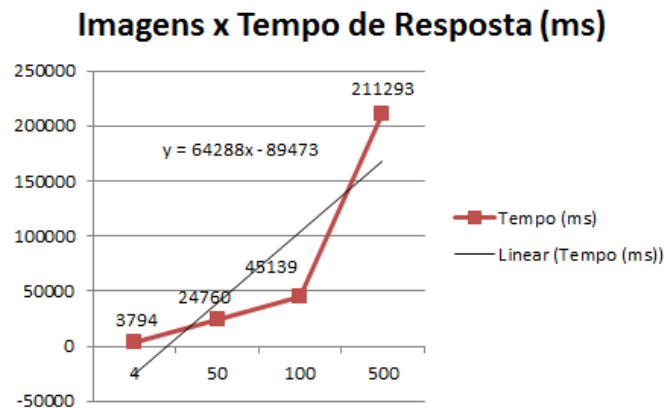
A análise de desempenho que foi aplicada ao SIROAM está direcionada para a escalabilidade de desempenho, conforme [15], escalabilidade é a capacidade de um componente ou de um software manter ou aumentar o desempenho (tempo de resposta) quando há um aumento no número de usuários e/ou de requisições simultâneas / processamento. O SIROAM, visto de uma maneira abrangente, tem características de um sistema escalável por possuir diversos módulos independentes que executam determinadas funções e retornam o resultado de um processamento ao módulo requisitante, teoria esta defendida por [15].

Como não foi avaliada a ferramenta como um todo e sim apenas o módulo de processamento de imagens, cabe apresentar nesta seção como foi feita a análise de escalabilidade. O serviço de processamento de imagem guarda em sua execução o momento inicial de um reconhecimento de imagem e também o momento final, após a execução é feito

uma subtração do momento final com o momento inicial para obter-se o tempo gasto no processamento. Este dado é armazenado no banco de dados do SIROAM e será consultado após cada processamento para ser feito uma análise final de escalabilidade do algoritmo.

Para efetuar os testes, foram feitas inclusões de registros no banco de dados de diversas imagens que pertencem ao ambiente onde o dispositivo móvel está localizado. A quantidade de imagem será divulgada, bem como o tempo de processamento com a quantidade de imagem atual do banco de dados e estas informações serão os dados utilizados para efetuar a análise.

Gráfico 1 - Gráfico com resultado da análise de escalabilidade do SIROAM



Fonte: Autoria própria, 2014.

Analisando o gráfico da 0, nota-se que foram utilizadas para o teste quantidades diferentes de imagens e conforme esta quantidade aumenta o tempo de resposta do sistema aumenta consideravelmente. Pode-se concluir que o algoritmo de processamento de imagens utilizado pelo SIROAM é escalável, embora não se comporte conforme citado em [15], pois, apresenta um comportamento de tempo de resposta de ordem linear, permitindo que o desenvolvedor do sistema preveja estas situações e trabalhe em melhorias para o mesmo.

Este comportamento ocorre porque o módulo carrega a cada requisição todas as imagens a serem comparadas do local onde o dispositivo móvel está localizado, posteriormente essas imagens são processadas pelo algoritmo para em um terceiro momento ser calculado quantos pontos de semelhança cada imagem tem com o objeto que está sendo localizado.

Contudo, com a utilização de coordenadas geográficas, recurso presente no SIRAM para efetuar um filtro sobre quais imagens deverão ser comparadas, o problema da

escalabilidade pode ser reduzido, visto que, um local pode ser segmentado em vários polígonos menores que, por sua vez, conterão pequenas quantidades de imagens.

7 CONSIDERAÇÕES FINAIS

Com este trabalho, foi possível identificar a contribuição que um software para reconhecimento de imagens pode trazer para diversos tipos de ambientes, em especial, aqueles que não podem sofrer alterações físicas, por se tratarem de ambientes de patrimônio histórico ou até mesmo, para reduzir custos em uma implementação de um sistema, pois, com a utilização do SIROAM, não haveria necessidade de investimentos caros com hardware.

Quanto ao SIROAM, ficou evidente que o seu funcionamento vem de encontro ao que foi proposto neste trabalho, através dos testes efetuados, nota-se que mesmo com variações de cada ambiente, especificamente nos quesitos de ângulo de foto, luminosidade de ambiente e flash da câmera do dispositivo móvel, foi possível efetuar identificações de objetos corretamente, mudando apenas, a nível de sistema, o quanto uma imagem é semelhante com a outra, porém isto não significou que uma identificação de objeto foi feita de forma errada.

Os resultados obtidos nos testes demonstram que é possível identificar um objeto ou ambiente apenas com reconhecimento de imagens utilizando técnicas de processamento de imagens e assim auxiliar as pessoas e as fazer entender a significância dos objetos ou locais que elas estão frequentando, pois, além do reconhecimento de imagem, o SIROAM traz em uma identificação a descrição completa do objeto que foi localizado ou do local onde o dispositivo se encontra.

O desenvolvimento do SIROAM e a análise de seu algoritmo de processamento de imagens estão de acordo com os objetivos deste trabalho de investigar e implementar o reconhecimento de imagens através de técnicas de processamento de imagens em ambientes previamente mapeados e avaliar a qualidade, confiabilidade o desempenho do algoritmo utilizado.

Mesmo com a contribuição que o SIROAM traz, diversos recursos podem ser melhorados neste projeto, como por exemplo, melhorias na interface gráfica do ambiente Mobile e também, melhorias no problema da escalabilidade identificado. Em trabalhos futuros, implementações multi-task podem ser adicionadas, para que o SIROAM utilize todos os núcleos disponíveis nos processadores dos servidores que o mesmo será instalado e também, o pode ser considerado uma melhoria, o pré-carregamento das imagens de um local

em forma de serviço que fica disponível para utilização, diminuindo, no tempo de resposta, requisições de I/O que ocorrem, bem como a eliminação do tempo de treino do algoritmo, restando apenas o tempo necessário para calcular a similaridade entre um objeto e as fotos previamente cadastradas.

Os códigos do projeto, podem ser adotados para fins acadêmicos, para utilização em cadeiras de processamento de imagens, geoprocessamento, tecnologias Java, Android e cadeiras vinculadas à arquiteturas de projetos Microsoft .NET.

ABSTRACT

This article presents the results of a software tool developed using techniques of image processing for recognition and identification of objects. The developed tool, named SIROAM (Integrated Object Recognition Mapped Environments) was evaluated through their use, measuring its reliability through feedback and performance analysis of its identification algorithm. By testing a non-intrusive mode for image recognition, so it was discovered that some environment variables influence the identification of an object and the indices hit ratio are good. Also as a result of the evaluation it was discovered that the identification algorithm used was scalable due to their performance decreases linearly.

REFERENCIAS

- [1] COSTA, Yandre Maldonado e Gomes da. **Indexação e Recuperação de Imagens por Cor e Estrutura**. 2002. 77 f. Dissertação (Mestrado em Ciência da Computação) - Universidade Federal do Rio Grande do Sul, Porto Alegre, 2002.
- [2] HOSHIRO, H. S.; GOLDSCHIMIDT, R. R. **Processamento de Imagens**. Rio de Janeiro: Instituto Militar de Engenharia, 2008. Disponível em: <http://www.comp.ime.eb.br/techreports/repositorio/2008_05.pdf>. Acesso em: jun. 2014.
- [3] HALDAR, Poulami. **Content based Image Retrieval using Histogram, Color and Edge**. 2012. 57 f. Dissertação (Mestrado em Desenvolvimento Multimídia) - Jadavpur University, Índia: Kolkata, 2012.
- [4] BHERING, Fabiano Pereira. **Recuperação de Imagens Utilizando Técnicas de Agrupamento de Dados**. 2012. 55 f. Dissertação (Mestrado em Informática) - Pontifícia Universidade Católica de Minas Gerais, Minas Gerais, 2012.
- [5] ANDRIJAUSKAS, Fabio. **Sistema de Indexação de Imagem Baseado em Conteúdo Utilizando Reconhecimento de Padrão**. 2007. Universidade São Francisco, São Paulo, 2007.

- [6] CLARK, Donald. **Google Goggles** – mind goggling application. 2010. 1 Fotografia. Color. Disponível em: <<http://donaldclarkplanb.blogspot.com.br/2010/08/google-goggles-mind-goggling.html>>. Acesso em: 05 maio 2014.
- [7] RUDEK, Marcelo. Uma Abordagem para Indexação e Recuperação Automática de Imagens Baseada em Vetor de Comportamento para Automação Industrial. In: SIMPÓSIO DE AUTOMAÇÃO INTELIGENTE, SBAI, 5, 2001, Gramado. **Anais...**, Gramado, 2001.
- [8] LARMAN, C. **Utilizando UML e Padrões: uma introdução à análise e ao projeto orientados a objetos e desenvolvimento iterativo**. Porto Alegre: Bookman, 2007.
- [9] BISSI, Wilson. Scrum: metodologia de desenvolvimento ágil. **Cesumar**. Maringá: Centro Universitário de Maringá, v.09, n.2, p. 143-151, dez. 2007.
- [10] HELENA, Denise. As linguagens de programação mais populares para 2014. **WWWhatsNew**. Espanha, 2014. Disponível em: <<http://br.wwwwhatsnew.com/2014/02/as-linguagens-de-programacao-mais-populares-para-2014/>>. Acesso em: 09 jun. 2014.
- [11] ITSEEZ. **OpenCV Documentation**. Disponível em: <<http://opencv.org/documentation.html>>. Acesso em: 09 jun. 2014.
- [12] EMGU. **EmguCV**. Disponível em: <http://www.emgu.com/wiki/index.php/Main_Page>. Acesso em: 11 jun. 2014.
- [13] PULLI, Kari et al. Realtime Computer Vision With OpenCV. **ACM Queue**, v.10, n.4, abr., 2012. 40p.
- [14] TWITTER. **Bootstrap**. Disponível em: <<http://getbootstrap.com>>. Acesso em 16 jun. 2014.
- [15] PORTO, Ives Oliveira. **Padrões e Diretrizes Estruturais Para Escalabilidade de Sistemas**. 2009. 161 f. Dissertação (Mestrado em Ciência da Computação) - Universidade Federal de Uberlândia, Minas Gerais, 2009.
- [16] PORTAL DA EDUCAÇÃO. **Georreferenciamento e Geoprocessamento**. 2013. Disponível em: <<http://www.portaleducacao.com.br/informatica/artigos/51022/georreferenciamento-e-geoprocessamento>>. Acesso em: 17 jun. 2014.
- [17] CALA A BOCA E CLICA. **Introdução à Fotografia**. 2013. Disponível em: <<http://www.calabocaeclica.com/p/curso-de-introducao-fotografia.html>>. Acesso em: 18 jun. 2014.
- [18] DEVIN. **Modelo de Desenvolvimento Ágil SCRUM**. 2009. Disponível em: <<http://www.devin.com.br/modelo-scrum>>. Acesso em: 25 jun. 2014.
- [19] 2GATHER TECNOLOGIA. **Desenvolvimento de Aplicações Inovadoras**. 2014. Disponível em: <http://2gather.com.br/?page_id=78>. Acesso em: 25 jun. 2014.