

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
COORDENAÇÃO DO CURSO SUPERIOR DE TECNOLOGIA EM SISTEMAS PARA
INTERNET
CURSO SUPERIOR DE TECNOLOGIA EM SISTEMAS PARA INTERNET

MARCELO KOTI NISHI

**UM SISTEMA BASEADO EM INTELIGÊNCIA COLETIVA PARA
VISUALIZAÇÃO DE PROBLEMAS EM VIAS PÚBLICAS**

TRABALHO DE CONCLUSÃO DE CURSO

CAMPO MOURÃO
2012

MARCELO KOTI NISHI

**UM SISTEMA BASEADO EM INTELIGÊNCIA COLETIVA PARA
VISUALIZAÇÃO DE PROBLEMAS EM VIAS PÚBLICAS**

Trabalho de Conclusão de Curso de graduação, apresentado à disciplina de Trabalho de Conclusão de Curso II, do Curso Superior de Tecnologia em Sistemas para Internet da Coordenação do Curso Superior de Tecnologia em Sistemas para Internet da Universidade Tecnológica Federal do Paraná – UTFPR, como requisito para aprovação na disciplina.

Orientadora: Prof^ª. MSc. Ana Paula Chaves Steinmacher

CAMPO MOURÃO
2012



ATA DA DEFESA DO TRABALHO DE CONCLUSÃO DE CURSO

As **dezenove horas** do dia **dezenove de novembro de dois mil e doze** foi realizada na sala B105 da UTFPR-CM a sessão pública da defesa do Trabalho de Conclusão do Curso Superior de Tecnologia em Sistemas para Internet do acadêmico **Marcelo Koti Nishi** com o título **UM SISTEMA BASEADO EM INTELIGÊNCIA COLETIVA PARA VISUALIZAÇÃO DE PROBLEMAS EM VIAS PÚBLICAS**. Estavam presentes, além do acadêmico, os membros da banca examinadora composta pelo professor **Me. Ana Paula Chaves Steinmacher** (Orientador-Presidente), pelo professor **Me. Lucio Geronimo Valentin** e pelo professor **Me. Igor Fábio Steinmacher**. Inicialmente, o aluno fez a apresentação do seu trabalho, sendo, em seguida, arguido pela banca examinadora. Após as arguições, sem a presença do acadêmico, a banca examinadora o considerou **APROVADO** na disciplina de Trabalho de Conclusão de Curso e atribuiu, em consenso, a nota ____ (_____). Este resultado foi comunicado ao acadêmico e aos presentes na sessão pública. A banca examinadora também comunicou ao acadêmico que este resultado fica condicionado à entrega da versão final dentro dos padrões e da documentação exigida pela UTFPR ao professor Responsável do TCC no prazo de **quatro dias**. Em seguida foi encerrada a sessão e, para constar, foi lavrada a presente Ata que segue assinada pelos membros da banca examinadora, após lida e considerada conforme.

Observações:

Campo Mourão, 19 de novembro de 2012.

Prof. Me. Lucio Geronimo Valentin
Membro

Prof. Me. Igor Fábio Steinmacher
Membro

Prof. Me. Ana Paula Chaves
Steinmacher
Orientador

Aos meus pais, por estarem sempre me apoiando em todos os momentos de minha vida.

AGRADECIMENTOS

Agradeço aos meus pais, Paulo Koiti Nishi e Akeme Marta Fugisawa Nishi, e minhas irmãs, Karla Sanae Nishi, Cíntia Kazumi Nishi e Priscila Nishi, por terem acreditado em mim, e, sem eles, tudo se tornaria mais difícil.

Agradeço a Professora MSc. Ana Chaves Steinmacher pela orientação e dedicação por este trabalho.

Ao Professor MSc. Igor F. Steinmacher, que me apoiou nos momentos difíceis encarados neste trabalho.

Aos professores da banca examinadora pela atenção e contribuição a este estudo.

A todos os professores da Universidade Tecnológica Federal do Paraná (UTFPR) por terem me ensinado ao longo dos últimos anos.

A todos os colegas, gostaria de deixar minha satisfação de ter convivido com eles antes e durante este estudo.

Agradeço ao CNPq (processo 560135/2010-6) e à RNP (projeto SIMTUR - edital CTIC) pelo apoio financeiro.

RESUMO

NISHI, Marcelo K. **Um Sistema Baseado em Inteligência Coletiva para Visualização de Problemas em Vias Públicas** 2012. 41 f. Trabalho de Conclusão de Curso (Graduação) – Curso Superior de Tecnologia em Sistemas para Internet. Universidade Tecnológica Federal do Paraná. Campo Mourão, 2012.

O surgimento da Web 2.0 facilitou o compartilhamento do conhecimento pela Internet. A inteligência coletiva aproveita esse conhecimento compartilhado para a resolução de problemas. Utilizar inteligência coletiva em conjunto com as técnicas e tecnologias da área de TI (Tecnologia da Informação) nos sistemas de transporte pode torná-los mais seguros, eficientes e confiáveis, sem a necessidade de alterações nas estruturas físicas existentes. Os produtos gerados a partir dessa combinação são conhecidos como ITS (Sistemas de Transporte Inteligente, do inglês *Intelligent Transportation Systems*). Este trabalho apresenta um aplicativo que utiliza o conceito de ITS e inteligência coletiva para fornecer a visualização por meio de mapas de possíveis problemas em linhas de ônibus ocasionados por más condições de trânsito. O aplicativo faz parte do projeto UbiBus, que foca em reunir soluções a fim de melhorar os serviços fornecidos aos usuários de transporte público do Brasil.

Palavras-chave: Inteligência Coletiva. Sistemas de Transporte Inteligente. Transporte Público.

ABSTRACT

NISHI, Marcelo K. **A Collective Intelligence Based System for Visualizing Problems in Public Roads** 2012. 41 f. Trabalho de Conclusão de Curso (Graduação) – Curso Superior de Tecnologia em Sistemas para Internet. Universidade Tecnológica Federal do Paraná. Campo Mourão, 2012.

The emergence of Web 2.0 made knowledge sharing via Internet an easy task. Collective intelligence leverages this knowledge shared to solve problems. Using collective intelligence along with IT (Information Technology) techniques in transportation systems can make them more secure, efficient and reliable without changing existing physical structures. The products generated by this combination are known as ITS (Intelligent Transportation Systems). This paper presents an application that uses the concept of ITS and collective intelligence to enable users to visualize possible problems on bus lines due to traffic conditions. The application is part of the project UbiBus, which focuses on bringing solutions together to improve services provided to public transportation users in Brazil.

Keywords: Collective intelligence. Intelligent Transportation Systems. Public Transportation

LISTA DE TABELAS

Tabela 1 - Porcentagem de mensagens retornadas com relação à problema no trânsito.....	27
Tabela 2 - Escala de valores.....	28

LISTA DE SIGLAS

AJAX	Asynchronous Javascript and XML
API	Application Programming Interface
DAO	Data Access Object
GPS	Global Positioning System
ITS	Intelligent Transportation System
JPA	Java Persistence API
JSF	JavaServer Faces
MVC	Model-view-controller
ORM	Object-Relational Mapping
POI	Point of Interest
POJO	Plain Old Java Object
SGBD	Sistema de Gerenciamento de Banco de Dados
SMS	Short Message Service
TI	Tecnologia da Informação

LISTA DE ILUSTRAÇÕES

Figura 1 - Metodologia.....	13
Figura 2 - OneBusAway.....	16
Figura 3 - ToTransit.....	17
Figura 4 - Trânsito.....	18
Figura 5 - GoogleMaps.....	19
Figura 6 - Waze.....	19
Figura 7 - Wikipédia.....	21
Figura 8 - WikiCrimes.....	22
Figura 9 - Arquitetura UbiBus.....	24
Figura 10 - Ocorrência.....	25
Figura 11 - Processo Batch.....	26
Figura 12 - Modelo de dados.....	29
Figura 13 - Diagrama de classes.....	31
Figura 14 - Tela de escolha da linha.....	33
Figura 15 - Tela de escolha dos pontos de origem e destino.....	34
Figura 16 - Tela de resultado.....	34
Figura 17 - Classe batch.....	35

SUMÁRIO

Coordenação do Curso Superior de Tecnologia em Sistemas para Internet.....	1
AGRADECIMENTOS.....	3
RESUMO.....	4
ABSTRACT.....	5
1 INTRODUÇÃO.....	10
1.1 OBJETIVO.....	12
1.2 OBJETIVOS ESPECÍFICOS.....	12
2 METODOLOGIA.....	13
3 REFERENCIAL TEÓRICO.....	15
3.1 INTELLIGENT TRANSPORTATION SYSTEM (ITS).....	15
3.2 INTELIGÊNCIA COLETIVA.....	20
3.3 TECNOLOGIAS UTILIZADAS.....	22
4 ANÁLISE E PROJETO DO SISTEMA.....	24
4.1 ANÁLISE DAS MENSAGENS	26
5 IMPLEMENTAÇÃO DA APLICAÇÃO.....	29
5.1 INTERFACE GRÁFICA.....	33
5.2 SERVIÇOS WEB.....	36
6 CONCLUSÕES E TRABALHOS FUTUROS.....	38
REFERÊNCIAS.....	40

1 INTRODUÇÃO

Nas últimas décadas, houve um grande aumento na quantidade de veículos no mundo. Em 1950, havia cerca de 70 milhões de veículos. Até o ano de 1994, foi constatada que, a quantidade de veículos passou para 630 milhões, nove vezes a quantidade estimada em 1950. Desde 1970 a quantidade de veículos tem aumentando em 16 milhões ao ano. Nesse ritmo, estima-se que até o ano de 2025 haverá mais de um bilhão de veículos (SHAH; DAL, 2006).

A infraestrutura das vias urbanas não pode acompanhar o ritmo de aumento de veículos, pois, Shah e Dal (2006) afirmam que uma das principais razões é a necessidade de um longo período de investimento. Além disso, Shah e Dal (2006) citam vários estudos que comprovam que os custos para novas infraestruturas são altamente elevados, assim, não sendo uma boa opção. O resultado é o aumento do tráfego de veículos nas vias, dificultando o seu fluxo e afetando diretamente os transportes públicos. Segundo Chaves, Steinmacher e Vieira (2011) o maior causador de atrasos dos transportes públicos é o congestionamento, seguido de acidentes automotivos, roubos, alagamentos e outros tipos de incidentes. Devido a esses atrasos e à falta de informações relacionadas a eles, muitas pessoas deixam de utilizar o transporte público como meio de locomoção (CHAVES; STEINMACHER; VIEIRA, 2011), procurando outros meios, sendo o mais provável o transporte particular. Aumenta-se, então, o número de veículos circulando em vias urbanas, dificultando o fluxo, principalmente nos centros das grandes cidades e em horários com maiores fluxos, a chamada “Hora do *Rush*”.

A aplicação de ITS (*Intelligent Transportation Systems* – em português, Sistemas de Transporte Inteligente) no sistema de transporte público, pode ser utilizada para contribuir na melhoria dos fluxos nas vias. Informações como tempo de atraso e condições da via disponibilizadas em tempo real, podem ajudar os usuários no planejamento da locomoção, motivando-os a utilizar o transporte público e abrindo portas para potenciais novos usuários que deixariam de utilizar seus veículos particulares. Segundo um estudo citado por Barry (2011), as pessoas estão dispostas a deixar seus carros, caso haja aplicações fornecendo horários, atrasos, lojas e serviços ao longo das rotas em tempo real.

Diante desse cenário, um projeto, denominado UbiBus¹, foi proposto com o objetivo de oferecer um conjunto de soluções tecnológicas para facilitar o acesso à informações de transporte público aos usuários. Essas informações devem ser disponibilizadas em tempo real, baseadas em informações dinâmicas de contexto, integradas em um sistema de transporte público inteligente, ubíquo e sensível ao contexto (VIEIRA; CALDAS; SALGADO, 2011).

O projeto UbiBus prevê o desenvolvimento de várias aplicações, desde soluções web e redes sociais a aplicações para dispositivos móveis, terminais e quiosques. Um conjunto dessas aplicações é voltado à utilização da inteligência coletiva para registrar e exibir informações relacionadas a ocorrências de trânsito. Ocorrências de trânsito, nesse contexto, são quaisquer fatos que possam acontecer durante o percurso que podem influenciar a eficiência do serviço de transporte, como congestionamentos, alagamentos, assaltos, acidentes, superlotação, más condições de veículos, entre outros.

Este trabalho apresenta uma aplicação que utiliza essas ocorrências produzidas pela inteligência coletiva, para exibir em um mapa a possibilidade de haver problemas em determinado trecho, que afetam a eficiência do transporte público. Esse sistema será integrado ao projeto UbiBus para permitir que os usuários consultem a situação de determinada linha do transporte público, obtendo informações de possíveis problemas em determinados trechos dessa linha. Caso haja problemas, os usuários poderão tomar ciência se aqueles problemas podem ou não gerar atrasos, dependendo da intensidade do problema informada pelo aplicativo por meio de mapa de calor. Um mapa de calor é uma representação gráfica de um conjunto de dados em que a intensidade é representada por cores.

A partir do aplicativo, pode-se melhorar os serviços fornecidos aos usuários de transporte público com o uso do conceito de ITS, utilizando as ocorrências postadas por usuários do projeto UbiBus para informar a possível situação do trânsito das vias que afetam o transporte público através da inteligência coletiva.

¹ Dos editais: "UbiBus: Um Sistema de Transporte Público Inteligente, Ubíquo e Sensível ao Contexto" - Edital CNPq nº 09/2010 - PDI - Grande e Pequeno Porte, processo número 560135/2010-6 (Grande Porte); e "SIMTUR: Sistema Inteligente para Monitoramento de Tráfego Urbano" - Edital CTIC/RNP - Cidades Inteligentes, 2011.

1.1 OBJETIVO

Fornecer uma interface que exibe em tempo real, através de mapa de calor, a possibilidade de haver problemas em determinadas vias que afetam os transportes públicos, com base em ocorrências coletadas através da inteligência coletiva.

1.2 OBJETIVOS ESPECÍFICOS

A seguir estão listados os objetivos específicos necessários para alcançar o objetivo do Trabalho de Conclusão de Curso:

- definição das palavras-chave que serão utilizadas para classificar/agrupar as ocorrências de acordo com o problema que relatam;
- definição das escalas que serão utilizadas para cada tipo de ocorrência, a fim de calcular a possibilidade de problemas no trânsito naquele momento;
- implementação do algoritmo que define as condições prováveis da rota com base nas informações registradas pelos usuários sobre a situação do trânsito naquele trajeto;
- implementação da interface da aplicação;
- implementação dos serviços web.

2 METODOLOGIA

Para alcançar os objetivos do trabalho, adotou-se a metodologia apresentada na Figura 1.

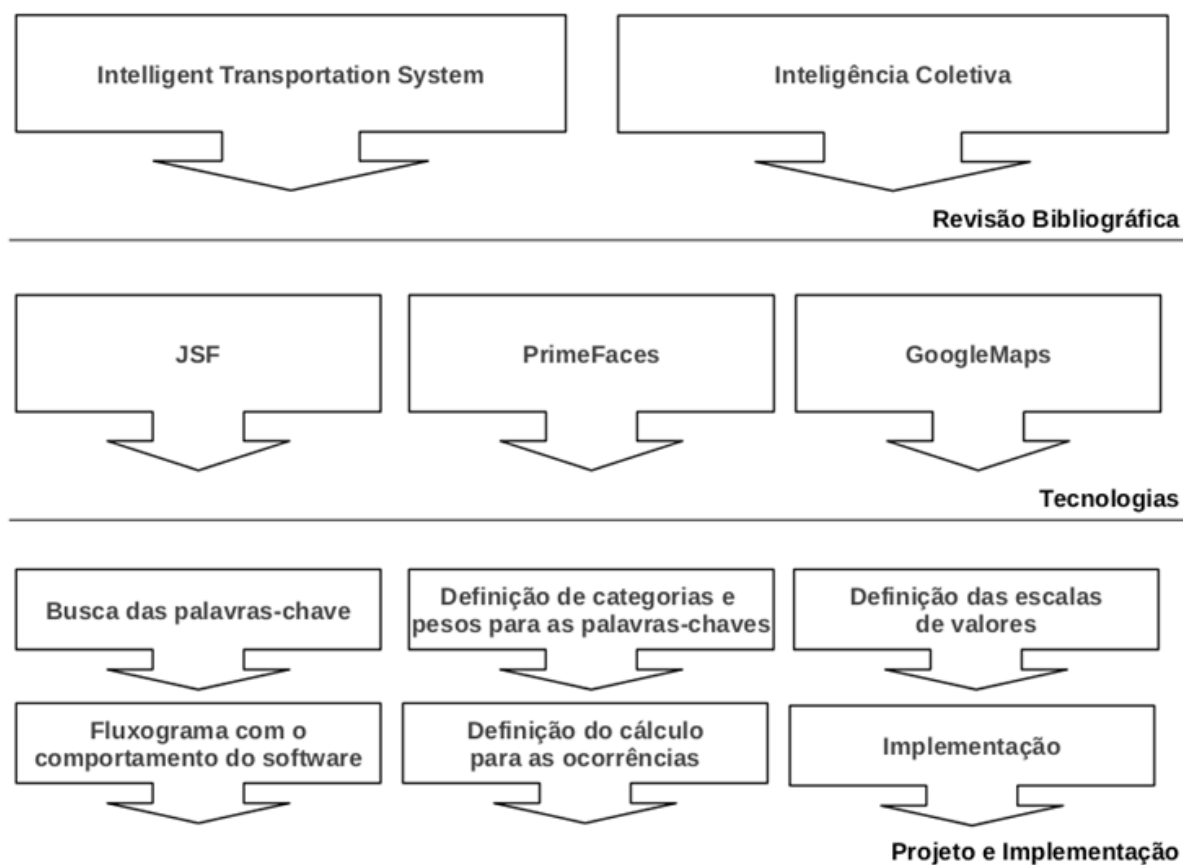


Figura 1 - Metodologia
Fonte: Autoria própria.

Através da revisão bibliográfica, foi realizado o estudo dos conceitos que foram utilizados nesse trabalho. Os conceitos estudados foram: ITS (*Intelligent Transportation Systems*) e inteligência coletiva. Os resultados destes estudos podem ser vistos nas Seções 3.1 e 3.2.

As tecnologias foram definidas através do estudo de tecnologias web e APIs (*Application Programming Interface* – em português, Interface de Programação de Aplicativos) para a integração do GoogleMaps. Logo, as tecnologias utilizadas foram: JSF (*JavaServer Faces*), PrimeFaces e o GoogleMaps, neste último utilizando a biblioteca Gmaps4JSF. Os detalhes podem ser vistos na Seção 3.3.

A análise para definição do comportamento e arquitetura do sistema foram

feitos na atividade de Projeto e Implementação, descritos no Capítulo 4. Nesta atividade, definiram-se os elementos necessários para a aplicação, como a base de dados, POJOs (*Plain Old Java Objects*), classes e APIs para auxílio no cálculo da possibilidade de problemas, como é realizado o cálculo com base nas ocorrências para decidir a intensidade dos problemas, a implementação da interface para exibição do mapa de calor e, por fim, os serviços web.

3 REFERENCIAL TEÓRICO

Nesta seção serão apresentados os conceitos utilizados no Trabalho de Conclusão de Curso.

3.1 INTELLIGENT TRANSPORTATION SYSTEM (ITS)

Uma pesquisa realizada por Caulfield e O'Mahony (2007), que foca em verificar quais tipos de informações as pessoas podem utilizar quando estão planejando suas rotas para utilizar o transporte público, demonstra que a falta de informações quanto à chegada dos transportes públicos aos seus pontos é um fator que causa frustração entre essas pessoas. A pesquisa ainda aponta que o método mais importante para fornecimento de informações é a informação em tempo real. Além disso, o resultado demonstra que a forma mais popular de adquirir informações sobre o transporte público é pela Internet.

Aplicações que utilizam computadores, sensores e tecnologia podem ajudar as pessoas a obterem várias informações, inclusive em tempo real, sobre o transporte público, assim, resolvendo muitos problemas que as pessoas que utilizam esses meios de locomoção encaram no dia a dia.

O uso de computadores, sensores e tecnologia de comunicações em várias outras áreas é denominado TI (Tecnologia da Informação). O uso de TI nos sistemas de transportes é denominado ITS (*Intelligent Transportation Systems* – em português, Sistemas de Transporte Inteligente) (YOKOTA, 2004).

ITS é o uso de várias tecnologias nos sistemas de transportes, para torná-los mais seguros, eficientes, confiáveis e mais ambientalmente amigável sem a necessidade de alterações nas estruturas físicas existentes (YOKOTA, 2004)(ZHAO, 1997).

Segundo Yokota (2004), ITS fornece dois tipos de benefícios: resoluções dos problemas de tráfego e o melhoramento dos serviços fornecidos aos usuários, aumentando a eficiência do transporte.

Várias aplicações que fazem o uso de ITS estão surgindo atualmente. Esse é o caso do OneBusAway (Figura 2), que informa aos seus usuários os horários de chegada, em tempo real, de cada veículo do transporte público em seus respectivos pontos em Washington, nos Estados Unidos, informando-os se há atraso ou não, além de fornecer os horários estimados de chegada de cada veículo. Há também informações estáticas como tabelas de horários de determinadas linhas. O aplicativo está disponível para acesso através de várias interfaces: pela web, telefone, SMS, smartphones e outros celulares.

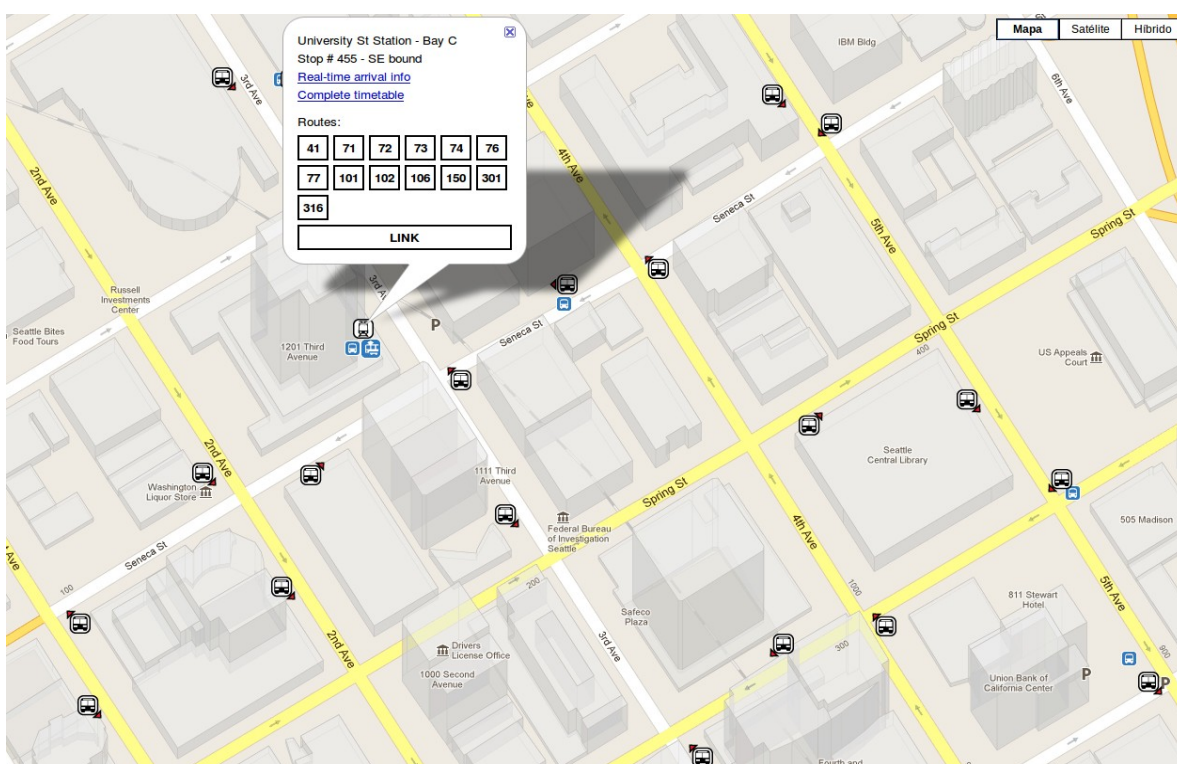


Figura 2 - OneBusAway

Fonte: <http://onebusaway.org/where/standard/>.

Outro aplicativo que faz o uso de ITS é o ToTransit (Figura 3), um aplicativo que informa em tempo real onde os bondes de determinadas linhas da cidade de Toronto no Canadá estão naquele exato momento. Cada bonde é mostrado no mapa com seus deslocamentos em tempo real, assim, cada usuário pode estimar o tempo que cada bonde levará para chegar a determinado ponto. O ToTransit está disponível somente via web.

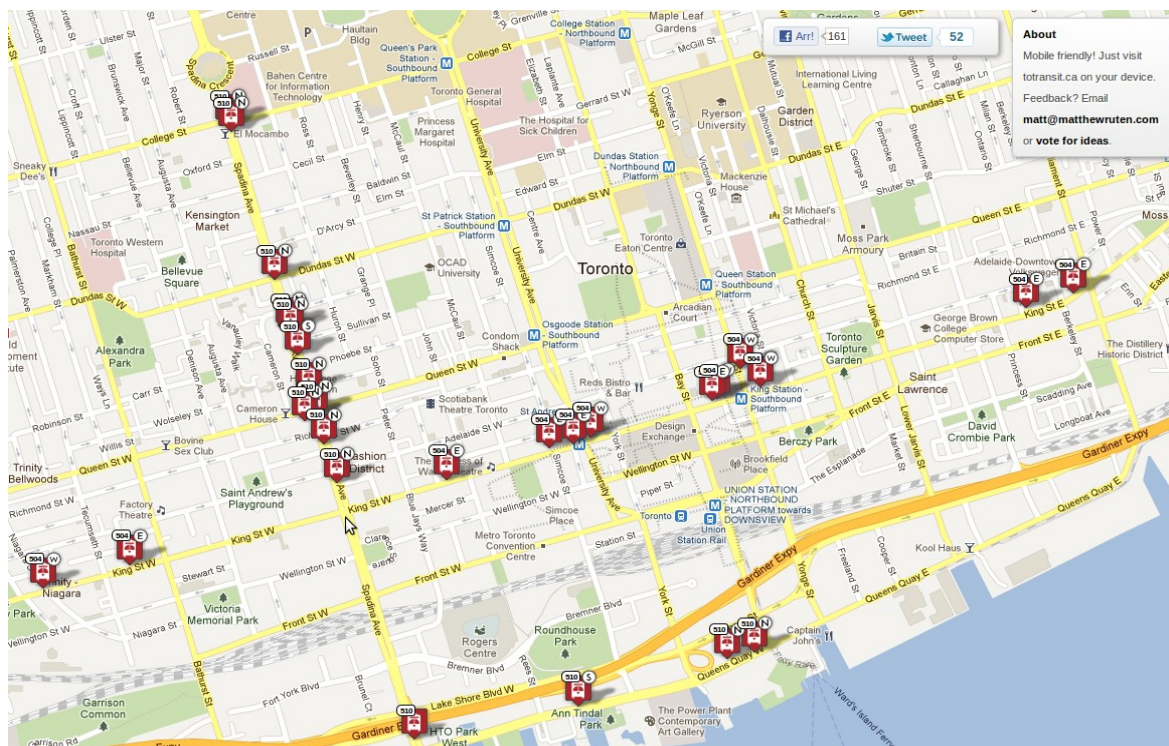


Figura 3 - ToTransit
 Fonte: <http://totransit.ca/>.

O portal Estação criou um aplicativo chamado Trânsito (Figura 4), que visa ajudar os usuários do aplicativo a encontrar o melhor caminho a ser seguido pela cidade. O aplicativo mostra em tempo real como está a situação do trânsito ao redor da localização do usuário em um mapa digital. Além disso, há uma função que mostra em tempo real a situação das principais vias e estradas, através de imagens de câmeras. Há informações também de notas sobre incidentes viários. O aplicativo funciona nas seguintes cidades: Barueri, Belo Horizonte, Cuiabá, Curitiba, Diadema, Goiânia, Recife, Rio de Janeiro, Salvador, São Paulo e Vitória. O Trânsito é gratuito e está disponível para iPhone e aparelhos com o sistema operacional Android.

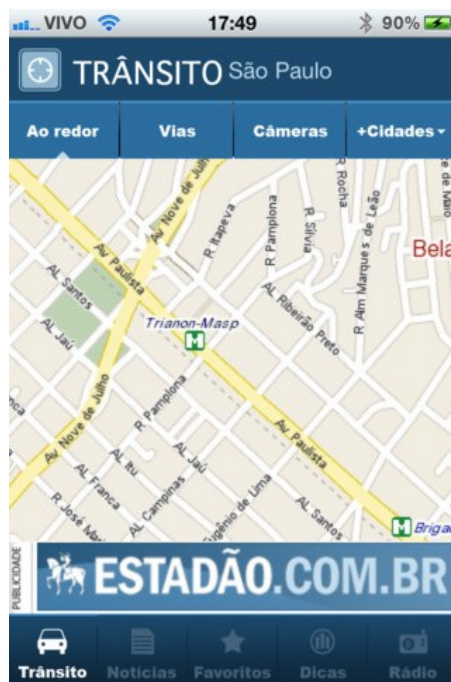


Figura 4 - Trânsito

Fonte: <http://www.estadao.com.br/celular/transito/>.

No GoogleMaps (Figura 5), há um recurso que mostra em tempo real a situação do trânsito das vias mais importantes de algumas cidades brasileiras. Essas informações são mostradas através de linhas de calor traçadas nas vias. As linhas podem ser mostradas pelas cores: verde (trânsito rápido), amarelo (trânsito normal), vermelho (trânsito lento) e vermelho e preto (trânsito muito lento).

Há também aplicativos que além de utilizarem o conceito de ITS, fazem o uso de inteligência coletiva. Waze (Figura 6) é um aplicativo que fornece informações em tempo real sobre a situação do trânsito com o auxílio do GPS (*Global Positioning System* – em português, Sistema de Posicionamento Global). Essas informações são adquiridas através da inteligência coletiva. Os usuários postam essas informações, e o Waze utiliza essas ocorrências para mostrar em tempo real a situação do trânsito. O objetivo do Waze é permitir aos usuários traçarem suas rotas como se fosse um GPS automotivo normal, porém, caso essa rota que o usuário está seguindo obtiver informações de ocorrências reportadas por outros usuários, ele recalcula automaticamente outra rota. Além de informar a situação do trânsito, há também informações de POI (*Point of Interest* – em português, Ponto de Interesse) que são pontos que podem ser úteis aos usuários, como restaurantes, postos policiais entre outros.

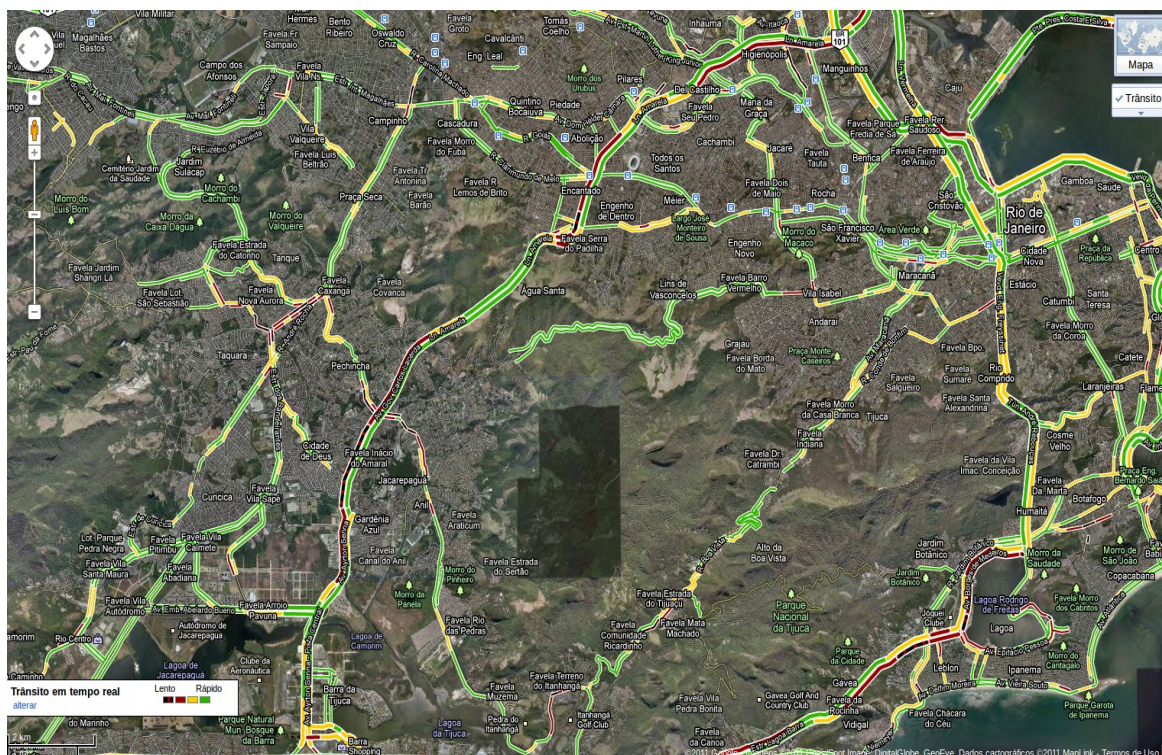


Figura 5 - GoogleMaps
 Fonte: <http://maps.google.com.br/>.

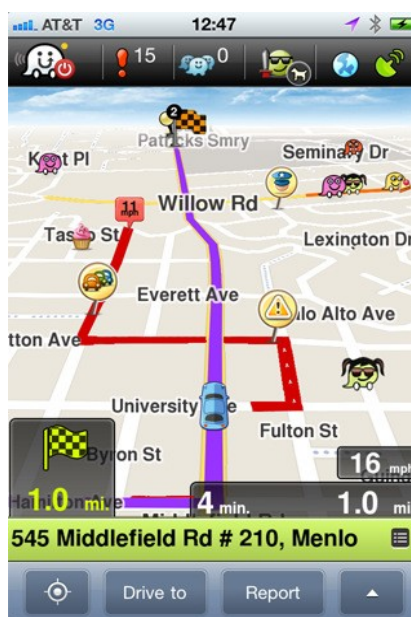


Figura 6 - Waze
 Fonte: <http://www.waze.com/>.

Além desses aplicativos citados, existem vários outros que fazem o uso do conceito de ITS, e alguns utilizam juntamente com o ITS, o conceito de inteligência coletiva, usando os dados informados por usuários e os processando para utilizá-los nas informações passadas pelos aplicativos. O conceito de inteligência coletiva será apresentado na seção a seguir.

3.2 INTELIGÊNCIA COLETIVA

Nos dias de hoje, existe uma grande quantidade de sistemas de conhecimento colaborativo como redes sociais, wikis, blogs, fóruns de discussões, entre outros, que possibilitam o compartilhamento de conhecimento pela Internet. Isso foi possível graças ao surgimento da segunda geração de comunidades e serviços tendo a web como plataforma, a Web 2.0.

Web 2.0 surgiu através da utilização do termo em uma conferência entre as empresas *O'Reilly Media*² e *MediaLive International*³, e a partir de então, o termo se popularizou (O'REILLY, 2005).

O'Reilly (2006) conceitua Web 2.0 como a revolução de negócios na indústria de informática causada pela mudança para a Internet como plataforma, e uma tentativa de entender as regras para o sucesso nessa nova plataforma. A principal regra, entre outras, é o desenvolvimento de aplicativos que aproveitem os efeitos de redes para se tornarem melhores, à medida que são utilizados por um número maior de pessoas, aproveitando a inteligência coletiva. Vários sistemas de conhecimento colaborativo foram surgindo, aproveitando, então, o conceito de inteligência coletiva.

Singh e Gupta (2009) definem inteligência coletiva como a habilidade de um grupo de simples pessoas trabalhando juntas para resolver problemas maiores do que poderiam ser resolvidos por uma única pessoa.

O conceito de inteligência coletiva é utilizado antes mesmo da Internet. Os censos são as formas mais simples disso. Através de um censo, é possível coletar dados de grupos de pessoas. Ao final, esses dados podem ser combinados e analisados. O resultado de um censo pode ser mais satisfatório, pois permite tirar conclusões estatísticas onde pessoas individuais podem não ter esse conhecimento (SEGARAN, 2008).

Após o surgimento da Web 2.0, houve uma explosão na utilização do conceito de inteligência coletiva. Segaran (2009) afirma que a utilização da rede para coletar informações de usuários tem aberto muitas novas possibilidades.

² <http://oreilly.com/>

³ <http://www.medialiveintl.com>

Sites como a Wikipédia (Figura 7), uma enciclopédia online livre disponível em vários idiomas, fazem o uso do conceito de inteligência coletiva. Quase todo seu conteúdo é feito e atualizado por usuários comuns, mediante a aprovação de administradores do site. Em geral, a maioria de seu conteúdo é preciso (SEGARAN, 2008).

The image shows a screenshot of the Wikipedia article "Inteligência emergente" in Portuguese. The page layout includes a top navigation bar with user links (MarceloNishi, Discussão, Preferências, Páginas vigiadas, Contribuições, Sair) and a search bar. The article title is "A editar Inteligência emergente". Below the title, there is a warning box with the heading "Seguiu uma hiperligação para um artigo que ainda não existe." and a list of guidelines for editing Wikipedia articles. A red-bordered box highlights a section titled "Atenção! Esta é uma página do domínio principal (artigo)!" with a list of rules for editing, such as not copying text from other sources, including sources, and not using "internetês".

Figura 7 - Wikipédia
Fonte: <http://pt.wikipedia.org>.

Outro aplicativo que faz o uso do conceito de inteligência coletiva é o WikiCrimes (Figura 8). O objetivo desse aplicativo é permitir pesquisar, visualizar e registrar ocorrências de crimes em um mapa. Qualquer pessoa pode pesquisar e visualizar, porém, para registrar ocorrências de crimes, é necessário haver um cadastro no sistema. As informações de crimes são resultantes do uso da inteligência coletiva, onde cada usuário registra os crimes cometidos em determinado ponto e essas informações são mostradas em um mapa. O WikiCrimes está disponível em cinco idiomas: português brasileiro, inglês, francês, espanhol e italiano. Qualquer pessoa pode colaborar desde que tenha um cadastro no sistema.

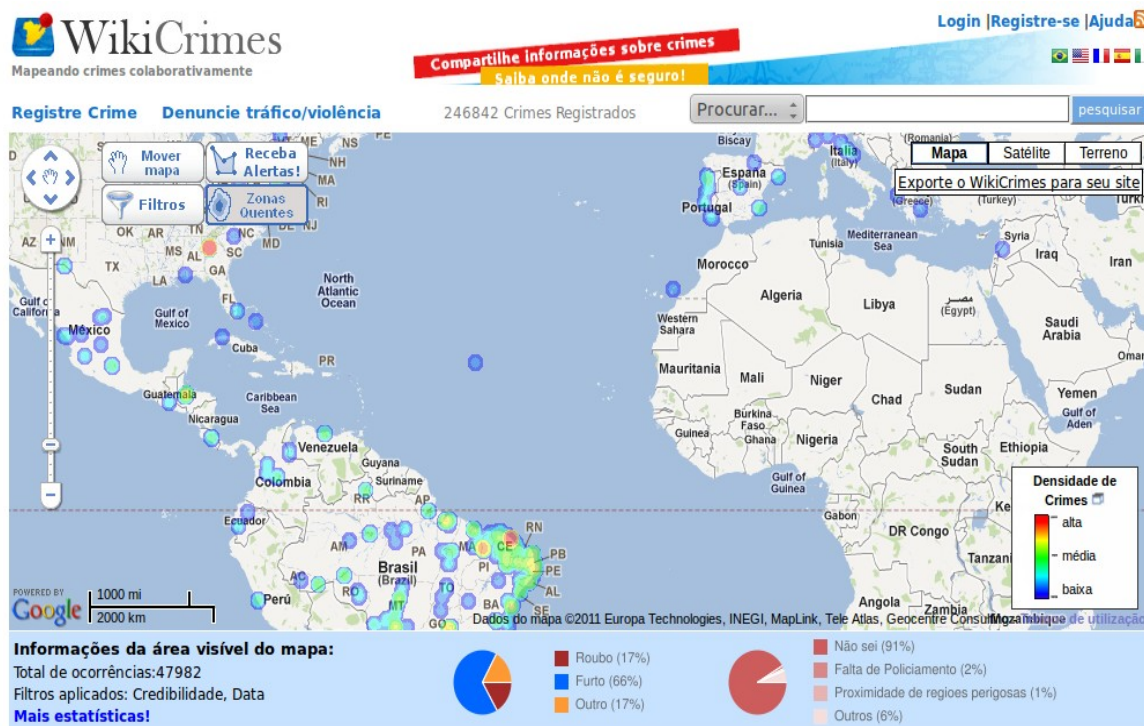


Figura 8 - WikiCrimes
Fonte: <http://www.wikicrimes.org>.

O aplicativo Waze (Figura 6) apresentado na seção 3.1, além de utilizar o conceito de ITS, faz uso de inteligência coletiva. Todas as informações das situações de trânsito são postadas por seus usuários através da própria aplicação.

Juntar os dois conceitos, ITS e inteligência coletiva, em um aplicativo para fornecer, em tempo real, informações sobre o trânsito de determinadas vias que afetam os usuários de transporte público é a proposta desse trabalho. Algumas tecnologias serão necessárias para alcançar essa proposta. Essas tecnologias serão apresentadas na Seção 3.3.

3.3 TECNOLOGIAS UTILIZADAS

O aplicativo apresentado neste trabalho fornece aos seus usuários uma interface web, pela qual os usuários podem ver informações da possível situação do trânsito de determinada linha de transporte público para uso em seus trajetos, através de um mapa digitalizado, fornecido pelo GoogleMaps. Para isso, foi utilizada

a linguagem de programação Java junto com a tecnologia JSF (*JavaServer Faces*)⁴ e a suíte de componentes JSF customizados, o PrimeFaces⁵. Para a persistência de dados, foi utilizada a JPA (*Java Persistence API*)⁶ juntamente com o *framework* EclipseLink⁷ e o SGBD (Sistema de Gerenciamento de Banco de Dados) MySQL⁸ para o gerenciamento do banco de dados.

A escolha das tecnologias foi feita com base no conhecimento maior dessas tecnologias pelo pesquisador.

O JSF é um padrão para a construção de interfaces de usuário do lado do servidor. As APIs (*Application Programming Interface* – em português, Interface de Programação de Aplicativos) do JSF são projetadas para facilitar o desenvolvimento de aplicativos web.

PrimeFaces é uma suíte de componentes JSF customizados de código aberto com suporte a Ajax (*Asynchronous Javascript and XML* – em português, Javascript e XML Assíncronos). Foi utilizado para melhorar a interface gráfica do usuário do aplicativo.

JPA é uma API do Java para a persistência de dados. Esta API deve ser implementada por um *framework* ORM (*Object-Relational Mapping* - em português, Mapeamento Objeto-Relacional). O *framework* que implementou a JPA foi o EclipseLink.

O *framework* EclipseLink, fornece uma estrutura para interação entre o desenvolvedor e vários serviços, entre eles, o banco de dados.

A integração do GoogleMaps com o JSF foi feita através da biblioteca Gmaps4JSF que fornece *tags* JSF para facilitar a criação de mapas.

Foram utilizadas as últimas versões estáveis das tecnologias citadas para a implementação do aplicativo.

⁴ <http://www.oracle.com/technetwork/java/javaee/javaserverfaces-139869.html>

⁵ <http://primefaces.org/>

⁶ <http://www.oracle.com/technetwork/articles/javaee/jpa-137156.html>

⁷ <http://www.eclipse.org/eclipselink/>

⁸ <http://www.mysql.com/>

4 ANÁLISE E PROJETO DO SISTEMA

Como descrito anteriormente no Capítulo 1, o projeto UbiBus foca em reunir várias soluções a fim de melhorar os serviços fornecidos aos usuários do transporte público no Brasil. A arquitetura do sistema UbiBus é apresentada na Figura 9.

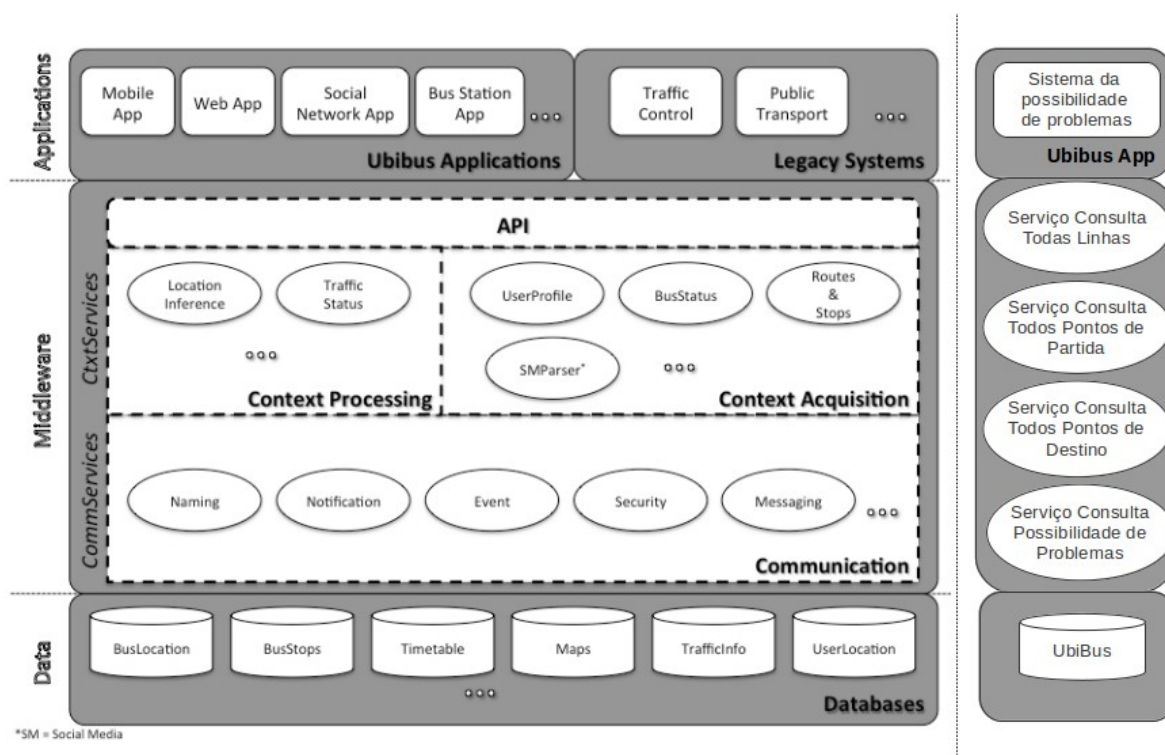


Figura 9 - Arquitetura UbiBus

Fonte: (VIEIRA; CALDAS; SALGADO, 2011).

Dentro dessa arquitetura, a camada de aplicações (*Applications*) concentra as diferentes aplicações que serão desenvolvidas para compor o UbiBus. O aplicativo deste trabalho deve ser integrado a essa camada, e terá como objetivo fornecer informações em tempo real sobre a situação do trânsito de determinadas vias que afetam usuários de transportes públicos, baseadas em dados de ocorrências registrados pelos usuários através da inteligência coletiva. Essas informações devem indicar trechos em que há maior ou menor possibilidade de haver problemas que afetem a eficiência dos transportes naquele trajeto.

Os dados das ocorrências utilizados por esse aplicativo são registrados pelos usuários de transporte público através de outros aplicativos pertencentes à camada de aplicações (*Applications*). Essas ocorrências são armazenadas em uma base de dados central, comum a todos os aplicativos do projeto UbiBus.

Cada ocorrência (Figura 10) contém informações como: mensagem escrita pelo usuário, data e horário que foi enviada essa mensagem, localização geográfica do problema relatado pela ocorrência, entre outros.

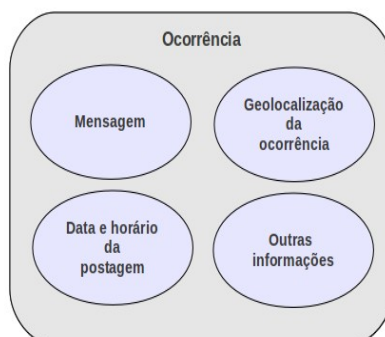


Figura 10 - Ocorrência
Fonte: Autoria própria.

O aplicativo funciona da seguinte forma: o usuário informa à aplicação a linha de ônibus para a qual deseja visualizar as informações e os pontos **A** (ponto de partida do usuário) e **B** (ponto de destino do usuário) dessa linha; o aplicativo recupera as informações sobre a possibilidade de problemas do intervalo requisitado e exibe o resultado através de uma interface gráfica baseada em mapas. As informações recuperadas são resultantes de um processo em *batch*, agendado, exibindo os resultados para os últimos 10 minutos. Este processo é disparado então, de 10 em 10 minutos, funcionando da seguinte forma (Figura 11): recupera da base de dados todos os trechos; de cada trecho, recupera todas as suas ocorrências; analisa as ocorrências verificando se são atuais (para o propósito desse trabalho, são consideradas atuais ocorrências, aquelas relatadas nos últimos 60 minutos); as mensagens de cada ocorrência são analisadas e classificadas de acordo com as palavras-chave (Subseção 4.1); um algoritmo é responsável por somar os pesos de todas as mensagens de acordo com a classificação das mensagens e analisar esse número de acordo com a escala de valores padrão definida; em seguida, o resultado é atribuído a uma variável utilizada para exibir as informações.

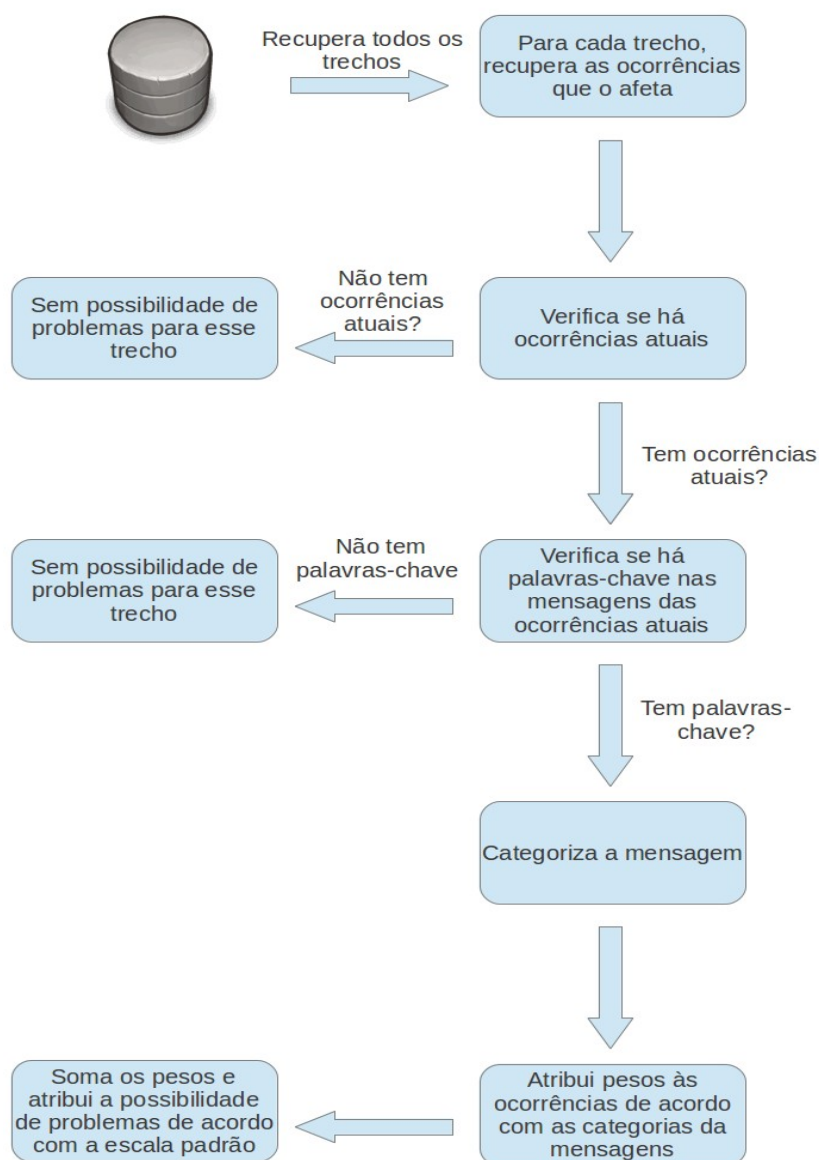


Figura 11 - Processo Batch
Fonte: Autoria própria.

A próxima seção (Seção 4.1) apresentará detalhadamente como é feita a análise das mensagens.

4.1 ANÁLISE DAS MENSAGENS

Esta subseção apresenta como é feita a análise das mensagens que foram postadas por usuários de transportes públicos.

As mensagens de ocorrências extraídas da base de dados são analisadas, verificando se contém as palavras-chave (Tabela1): acidente, atropelamento, enchente, alagamento, inundação, congestionamento, engarrafamento, trânsito parado, trânsito lento e ônibus quebrado. Caso contenha uma dessas palavras-chave, as mensagens são agrupadas em categorias e pesos são atribuídos a elas, para auxiliar o processamento das informações sobre a possível situação do trânsito. Caso contrário, para propósito desse trabalho, elas são descartadas. Vale ressaltar que, até este momento, todas as mensagens de ocorrências são consideradas problemas, mesmo aquelas que relatam a ausência de problemas. Pesquisas relacionadas à análise de sentimentos serão desenvolvidas como trabalhos futuros para corrigir esse problema, com o objetivo de classificar as mensagens de acordo com o seu conteúdo semântico e aumentar a precisão dos resultados da aplicação. A análise de sentimentos, de acordo com Liu (2010), é o estudo de opiniões, sentimentos e emoções expressas em textos. Outras limitações deste trabalho serão discutidas na Seção 6.

Tabela 1 - Porcentagem de mensagens retornadas com relação à problema no trânsito

Categoria	Palavras-chave e sua porcentagem	Média	Peso
Acidente	Acidente (74%) e atropelamento (74%)	74%	4
Enchente	Enchente (100%), alagamento (80%) e inundação (50%)	76%	8
Congestionamento	Congestionamento (82%), engarrafamento (80%), trânsito parado (98%) e trânsito lento (100%)	90%	16
Outros problemas	Ônibus quebrado (70%)	70%	2

Fonte: Autoria própria.

As palavras-chave foram escolhidas através de uma pesquisa manual pela rede social Twitter⁹, a fim de descobrir quais palavras as pessoas frequentemente usam para postarem ocorrências relacionadas ao trânsito. A pesquisa foi feita através de consultas de cada palavra-chave com suas variações. Foram selecionadas palavras em que, dos cinquenta primeiros *tweets* retornados, mais de 50% estavam relacionados ao trânsito. Outras palavras foram descartadas, pois dos cinquenta primeiros *tweets* retornados, houve um resultado abaixo de 50% com relação ao trânsito. Em seguida, as palavras-chave foram separadas para que as semelhantes pudessem ser agrupadas. A Tabela 1 mostra a porcentagem de retorno de cada palavra-chave e as categorias na qual foram separadas.

⁹ <http://twitter.com>

Para cada categoria de mensagem foram definidos pesos de acordo com a porcentagem de mensagens com relação ao trânsito retornadas na pesquisa pelo Twitter. Por exemplo, a categoria **Congestionamento** teve maior porcentagem de retorno com relação ao trânsito, assim, o peso atribuído a essa categoria foi maior. As outras categorias tiveram seus pesos atribuídos pela mesma metodologia aplicada à categoria **Congestionamento**.

Em seguida, foi estipulada uma escala de valores (Tabela 2), para indicar a quantidade de ocorrências que informam a possibilidade de haver problemas. A escala de valores foi desenvolvida, inicialmente, em um intervalo numérico de zero a sessenta, com base nas simulações feitas para o cálculo da possibilidade de problemas, baseadas na quantidade de 100 *tweets* (mensagens de ocorrências). Inicialmente, essa escala, bem como os pesos determinados para as categorias de mensagens, é estática e determinada apenas com base na percepção do pesquisador. Futuramente, pesquisas serão desenvolvidas dentro do projeto UbiBus com o intuito de criar algoritmos capazes de determinar valores para a escala e para os pesos dinamicamente, com base na análise histórica dos dados sobre as ocorrências ou outros fatores externos que possam impactar nesses valores.

Tabela 2 - Escala de valores

Total (peso)	Possibilidade
0	Nenhuma
2 à 19	Baixa
20 à 59	Média
60 para cima	Alta

Fonte: Autoria própria.

A próxima seção explicará detalhadamente como foi feita a implementação da aplicação.

5 IMPLEMENTAÇÃO DA APLICAÇÃO

Como descrito anteriormente no Capítulo 4, os dados das ocorrências são recuperados a partir de uma base de dados central, comum a todos os aplicativos do projeto UbiBus. Para a implementação do aplicativo, houve a necessidade de criar uma base de dados para simulação, uma vez que a base de dados do projeto UbiBus não estava definida para ser utilizada. A base de dados criada para simulação é mostrada graficamente na Figura 12.

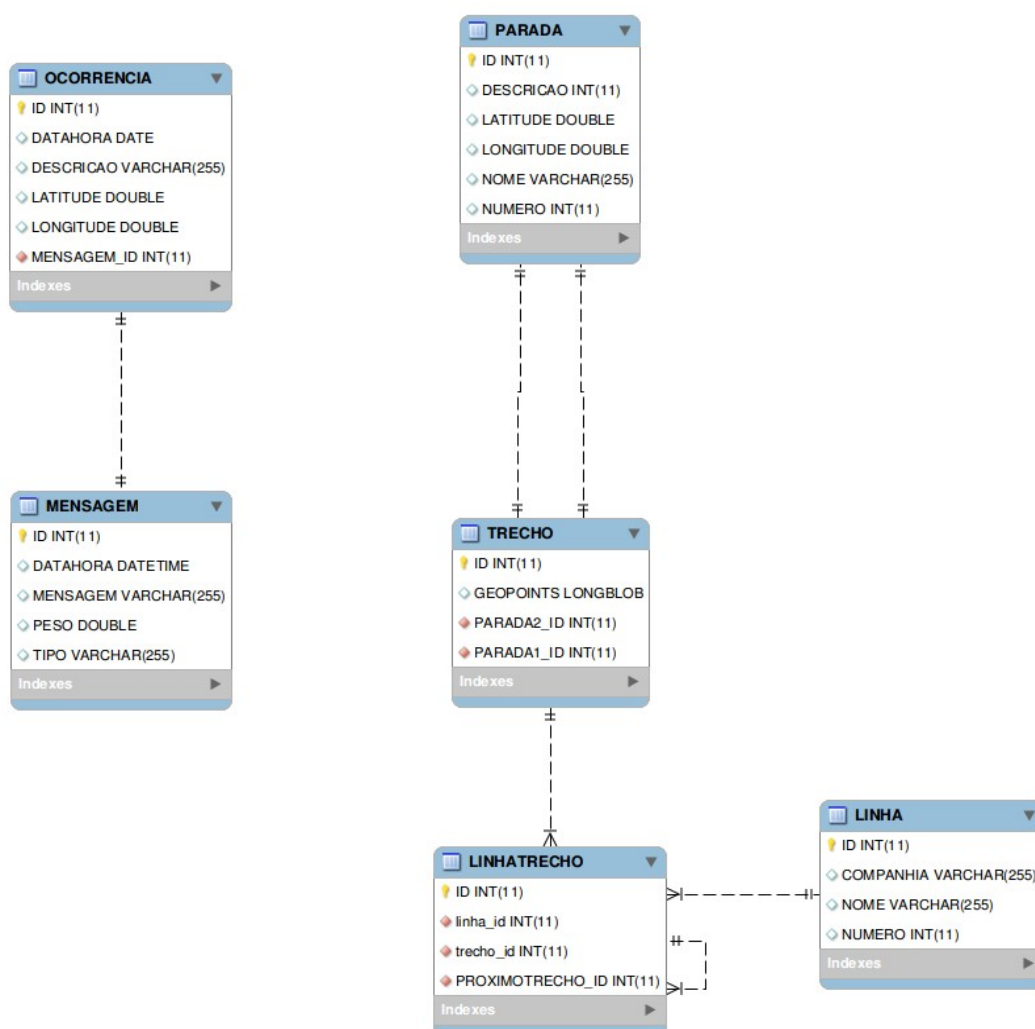


Figura 12 - Modelo de dados
Fonte: Autoria própria.

O modelo criado foi feito de acordo com o modelo esboçado pelo projeto UbiBus, utilizando somente as entidades e os atributos necessários ao aplicativo deste trabalho.

A entidade OCORRENCIA representa uma ocorrência. Cada ocorrência contém uma data e hora da postagem da ocorrência, uma descrição, a coordenada geográfica do ponto onde aconteceu o problema e uma mensagem, representada pela entidade MENSAGEM.

Cada MENSAGEM contém a data e a hora de quando foi enviada, a mensagem descrita pelo usuário, o peso que é utilizado para calcular a possibilidade de problemas e o tipo da mensagem obtido através do algoritmo que categoriza a mensagem de acordo com as palavras-chave.

Para simular as linhas de ônibus, foi criada a entidade LINHA, contendo o nome da companhia, o nome da linha, o número e os trechos que a compõem.

Um TRECHO é composto de uma parada inicial e uma final, ou seja, de um ponto de ônibus ao outro. Cada trecho contém vários pontos de coordenadas geográficas, que determinam em qual ponto do trecho há uma mudança de direção.

Após definir o modelo de dados, deu-se início ao desenvolvimento do núcleo da aplicação, utilizando a linguagem de programação Java e seguindo o modelo arquitetural MVC (*Model-view-controller*). Primeiramente foram criados os POJOs (*Plain Old Java Objects*) seguindo o padrão *JavaBean*. Para cada classe foi criado um POJO. Após, houve a implementação dos DAOs (*Data Access Object*) para acesso aos dados, e a implementação dos controladores, responsáveis pelas regras de negócios e a interação entre a interface gráfica e o acesso aos dados. Em seguida, iniciou-se a implementação da interface gráfica utilizando o JSF e a suíte de componentes PrimeFaces. Por fim, para integrar o sistema ao projeto UbiBus, houve a necessidade da implementação de serviços web, podendo ser visto detalhadamente na Subseção 5.1.

Os POJOs criados são (Figura 13): GeoPoint, Linha, LinhaTrecho, Mensagem, Ocorrencia, Parada, TipoMensagem e Trecho.

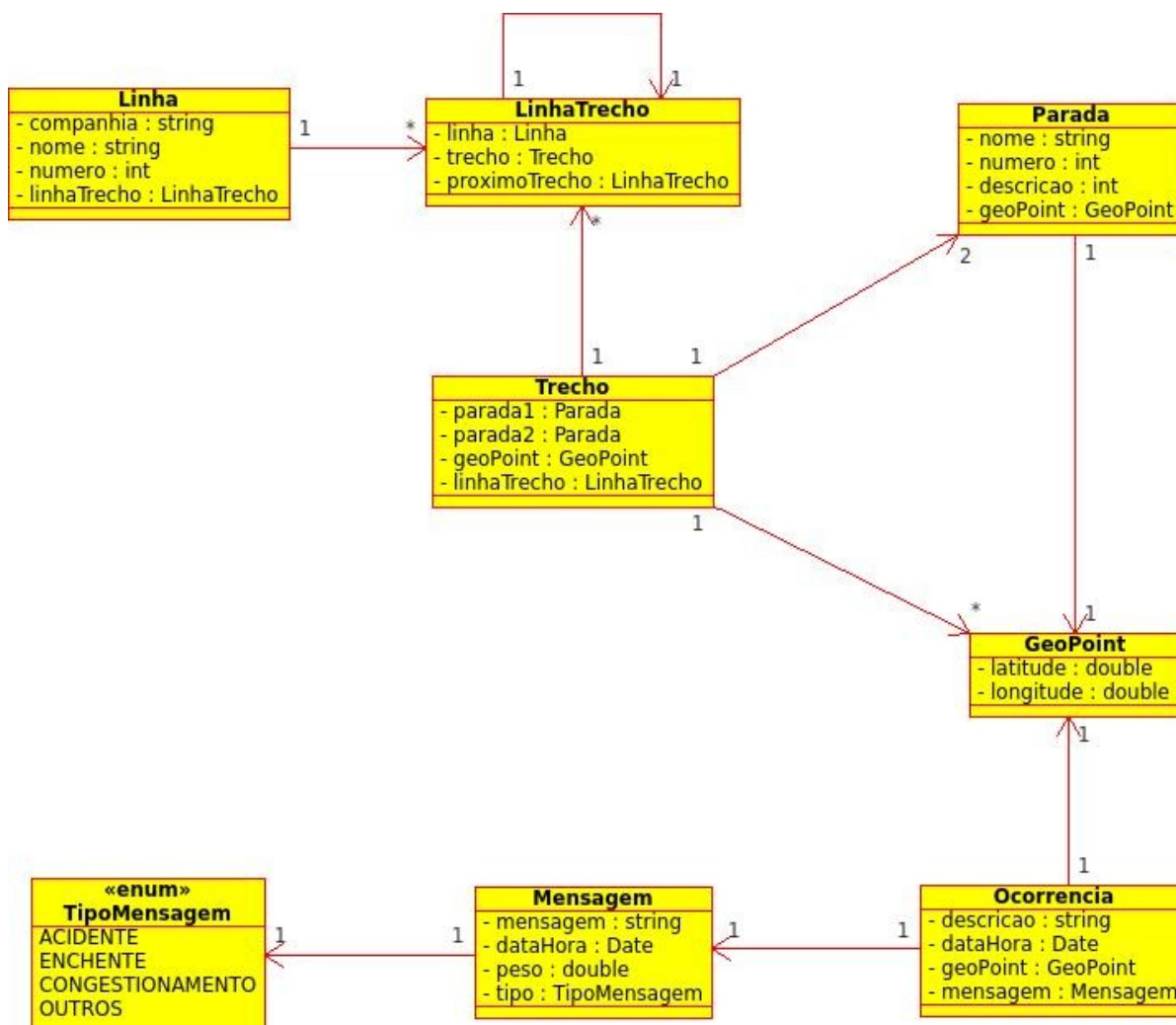


Figura 13 - Diagrama de classes
Fonte: Autoria própria.

A classe `GeoPoint` foi implementada para ajudar a trabalhar com pontos de coordenadas geográficas. Nesta classe existem os campos `latitude` e `longitude` ambos do tipo `double`, formando, assim, um ponto de coordenadas geográficas.

Como uma linha é formada por vários trechos, e estes trechos podem estar em mais de uma linha (Figura 11), houve a necessidade de saber qual seria o próximo trecho de determinado trecho de uma linha. A solução encontrada foi a implementação da classe `LinhaTrecho`, que contém os campos `linha` do tipo `Linha`, `trecho` do tipo `Trecho` e `proximoTrecho` do tipo `LinhaTrecho`. Desta forma, quase todos os trechos das linhas contém seu respectivo trecho seguinte. Os trechos que não contém seu respectivo trecho, são os últimos trechos de uma linha não circular.

Houve, ainda, a necessidade de criar um algoritmo capaz de identificar quais ocorrências afetam determinados trechos. Como um trecho é formado por vários pontos de coordenadas geográficas, que indicam onde há uma mudança de direção, e uma ocorrência contém o ponto geográfico exato do local onde ocorreu, um algoritmo teria que verificar se a ocorrência está dentro de cada seguimento (de um ponto de coordenada geográfica ao outro) do trecho.

De acordo com o Teorema de Tales, duas retas paralelas cortadas por uma transversal formam ângulos correspondentes iguais. Sabe-se ainda que o coeficiente angular de uma reta é a tangente de seu ângulo de inclinação (WIKIPEDIA). Assim, dado um ponto qualquer, pode-se calcular se este ponto encontra-se próximo a uma reta em um plano cartesiano utilizando a fórmula (1) abaixo, que corresponde a diferença entre o ângulo de inclinação de um seguimento e o coeficiente angular desse ponto (uma ocorrência).

$$\text{valor} = \text{incl} - \text{ca} \quad (1)$$

Para calcular o ângulo de inclinação (*incl*) de uma reta em um plano cartesiano foi utilizada a fórmula (2). A razão entre a diferença entre o ponto *x* inicial e o *x* final e a diferença entre o ponto *y* inicial e o *y* final, ambos da reta, resulta no valor do ângulo de inclinação. Substituindo esta reta por um seguimento de um trecho, pode-se, então, calcular qual é o ângulo de inclinação do trecho.

$$\text{incl} = \frac{(x_{\text{fim}} - x_{\text{início}})}{(y_{\text{fim}} - y_{\text{início}})} \quad (2)$$

Para calcular o coeficiente angular de um ponto qualquer (nesse caso, uma ocorrência), foi aplicada a fórmula (3). Esse cálculo é feito através da razão entre a diferença entre o ponto *x* da ocorrência e o ponto *x* fim do seguimento e a diferença entre o ponto *y* da ocorrência e o ponto *y* fim do seguimento.

$$\text{ca} = \frac{(x_{\text{ocorrência}} - x_{\text{fim}})}{(y_{\text{ocorrência}} - y_{\text{fim}})} \quad (3)$$

Calculada a diferença entre a inclinação do seguimento e o coeficiente angular, obtém-se como resultado um valor entre 0 e 1. Caso o valor seja menor que a tolerância aplicada, então o ponto está próximo ao seguimento. De acordo com simulações feitas, a tolerância utilizada é de 0,5. As simulações foram realizadas pelo pesquisador através de testes realizados pela interface baseada em mapa.

Através de ocorrências fictícias adicionadas pertos de trechos em uma linha simulada, foram realizados ajustes no valor da tolerância até alcançar um resultado satisfatório, onde a distância entre o seguimento e a ocorrência é tolerável, não atingindo outros seguimentos ao redor.

Assim, pode-se concluir se uma ocorrência afeta um determinado seguimento de um trecho.

5.1 INTERFACE GRÁFICA

A interface gráfica da aplicação é composta de três telas entre o início da requisição do usuário e o resultado retornado à interface. Na primeira tela, o usuário escolhe a linha sobre a qual deseja ver a informação. Em seguida, o usuário informa o ponto de partida e o ponto de chegada dessa linha. Finalmente, na terceira tela, o aplicativo retorna o resultado da possibilidade de problemas no trajeto requisitado, em forma de mapa de calor. As Figuras 13, 14 e 15 mostram esses passos.



Figura 14 - Tela de escolha da linha
Fonte: Autoria própria.



Figura 15 - Tela de escolha dos pontos de origem e destino
Fonte: Autoria própria.



Figura 16 - Tela de resultado
Fonte: Autoria própria.

Como descrito anteriormente na Seção 4, o resultado da possibilidade de problemas demonstrado na Figura 15 é obtido através de um processo em *Batch*. Foi criada a classe *Batch* (Figura 16), que contém os campos *delay* e *period* do

tipo `int`, `timer` do tipo `Timer` e o campo `todosProblemas` do tipo `Map`. Essa classe contém um método denominado `executa()`, que atualiza o campo `todosProblemas` com todas as possibilidades de problemas atuais cadastradas no banco de dados, utilizado para mostrar o resultado. Nesse método, é utilizado o método `schedule()` do campo `timer` responsável por agendar uma determinada tarefa e executá-la repetidamente de tempo em tempo. O `schedule()` espera como parâmetro um objeto do tipo `TimerTask`, um tempo de atraso (`delay`) que, após este tempo, será executada a tarefa e um período (`period`) que separa os intervalos de execução. No objeto do tipo `TimerTask` é onde de fato é executado o algoritmo responsável por calcular a possibilidade de problemas. Esta classe é abstrata, logo, é necessário sobrescrever o método `run()`, implementando a tarefa que será agendada. No método `run()`, então, é chamado o método que calcula todas as possibilidades de problemas atuais e atribui o resultado ao campo `todosProblemas`.

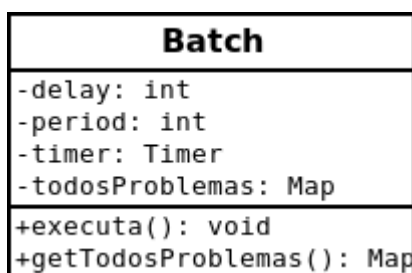


Figura 17 - Classe batch
Fonte: Autoria própria.

O mapa de calor apresenta as possibilidades através de cores nos trechos que estão entre o ponto de origem e destino escolhidos daquela linha. Como exemplo, foi simulada uma linha que contém três trechos (Figura 15). A cor amarela mostra que a possibilidade de problema é baixa para aquele trecho; a cor vermelha informa que a possibilidade é média; e a cor verde indica que não há relatos de possíveis problemas naquele trecho.

A aplicação disponibiliza alguns serviços web, mas, apesar de implementados, o aplicativo não os utiliza até o momento. No período de desenvolvimento deste trabalho, a arquitetura do projeto UbiBus não havia sido definida completamente. Desta forma, a aplicação foi implementada utilizando uma base de dados simulada através do esboço da base de dados do projeto UbiBus e sem a utilização de serviços. Assim que a arquitetura do projeto UbiBus foi definida,

perto do fim deste trabalho, optou-se por utilizar serviços web na camada de *middleware* (Figura 9). Assim, foram implementados e disponibilizados os serviços web necessários à aplicação, porém, não houve tempo hábil para fazer as mudanças necessárias na aplicação para utilizá-los. Os serviços web serão utilizados na integração do sistema ao projeto UbiBus em trabalhos futuros.

5.2 SERVIÇOS WEB

O aplicativo apresentado neste trabalho será integrado ao projeto UbiBus, que prevê o desenvolvimento de várias aplicações para facilitar o acesso à informações do transporte público aos usuários. Para facilitar esta integração, foi definido que, na camada de *middleware* do projeto UbiBus (Figura 9), deverão ser utilizados serviços que ficarão disponíveis para que outras aplicações desse projeto possam utilizá-los. Então, foram desenvolvidos serviços web e disponibilizados, mas, não houve tempo hábil para fazer as mudanças necessárias na aplicação para utilizá-los.

São disponibilizados quatro serviços web:

- serviço de consulta de todas as linhas: este serviço, quando requisitado, retorna todas as linhas de ônibus cadastradas no banco de dados. Assim, será possível utilizá-lo para listar todas as linhas ao usuário, permitindo a ele escolher qual a linha que deseja pesquisar a possibilidade de problemas. O serviço não necessita de entradas na requisição e o retorno são todas as linhas de ônibus cadastradas no banco de dados;
- serviço de consulta de todos os pontos de partida de determinada linha: o serviço fornece todos os pontos de partidas cadastradas no banco de dados para uma determinada linha. Será utilizado para listar todos os pontos de partida de uma linha ao usuário, assim, o usuário pode escolher qual o ponto inicial da consulta de possibilidade de problemas. Este serviço espera como entrada uma linha, e, a partir da linha, retorna todos os pontos de partida da mesma;
- serviço de consulta de todos os pontos de destino: fornece todos os pontos de destino a partir de uma linha e um ponto de partida. Desta forma, é

possível listar ao usuário todos os pontos de destino de acordo com a linha e o ponto de partida escolhidos. O serviço espera como entrada uma linha e um ponto de partida, retornando todos os pontos de destino.

- serviço de consulta da possibilidade de problemas: este serviço retorna a possibilidade de problemas de determinado trecho de uma linha. Deve ser utilizado para informar ao usuário a possibilidade de problemas em trechos de uma linha de acordo com os parâmetros passados a este serviço. Os parâmetros necessários, ou seja, as entradas esperadas são: uma linha, um ponto de partida e um ponto de destino. A saída deste serviço é o resultado da possibilidade de problemas.

Estes serviços estão disponíveis para serem integrados à aplicação, porém, como descrito anteriormente, a implementação atual ainda não os utiliza.

A utilização dos serviços apresentados nesta seção será feita na integração da aplicação deste trabalho à arquitetura do projeto UbiBus, em trabalhos futuros.

6 CONCLUSÕES E TRABALHOS FUTUROS

Neste trabalho, foi apresentado um aplicativo que permite aos usuários consultarem a possibilidade de problemas em determinadas vias que afetam os transportes públicos. Utilizar a inteligência coletiva para exibir informações relacionadas ao trânsito, juntamente com o conceito de ITS, possibilita aos usuários tomarem melhor suas decisões no momento que estão planejando suas jornadas pelo transporte público. Futuramente, este aplicativo será integrado ao projeto UbiBus.

O trabalho apresentado é apenas o início de um recurso que utiliza os conceitos de ITS e inteligência coletiva para o projeto UbiBus e muitas pesquisas ainda serão realizadas. Estas pesquisas serão desenvolvidas dentro do projeto UbiBus focando nas limitações encontradas até este momento como:

- palavras-chave: a pesquisa realizada para obter as palavras-chave utilizadas para classificar as mensagens não foram exaustivas, podendo não ter encontrado palavras relevantes para o estudo. Além disso, o algoritmo busca apenas pelos termos exatos nas mensagens. Para melhorar a eficiência do aplicativo, os radicais de cada palavra-chave deveriam ser utilizados, aumentando a cobertura dos resultados de mensagens relacionadas ao trânsito. Pesquisas mais aprofundadas serão realizadas para obter um melhor grupo de palavras-chave e para utilizar análise textual para buscá-las por meio de radicais;

- análise das mensagens: até este momento, todas as mensagens que contêm as palavras-chave são consideradas problemas de trânsito, podendo haver mensagens que não apresentam de fato problemas, afetando a eficiência do resultado no algoritmo que faz a análise das possibilidades de problemas. Por exemplo, uma mensagem pode estar descrita desta forma: "Sem congestionamento na Av. Brasil", como até este momento não há análise de sentimentos, o algoritmo interpreta esta mensagem como um problema. Serão realizados estudos para aplicar análise de sentimentos nas mensagens que contêm as palavras-chave, a fim de identificar apenas as que realmente afetam o fluxo do trânsito, melhorando a eficiência dos resultados do aplicativo;

- classificação das mensagens: a classificação das mensagens, assim

como os pesos atribuídos para cada categoria da classificação são dependentes das palavras-chave, desta forma, a partir das pesquisas que serão realizadas para obtenção das melhores palavras-chave, afetarão a classificação e os pesos das categorias. A partir disso, a classificação e os pesos serão melhor elaborados de acordo com o melhor grupo de palavras-chave;

- escala de valores: os valores atribuídos à escala de valores neste momento são estáticos, não levando em consideração o histórico dos dados que são analisados para os resultados da possibilidade de problemas. Serão realizadas pesquisas para criar algoritmos que possam determinar dinamicamente, de acordo com a análise histórica dos dados, os valores desta escala, assim, melhorando a eficiência dos resultados da possibilidade de problemas;

- testes com dados simulados: todos os testes realizados foram feitos através de dados simulados como linhas, trechos e ocorrências. Após a integração da aplicação ao projeto UbiBus, serão utilizados dados reais, possibilitando comparações dos resultados da possibilidade de problemas resultantes de testes com dados reais com informações de trânsito oferecidas em outras fontes, como o GoogleMaps, a fim de ajustar os pesos e melhorar a visualização.

Atualmente o aplicativo está em fase de integração ao projeto UbiBus, onde o banco de dados está sendo mudado do MySQL para o PostgreSQL/GIS, banco de dados utilizado pelo projeto. Dados reais da cidade de João Pessoa, Paraíba estão sendo utilizados para auxiliar nos testes da visualização.

Assim, será possível, através da inteligência coletiva, auxiliar usuários de transporte público, oferecendo informações que facilitem o seu dia a dia no trânsito.

Este trabalho possibilitou a publicação de um artigo intitulado "Um Sistema Baseado em Inteligência Coletiva para Visualização de Problemas em Vias Públicas"¹⁰, no IX Simpósio Brasileiro de Sistemas Colaborativos (SBSC), onde se discute o desenvolvimento e uso de ferramentas para dar suporte à colaboração entre pessoas.

¹⁰ "NISHI, M. K. ; STEINMACHER, I. F. ; CHAVES, A. P. . Um Sistema Baseado em Inteligência Coletiva para Visualização de Problemas em Vias Públicas. In: Simpósio Brasileiro de Sistemas Colaborativos (SBSC), 2012, São Paulo. Simpósio Brasileiro de Sistemas Colaborativos (SBSC 2012), 2012."

REFERÊNCIAS

BARRY, Keith; **How smarthphones can improve public transit**, *Wired Magazine*, Londres, abr. 2011. Disponível em: <<http://www.wired.com/autopia/2011/04/how-smartphones-can-improve-public-transit/>>. Acesso em: 25 ago. 2011.

CAULFIELD, Brian; O'MAHONY, Margareth. An examination of the public transport information requirements of users. In: IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEM, vol. 8, n. 1, mar 2007. **Anais eletrônicos...** pp.21-30.

CHAVES, Ana P.; STEINMACHER, Igor; VIEIRA, Vaninha. Social network and colletive intelligence applied to public transportation systems: a survey. In: SIMPÓSIO BRASILEIRO DE SISTEMAS COLABORATIVOS, 2011, Parati, RJ, **Anais eletrônicos...**, Parati: SBSC, 2011.

O'REILLY, Tim; **Web 2.0 Compact Definition: Trying Again**, dez. 2006. Disponível em: <<http://radar.oreilly.com/2006/12/web-20-compact-definition-tryi.html>>. Acesso em: 13 set. 2011.

O'REILLY, Tim; **What Is Web 2.0**, set. 2005. Disponível em: <<http://oreilly.com/web2/archive/what-is-web-20.html>>. Acesso em: 13 set. 2011.

SEGARAN, Toby. **Programando a inteligência coletiva: desenvolvendo aplicativos inteligentes web 2.0**. Rio de Janeiro: Alta Books, 2008.

SHAH, Akhtar Ali; DAL, Lee Jong. Intelligent transportation systems in transitional and developing countries. In: INTERNATIONAL CONFERENCE ON ADVANCES IN SPACE TECHNOLOGIES, 2006. **Anais eletrônicos...** pp.54-59.

SIGH, V. K.; GUPTA, A. K. From artificial to collective intelligence: perspectives and implications. In: INTERNATIONAL SYMPOSIUM ON APPLIED COMPUTATIONAL INTELLIGENCE AND INFORMATICS, 5, 2009. **Anais eletrônicos...** pp.545-550.

VIEIRA, V.; SALGADO, A. C.; TEDESCO, P.; TIMES, V. C.; FERRAZ, C.; HUZITA, E.; CHAVES, A. P.; STEINMACHER, I.. The UbiBus project: using context and ubiquitous computing to build advanced public transportation systems to support bus passengers. In: VIII SIMPÓSIO BRASILEIRO DE SISTEMAS DE INFORMAÇÃO, 2011, São Paulo, **Anais eletrônicos...**

WIKICRIMES. Mapeando crimes colaborativamente. Disponível em: <<http://www.wikicrimes.org/>>. Acesso em: 14 set. 2011.

WIKIPÉDIA. **Teorema de Tales (interseção).** Disponível em: <http://pt.wikipedia.org/wiki/Teorema_de_Tales_%28interse%C3%A7%C3%A3o%29>. Acesso em: 20 ago. 2012.

YOKOTA, Toshiyuki. ITS for developing countries. **ITS Technical Note** , v. 1, jul 2004. Disponível em: <<http://www.worldbank.org/transport/roads/its%20docs/ITS%20Note%201.pdf>>. Acesso em: 26 ago. 2011.

ZHAO, Yilin. **Vehicle location and navigation systems.** Norwood, MA: Artech House, 1997.