



UNIVERSIDADE FEDERAL DE PERNAMBUCO  
CENTRO DE INFORMÁTICA

GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

# **INDICAÇÃO DE POSSÍVEIS ROTAS DE ÔNIBUS BASEADAS EM INFORMAÇÕES CONTEXTUAIS**

Filipe de Moraes Rêgo Magalhães

Trabalho de Graduação

RECIFE,  
12 DE DEZEMBRO DE 2011

UNIVERSIDADE FEDERAL DE PERNAMBUCO  
CENTRO DE INFORMÁTICA

GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

FILIFE DE MORAES RÊGO MAGALHÃES

## **INDICAÇÃO DE POSSÍVEIS ROTAS DE ÔNIBUS BASEADAS EM INFORMAÇÕES CONTEXTUAIS**

**Trabalho de Conclusão de Curso  
apresentado pelo aluno Filife de  
Moraes Rêgo Magalhães sob a  
orientação da Professora Ana  
Carolina Salgado à Coordenação de  
Ciência da Computação da  
Universidade Federal de  
Pernambuco, como requisito para  
obtenção do título de Bacharelado  
em Ciência da Computação.**

RECIFE,  
12 DE DEZEMBRO DE 2011

Dedico este trabalho aos meus familiares e amigos, em especial aos meus pais, Petrônio Lira Magalhães e Regina de Moraes Rêgo.

## **AGRADECIMENTOS**

Agradeço primeiramente a Deus, por ter me dado a oportunidade de ingressar em uma instituição pública de alto nível de ensino. Aos meus pais, por me incentivarem do início ao fim, desde o vestibular até minha graduação, em momentos de tristeza e de alegria. À aluna de iniciação científica Vanessa Gomes Lima, que ajudou diretamente em todas as fases deste trabalho. À minha orientadora Ana Carolina Salgado, por estar sempre pronta e disposta a tirar qualquer tipo de dúvidas e orientar todo o trabalho. A todos os meus velhos amigos, por estarem comigo no dia a dia me dando forças. E, por fim, mas não menos importante, a todos os amigos que fiz durante o curso, que foram bastante importantes para o término da graduação, valeu Bala, Bibi, Cata, Drance, Fábulo, Laurete, Lili, Lulu, Maria, Panda, PT e Xico.

## RESUMO

O cenário atual do trânsito nas grandes cidades brasileiras vem piorando a cada dia que passa. Com o aumento do número de carros particulares nas vias, os congestionamentos tornaram-se cada vez mais frequentes. Não será possível melhorar o trânsito alargando ruas, construindo viadutos ou fazendo o rodízio de carros, a única forma para isto é melhorar o transporte público e mudar a cultura das pessoas para que passem a utilizá-lo. Este trabalho de graduação tem como objetivo a criação de um protótipo de um sistema web de indicação de rotas de ônibus levando-se em consideração informações contextuais capturadas de redes sociais. A ferramenta construída será de fácil acesso e irá levar em consideração o que estão falando sobre as ruas que os ônibus passarão, de forma que se houver um congestionamento muito grande, uma rota alternativa seja sugerida.

**Palavras chave:** ITS, Informações Contextuais, Transportes Públicos Inteligentes e Ubíquos.

## **ABSTRACT**

The current scenario of traffic in large Brazilian cities is getting worse every day. With the increasing number of private cars on roads, traffic jams have become increasingly frequent. It is not possible to improve the traffic widening streets, building bridges or making the carpooling, the only way to do this is to improve public transport and change the culture, so that people will start using it. This graduate work aims to create a prototype of a web system for indication of bus routes taking into account contextual information captured from social networks. The tool is built for easy access and will consider what they are talking about the streets that the bus pass, so if there is a great jam, an alternative route is suggested.

**Key words:** ITS, contextual information, Ubiquitous and Intelligent Public Transports

## Índice de Figuras

Figura 3.1 - Arquitetura de alto nível do sistema .....	20
Figura 4.1 - Modelo Conceitual das Paradas, Linhas e Informações Contextuais .....	24
Figura 4.2 - Modelo Lógico das Paradas, Linhas e Informações Contextuais .....	25
Figura 4.3 - Arquitetura do Subsistema de Comunicação com o Twitter .....	26
Figura 4.4 - Mapa com as paradas cadastradas no sistema. ....	31
Figura 4.5 - Parada selecionada e opções de marcar como origem ou destino. ....	32
Figura 4.6 - Mapa com as paradas escolhidas em destaque. ....	32
Figura 4.7 - Possíveis critérios de escolha de rotas. ....	32
Figura 4.8 - Mapa com a rota indicada.....	33
Figura 4.9 - Instruções a serem seguidas. ....	33
Figura 4.10 – <i>Tweet</i> de Informação de Trânsito.....	33
Figura 4.11 – Mudança no tempo total gasto na viagem.....	34
Figura 4.12 – Mudança na rota sugerida.....	34

# Sumário

1. Introdução .....	8
1.1. Contexto e Motivação.....	8
1.2. Problemas .....	9
1.3. Objetivos .....	9
1.4. Estrutura.....	10
2. Fundamentação .....	11
2.1. Expressões Regulares e Gramáticas Livres de Contexto .....	11
2.2. Definição de Contexto e Informação contextual .....	13
2.2.1. Onde encontrar Informações Contextuais? .....	14
2.3. ITS – Intelligent Transportation Systems.....	15
2.3.1. O que é ITS? .....	15
2.3.2. Quais são os ganhos da utilização de ITS?.....	16
2.3.3. Sugestão de Rotas .....	17
3. Especificação da Ferramenta.....	18
3.1. O que é? .....	18
3.2. O que faz?.....	18
3.3. Arquitetura do Sistema.....	19
3.4. Casos de Uso do Sistema.....	20
UC01 – Visualizar informações de uma Parada.....	20
UC02 – Selecionar parada de origem/destino.....	21
UC03 – Selecionar tipo de busca.....	21
UC04 – Realizar busca .....	22
4. Implementação da Ferramenta .....	23
4.1. Modelagens Utilizadas .....	23
4.1.1. Modelo conceitual.....	23
4.1.2. Modelo Lógico .....	24
4.2. Projeto do subsistema de comunicação com Twitter .....	25
4.2.1. Arquitetura do Subsistema .....	26
4.2.2. Como funciona?.....	26
4.2.3. Detalhes da Comunicação – API REST .....	28
4.3. Como foi feito? .....	29
4.4. Cenário com o Sistema .....	31
5. Conclusão .....	35
Referências .....	37



# 1. Introdução

## 1.1. Contexto e Motivação

O cenário atual do trânsito nas grandes cidades brasileiras vem piorando a cada dia que passa. Com o aumento do número de carros particulares nas vias, os congestionamentos tornaram-se cada vez mais frequentes. Não será possível melhorar o trânsito alargando ruas, construindo viadutos ou fazendo o rodízio de carros. A única forma para isto é melhorar o transporte público e mudar a cultura das pessoas para que passem a utilizá-lo [Zhang *et al.* 2011].

Com a Copa do Mundo de Futebol em 2014 e as Olimpíadas em 2016, o Brasil terá que solucionar este problema. Uma das propostas de solução é o objetivo do projeto UbiBus, um sistema de transporte público inteligente, ubíquo e sensível ao contexto, ao qual contribuiremos considerando a influência das informações contextuais na sugestão de rotas de transportes públicos para os usuários, a partir de dispositivos móveis.

As informações contextuais serão utilizadas neste trabalho por serem facilmente capturadas da Web, por serem abundantes e crescentes, por serem postas em tempo real e de qualquer lugar, por serem repetidas e confirmadas por várias pessoas, de forma que se tornem relevantes, entre outros [Magalhães 2008].

Porém, não é suficiente apenas capturar tais informações, é preciso que seja feito algo parecido com uma análise mais detalhada, por exemplo, a pergunta: “O que estão falando sobre a Avenida Caxangá?”, que é uma informação possível de ser obtida a partir de blogs, microblogs e redes sociais [Pak *et al.* 2010].

## **1.2. Problemas**

A solução dos problemas que estamos tratando pode ser resumida basicamente nas respostas de algumas perguntas. Porém, não queremos várias soluções separadas, e sim, uma solução única, integrada e que possa ser usada de qualquer lugar, por qualquer pessoa.

Alguns exemplos de perguntas que resumem os problemas são:

- Como saber quais ônibus são possíveis para ir de um ponto a outro em uma cidade?
- Qual a opção mais barata?
- Qual a opção mais rápida?
- Se houver um congestionamento em um lugar da cidade, que linhas de ônibus serão afetadas? E, neste caso, quais as rotas alternativas?

## **1.3. Objetivos**

Este trabalho de graduação tem como objetivo a criação de um protótipo de um sistema Web de indicação de rotas de ônibus. No protótipo, o usuário informa o ponto de partida e de chegada e o sistema calcula que ônibus são possíveis tomar para realizar este trajeto.

Para o cálculo dos possíveis ônibus, levam-se em consideração informações contextuais capturadas de redes sociais. Tais informações serão consideradas como pesos que influenciarão nas rotas de acordo com um padrão. Será avaliado se o que estão falando nas redes sociais sobre determinada rua afetará positiva ou negativamente as linhas de ônibus que passam por aquela rua.

Para isso, serão simuladas duas bases de dados, uma com os dados das linhas de ônibus, como paradas, preço, distâncias, entre outros. E na outra base, com as informações contextuais, serão armazenadas simulações de *tweets* em um formato de fácil interpretação. Além disso, serão usados algoritmos de busca em grafos, algoritmos de recuperação e mineração de dados na Web e para a análise de contexto, um dicionário de palavras que

remetam a frases com sentido de “trânsito muito congestionado” até “trânsito muito livre”.

#### **1.4. Estrutura**

Além deste capítulo introdutório com contextualização, motivação, problemas e objetivos, o presente trabalho é organizado na seguinte estrutura:

No Capítulo 2 será mostrada uma melhor fundamentação sobre os conceitos de expressões regulares, gramáticas livres de contexto e informações contextuais, o que são e onde capturá-las. Além de apresentar de forma um pouco mais aprofundada o que são os ITS (sistemas de transportes inteligentes), onde já estão presentes, quais os ganhos de usá-los, entre outros.

No Capítulo 3 será especificado o protótipo construído, o que é, o que faz, para quem e como. Ainda neste capítulo, serão mostrados, também, os casos de uso, a modelagem do sistema.

No Capítulo 4 será mostrado como foi feita a implementação de ambas as partes do projeto. Quanto às informações contextuais, mostraremos como capturá-las e como classificá-las, quanto ao sistema de sugestão, mostraremos os algoritmos envolvidos. Além disso, será mostrada a modelagem das bases de dados simuladas e os resultados obtidos nos testes realizados.

No último capítulo, será feita uma conclusão do que foi feito, uma análise sobre os resultados obtidos e uma sugestão de possíveis continuações para este projeto, assim como trabalhos futuros.

## 2. Fundamentação

No presente capítulo será mostrada uma visão geral sobre os conceitos de expressões regulares e gramáticas livres de contexto, que serão utilizadas para a captura das informações contextuais, que, por sua vez, também serão explicadas. Também será apresentado, de forma um pouco mais aprofundada, o que são os sistemas de transportes inteligentes, onde já estão presentes, quais os ganhos de usá-los, entre outros.

### 2.1. *Expressões Regulares e Gramáticas Livres de Contexto*

Na área da lógica e da informática teórica, são estudadas desde o início do século XX o que chamamos de expressões, gramáticas, linguagens, autômatos, etc. Sabe-se que há vários tipos de gramáticas e linguagens, mas tomaremos como exemplo um tipo especial de linguagem chamada Linguagem Livre de Contexto. Mas antes de chegarmos até ela, iremos mostrar a definição do que são Expressões Regulares.

Expressões regulares são utilizadas para descrever linguagens, estas expressões são formadas por operações regulares e elementos. Uma boa forma de se entender o que são operações regulares é fazendo um comparativo com a aritmética, onde temos, por exemplo, as operações de somar e multiplicar formando a expressão aritmética  $(5 + 3) \times 4$ . Na área da teoria da computação, exemplos de operações regulares são a união e a operação estrela que formam uma expressão do tipo  $(0 \cup 1)^*$ .

Nos exemplos vistos acima temos como resultado da expressão aritmética o número 32, enquanto que o resultado da expressão regular compreende todas as cadeias de 0's e 1's que obedeçam à regra de começar com um 0 ou um 1 concatenada com uma cadeia apenas de 0's de qualquer tamanho inclusive vazia.

Analisando as expressões regulares, percebemos que podemos criar algo um pouco mais trabalhado e que represente melhor o que queremos estudar. Esta ferramenta que será criada será uma gramática e que neste estudo será especificamente livre de contexto.

A gramática livre de contexto foi criada para descrever linguagens e com sua característica de recursividade, consegue ser útil em várias aplicações. No início, esta gramática era utilizada para estudar a linguagem humana, relacionando *nome*, *verbo* e *preposição*, por exemplo.

O termo “Livre de Contexto” é aplicado nestes exemplos para indicar que uma frase escrita nesta linguagem não terá mais de uma interpretação, não haverá ambiguidade. Por causa disso, estas gramáticas são usadas na especificação e compilação de linguagens de programação. Tomemos como exemplo a gramática  $G_1$  retirada do livro de [Sipser 2007]:

<FRASE> → <SUJEITO><PREDICADO>  
<SUJEITO> → <NOME-CMPLX> | <NOME-CMPLX><PREP-CMPLX>  
<PREDICADO> → <VERBO-CMPLX> | <VERBO-CMPLX><PREP-CMPLX>  
<PREP-CMPLX> → <PREPOSIÇÃO><NOME-CMPLX>  
<NOME-CMPLX> → <ARTIGO><NOME>  
<VERBO-CMPLX> → <VERBO> | <VERBO><SUJEITO>  
<ARTIGO> → um | uma | o | a  
<NOME> → menino | menina | flor  
<VERBO> → fala | está | vê  
<PREPOSIÇÃO> → com

Com  $G_1$ , conseguimos entender frases deste tipo:

**A menina fala com o menino**

**O menino vê a menina com uma flor**

**A menina está com uma flor**

Vimos que com  $G_1$  não há como interpretarmos de forma diferente o que está escrito acima. O que temos como gramática pode ser visto como uma

concatenação de conjuntos de elementos permitindo passos recursivos que se autorrepresentam.

## **2.2. Definição de Contexto e Informação contextual**

Para chegarmos ao contexto de uma linguagem, deveremos passar primeiro pelos conceitos de elemento contextual, contexto e sistemas sensíveis ao contexto vistos em [Vieira *et al.* 2009].

*Um elemento contextual (CE, do inglês Contextual Element) é qualquer dado, informação ou conhecimento que permite caracterizar uma entidade em um domínio.*

*O contexto da interação entre um agente e uma aplicação, para executar alguma tarefa, é o conjunto de elementos contextuais instanciados que são necessários para apoiar a tarefa atual.*

*Sistemas Sensíveis ao Contexto (CSS, do inglês Context-Sensitive Systems) são aqueles que gerenciam CEs relacionados a uma aplicação em um domínio e usam esses elementos para apoiar um agente na execução de alguma tarefa. Esse apoio pode ser alcançado pelo aumento da percepção do agente em relação à tarefa sendo executada ou pelo provimento de adaptações que facilitem a execução da tarefa*

Para um melhor entendimento, podemos exemplificar alguns elementos contextuais “está chovendo”, “Recife, 18h30 de uma quarta-feira” e uma tarefa “saber a situação do trânsito na Avenida Agamenon Magalhães”.

Para nós, que já somos acostumados com o trânsito de Recife, ao saber que está chovendo, que é uma quarta-feira e que são 18h30, já poderíamos pressupor que praticamente todas as avenidas da cidade estão engarrafadas. Já para alguém de fora, ficaria mais difícil de saber sobre o engarrafamento. Para um sistema, será preciso treiná-lo ou informá-lo sobre o trânsito de

Recife, em outras palavras, fazer com que ele aprenda com o contexto de condição do tempo e horário de pico, por exemplo.

Ao construirmos um sistema sensível ao contexto, deveremos analisar o que está em volta da situação, porém devemos saber o que analisar exatamente. Para o nosso projeto, analisamos o contexto a partir de frases capturadas de redes sociais, delas, identificamos como elementos contextuais, a hora, a localização e a situação do trânsito [Vieira *et al.* 2007].

Em um trabalho futuro, será preciso fazer uma análise histórica da base de informações capturadas para que o sistema fique sensível à hora e à localização. Além disso, outras informações contextuais deverão ser analisadas, como “quem está fazendo a requisição da rota?”. Com isso, teremos um perfil do usuário de acordo com preferências de linhas, de preços, de horários, entre outras.

### **2.2.1. Onde encontrar Informações Contextuais?**

Para este trabalho, será utilizado um perfil no Twitter [Twitter 2011] para a captura das informações contextuais, o @UbiBus\_PE. Por meio dele, postaremos informações sobre o trânsito da área que será estudada.

Os elementos contextuais que serão extraídos das informações capturadas serão o horário, a localização e a situação do trânsito. Porém, nesta fase do projeto, ainda estamos capturando as informações apenas de um perfil de uma rede social. Mais tarde, passaremos a capturar de mais perfis de mais redes, com isso, haverá mais elementos contextuais neste sistema. A reputação do perfil poderá ser considerada, a quantidade de vezes que a situação aparece também, além de outros fatores.

## **2.3. ITS – Intelligent Transportation Systems**

Neste tópico, será explicado o que significa Sistemas de Transporte Inteligentes, assim como os ganhos da sua utilização. Também será enfatizado o sistema implementado, o de recomendação de rotas, mostrando o que já está sendo feito atualmente.

### **2.3.1. O que é ITS?**

Segundo o Departamento de Transporte do Estado de Nova Iorque [DoT NY 2011], ITS é um programa nacional que foca na utilização de poder computacional e de comunicação para prover uma viagem mais inteligente, segura, rápida e conveniente.

Para a Universidade de Ciência e Tecnologia de Missouri [Missouri S&T 2011], vários sistemas podem ser considerados como ITS. Por exemplo:

- Sistemas de gerenciamento de tráfego que monitora o fluxo de veículos e provê decisões de suporte à diminuição de congestionamentos;
- Sistemas de informação a viajantes (SIV) que dá direções e rotas, além de informações em tempo real sobre as condições das rotas;
- Sistemas de automação de vias que dá suporte ou até substitui algumas funções de realização humana no processo de dirigir;
- Iniciativas de veículos inteligentes que foca no desenvolvimento de veículos cada vez mais automatizados.

Para este projeto iremos implementar um APTS, do inglês *Advanced Public Transportation Systems*, que significa Sistemas Avançados de Transportes Públicos. O SIV está entre os que são classificados como APTS, além dele, o APTS provê pagamentos eletrônicos e tecnologias de rastreamento de veículos, podendo informar sobre acidentes e mudanças nos instantes de chegadas às paradas [Kern Concil 2011]. Outra funcionalidade é quanto aos sistemas de controles dos veículos como, por exemplo, a visão sob neblina ou escuridão que detecta obstáculos como animais ou pessoas [U.S. DOT 2001].



Em Dublin, segundo [Caulfield *et al.* 2007], há vários serviços de informação ao viajante, por exemplo, o serviço “BUSTXT” que provê em tempo real ao usuário informações através de SMS (mensagens enviadas por telefones celulares) sobre hora de chegada de trens nas estações requisitadas. Há também telas eletrônicas nas paradas de trens e ônibus que mostram em tempo real o tempo que falta para os veículos chegarem nelas.

Em Gothenburg, na Suécia, há uma página na internet desde 1995 que dá a localização dos veículos de transporte público. O número de visitas cresceu de 100.000 para 1,2 milhões em 2001 [Gothenburg 2002]. Em Londres, em Seattle, e em várias outras cidades de países desenvolvidos, há sistemas deste tipo em funcionamento há mais de 10 anos [Caulfield *et al.* 2007].

### **2.3.2. Quais são os ganhos da utilização de ITS?**

Com a utilização de sistemas de transporte inteligentes, haverá uma grande redução dos congestionamentos, pois mais pessoas passarão a usar transporte público e, com isso, também diminuirá o número de acidentes nas ruas [Zhang *et al.* 2011]. Apenas com a diminuição de congestionamentos, haverá um grande aumento na qualidade de vida das pessoas, pois estarão menos estressadas, haverá menos consumo de combustível, o que diminui a poluição do ar [Shawe-Taylor *et al.* 2006] e até serão diminuídos os riscos de ataques do coração [Peters *et al.* 2004].

Se um sistema de transporte está funcionando com menos do que 100% da sua capacidade, significa que sua utilização não está ótima, logo, pode ser melhorada. Para isso, a utilização de ITS se torna absolutamente cabível e necessária [Missouri S&T 2011].

Um outro ganho com relação à utilização de sistemas de transporte inteligentes por usuários de transporte público, está na relação da escolha da melhor hora para realizar um traslado. Com um sistema sensível a mudanças de velocidade das vias causadas por congestionamentos, pode-se calcular e mostrar para o

usuário o tempo que seu trajeto duraria. Tendo estas informações em mãos, a pessoa pode simplesmente ficar um pouco mais no trabalho e chegar em casa meia hora mais rápido do que saindo no horário de pico, por exemplo.

### **2.3.3. Sugestão de Rotas**

Este trabalho será focado em sugerir rotas de transporte público levando em consideração informações contextuais, como já foi visto. A sugestão de rotas significa, basicamente, mostrar ao usuário opções de caminhos diferentes explicitando a distância, o tempo gasto e o preço total.

Em ITS, a sugestão de rotas é feita pelos ATIS (Advanced Traveler Information Systems) que verifica em tempo real as condições das rotas entre os locais apontados pelo usuário como inicial e final e mostra um caminho adequado [Missouri S&T 2011]. Para nosso trabalho, não iremos mostrar o melhor caminho, mas um caminho bom para cada opção de busca, sendo ela por tempo, distância ou preço.

### **3. Especificação da Ferramenta**

Neste capítulo será apresentada a ferramenta desenvolvida. Mostraremos o que compreendemos como sendo a solução para os problemas apresentados, como seria um cenário de utilização e como o sistema funciona, sem muitos detalhes técnicos. Por fim um esboço da arquitetura do sistema por completo.

#### **3.1. O que é?**

A ferramenta criada trata-se de um protótipo de um sistema indicador de rotas de transportes públicos. O usuário pode acessar por meio de um celular ou qualquer dispositivo conectado à internet. O protótipo foi desenvolvido com base no *GoogleMaps*, com isso, o usuário visualizará a rota indicada com maior facilidade [GoogleMaps 2011].

O público alvo deste sistema são pessoas que estão familiarizadas com *smartphones*, que estão conectadas à internet e que utilizam transportes públicos.

#### **3.2. O que faz?**

Ao acessarmos o *site* do aplicativo, vemos um mapa com vários pontos de paradas de ônibus em várias ruas. Para utilizar o aplicativo, o usuário escolhe no mapa a parada na qual ele se encontra, ou a mais próxima de si, e também escolhe a parada à qual ele deseja ir. Feito isto, escolhe uma das opções de calcular rota, sendo por menor custo, menor tempo ou menor distância e realiza a busca.

O sistema calcula as possíveis rotas e indica no mapa o caminho a ser tomado. Além disso, indica também um conjunto de instruções relacionadas à tomada de ônibus, de acordo com número da linha e sentido, e relacionadas às trocas de linhas.

Para o cálculo da rota ser feito, são levados em consideração diversos fatores, entre eles, estão:

- Linhas de ônibus que passam pelas paradas de origem e destino;
- Horários de saída das linhas em seu terminal;
- Preço individual de cada linha;
- Velocidade média das vias que fazem parte dos itinerários das linhas;
- Informações sobre condições do trânsito retiradas de redes sociais.

Com a exceção do último ponto citado acima, todos os outros são classificados como informações estáticas, pois estas informações serão guardadas no sistema de forma que não mudem de uma busca para outra. Já as informações sobre as condições de trânsito estarão constantemente mudando, sendo chamadas de informações dinâmicas. Tais mudanças se devem aos parâmetros da busca como parada de origem, de destino e tipo da busca.

Para este projeto, o foco foi maior para as buscas por menor tempo, já que são estas as que são diretamente afetadas pelas informações dinâmicas. Para duas buscas por menor tempo feitas com as mesmas paradas de origem e destino os resultados podem divergir. A diferença pode ser acarretada por fatores contextuais, como horário ou acidentes. O sistema obtém tais respostas a partir das informações capturadas das redes sociais.

### **3.3. Arquitetura do Sistema**

A Figura 3.1 representa a arquitetura em alto nível do protótipo. Em azul, estão os componentes desenvolvidos que servirão para os próximos passos do projeto. Em vermelho, estão as bases de dados utilizadas, nesta fase do projeto, foram simuladas as paradas, linhas, preços e também as informações contextuais. Em verde, está o subsistema de captura de informações contextuais do Twitter, que poderá servir para os próximos passos também.

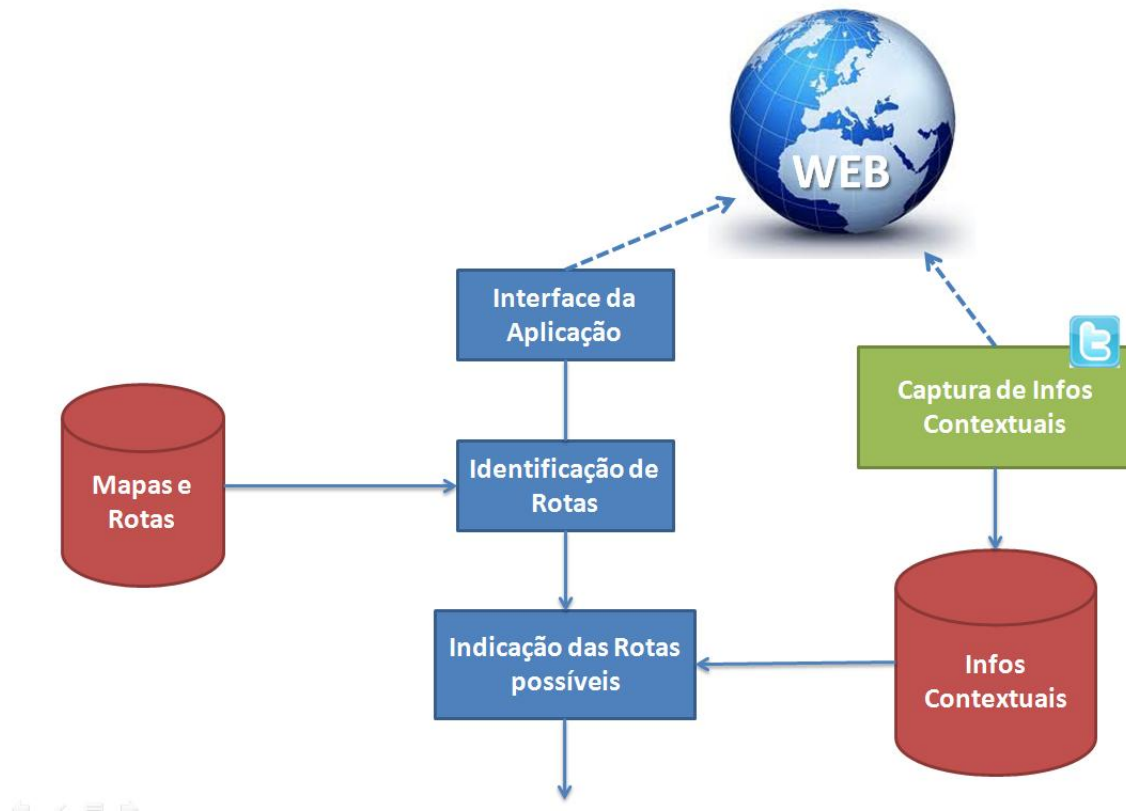


Figura 3.1 - Arquitetura de alto nível do sistema

Vemos que há uma seta apontando para baixo, ela representa a necessidade de um subsistema de indicação de melhor rota possível. Logo, nosso resultado pode não ser o melhor. Um fator que acreditamos ser de diferencial neste projeto é o de disponibilizarmos o serviço na internet, não será uma aplicação que rodará em um dispositivo específico, mas sim uma aplicação online acessível por qualquer navegador.

### 3.4. Casos de Uso do Sistema

A seguir serão descritos os casos de uso do sistema, após a descrição deles, partiremos para a implementação do sistema propriamente dita.

#### UC01 – Visualizar informações de uma Parada

Pré-condição: o usuário está na tela principal, onde há o mapa com as paradas.

Pós-condição: o sistema mostrará uma tela com as informações da parada selecionada.

Fluxo principal:

- 1- O usuário clica em uma das paradas mostradas no mapa;
- 2- O sistema faz uma busca interna para saber as informações da parada, como número, linhas que passam por ela e se ela já foi escolhida como origem ou destino;
- 3- O sistema mostra uma tela com as informações de número da parada e linhas que passam com ela, além de um conjunto de opções;

Fluxo secundário: no item 3-, as opções dependem se a parada já foi escolhida como origem ou destino, ou não. Caso tenha sido, aparecerá a opção de desmarcar, caso contrário, haverá as opções de marcar como origem e marcar como destino.

## **UC02 – Selecionar parada de origem/destino**

Pré-condição: o usuário está na tela de visualização de uma parada.

Pós-condição: o sistema mostrará o mesmo mapa inicial das paradas, com a parada escolhida em destaque.

Fluxo principal:

- 1- O usuário escolhe a opção de marcar a parada como origem ou destino;
- 2- O sistema retorna à tela inicial com a parada em destaque.

Fluxo secundário: no item 1-, o sistema irá marcar esta parada como origem ou destino desta viagem na seção do usuário.

## **UC03 – Selecionar tipo de busca**

Pré-condição: o usuário já escolheu as paradas de origem e destino da sua viagem.

Fluxo principal:

- 1- O usuário escolhe entre “custo”, “tempo” e “distância”;
- 2- O sistema retorna à tela inicial.

Fluxo secundário: no item 1, o sistema irá guardar a escolha da busca na seção do usuário

#### **UC04 – Realizar busca**

Pré-condição: o usuário já escolheu as paradas de origem e destino da sua viagem e o tipo da busca.

Pós-condição: o sistema retorna para o usuário um mapa contendo um caminho destacado que representa o caminho dos ônibus que o usuário deverá tomar. Além disso, o sistema retorna um conjunto de instruções para que o usuário realize durante sua viagem. A rota retornada pode levar em consideração uma consulta a um sistema externo de informações sobre o trânsito.

Fluxo principal:

- 1- O usuário clica no botão “buscar”;
- 2- O sistema retorna à tela inicial com as paradas selecionadas e a rota retornada em destaque.

Fluxo secundário: no item 2-, o sistema irá se comunicar com um sistema externo para verificar as condições do trânsito nas ruas que são utilizadas pelas possíveis rotas. Este sistema extrai informações de redes sociais.

## 4. Implementação da Ferramenta

Nesta sessão, apresentaremos as modelagens utilizadas no projeto, os modelos de banco de dados das Paradas, Linhas e Informações Contextuais. Além disso, será mostrado o projeto do subsistema de comunicação com o Twitter para a captura das informações contextuais. Também abordaremos como foi feita a busca das informações contextuais e por fim, será mostrado um cenário de utilização com as telas do sistema funcionando localmente.

O aplicativo foi programado na linguagem Python [Python 2011], com o *framework open source* Django [Django 2011] e as bases de dados em SQLite3 [SQLite 2011]. A comunicação com o *Twitter* foi feita a partir da API REST [API REST 2011] disponibilizada pelo próprio *Twitter* e o *crawler* (capturador de informações do *Twitter*) também foi implementado em Python.

### 4.1. Modelagens Utilizadas

Neste tópico, apresentaremos os modelos lógico e conceitual do modelo entidade-relacionamento utilizado para o banco de dados da base de paradas, linhas e informações contextuais.

#### 4.1.1. Modelo conceitual

Vemos neste modelo conceitual que temos poucas entidades básicas, porém todas estão se relacionando. As arestas possuem trechos, que são pontos de trajeto, as linhas passam pelas arestas e param nas paradas. As informações contextuais irão afetar alguns trechos, e não uma aresta toda, porém uma informação de trânsito só afetará apenas um trecho, mesmo que este trecho já esteja sendo afetado por outra informação.



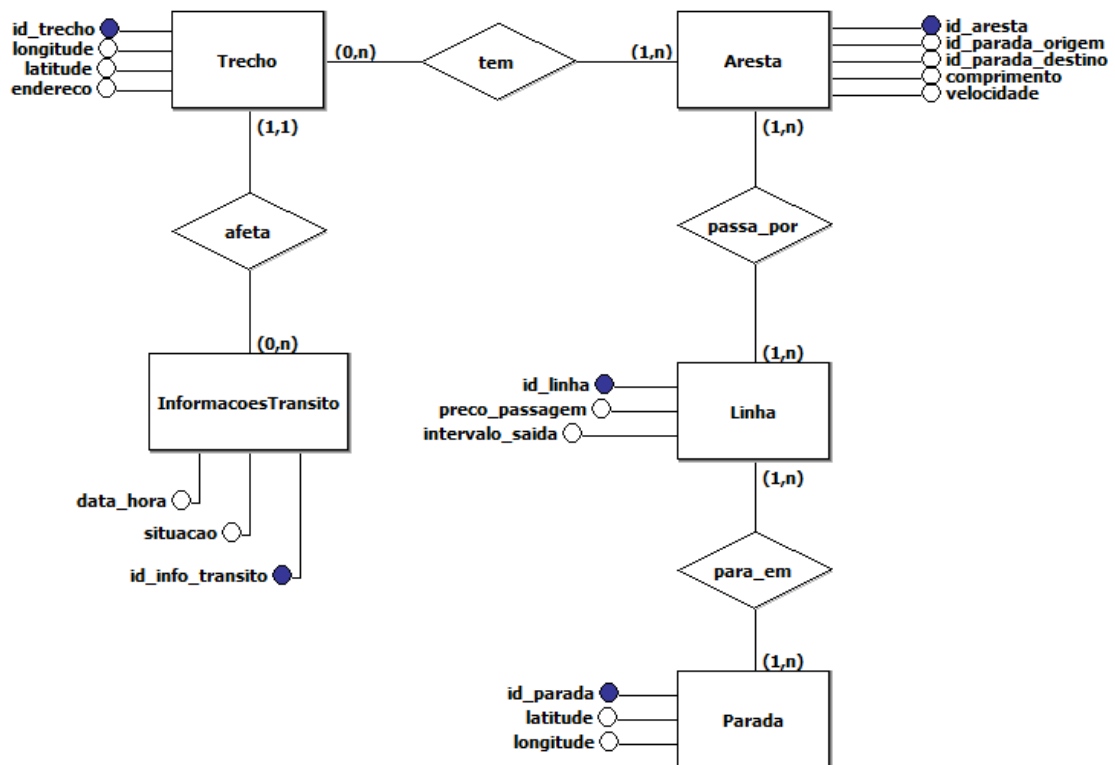


Figura 4.1 - Modelo Conceitual das Paradas, Linhas e Informações Contextuais

#### 4.1.2. Modelo Lógico

No modelo lógico podemos ver que com a exceção do relacionamento das informações de trânsito com os trechos, todos os relacionamentos se tornaram tabelas. Podemos perceber também que a aresta possui chave estrangeira para a parada, pois não seria preciso criar uma tabela apenas para guardar as paradas de origem e destino de cada aresta.

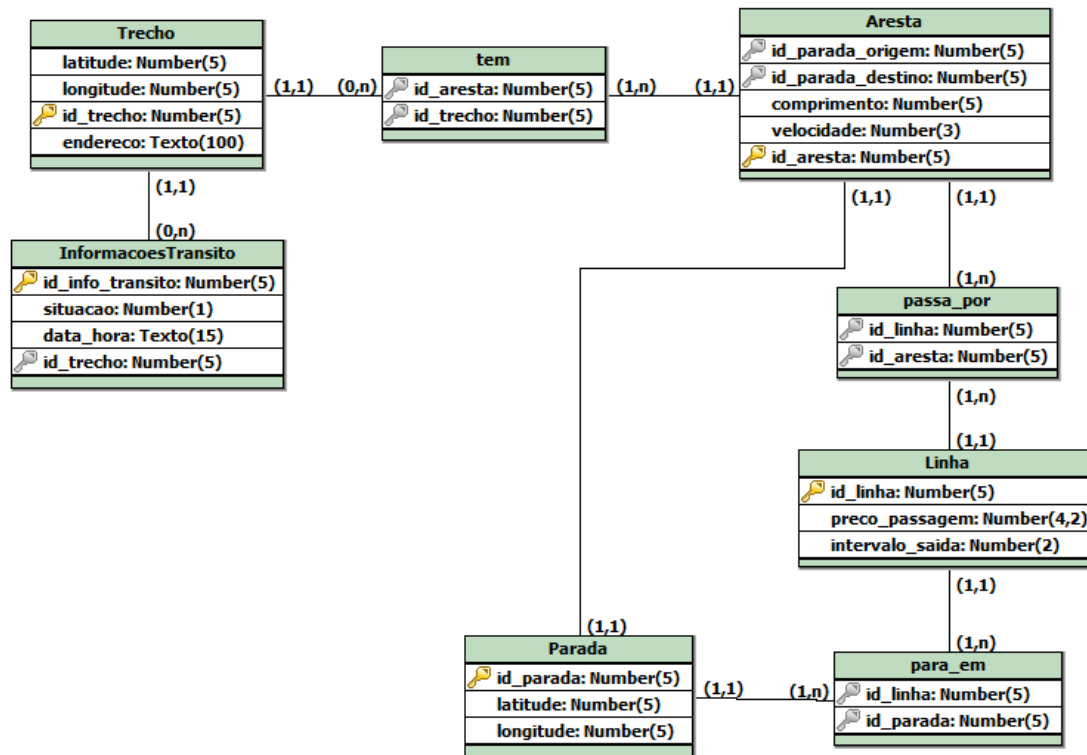


Figura 4.2 - Modelo Lógico das Paradas, Linhas e Informações Contextuais

## 4.2. Projeto do subsistema de comunicação com Twitter

Para a comunicação com o Twitter e a extração de informações, construímos um *crawler* que a cada intervalo de tempo pré-determinado verifica se foi falada alguma coisa sobre trânsito. Esta ferramenta foi construída para buscar em várias páginas da Web, porém aplicamos nela uma restrição para procurar por informações apenas no site do Twitter, e mais especificamente, do perfil @UbiBus\_PE.

## 4.2.1. Arquitetura do Subsistema

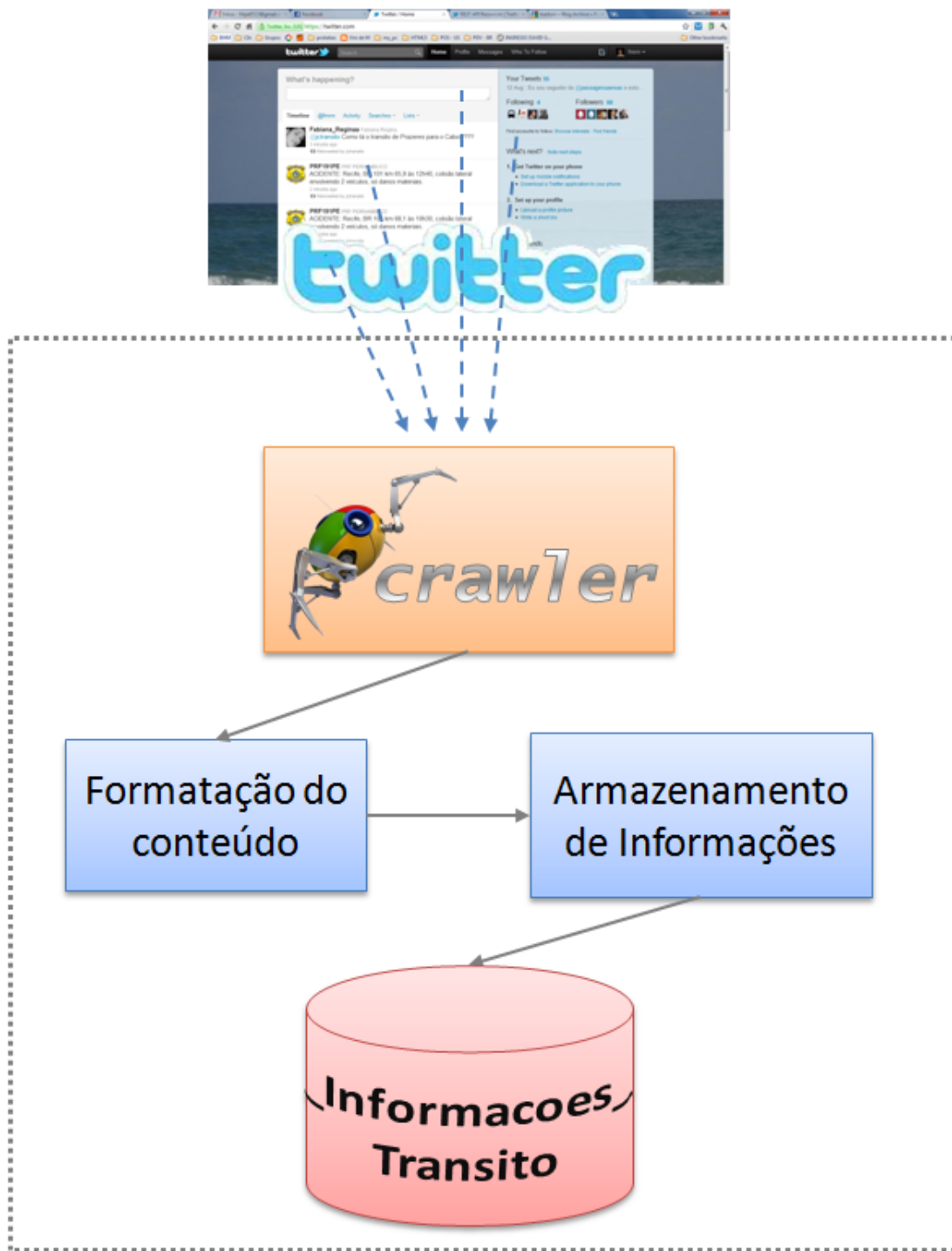


Figura 4.3 - Arquitetura do Subsistema de Comunicação com o Twitter

## 4.2.2. Como funciona?

O sistema foi construído para rodar independentemente de o servidor estar rodando ou não. O *crawler* irá capturar as informações do @UbiBus\_PE de acordo com dois parâmetros que poderão ser redefinidos no futuro.

- Quantidade máxima de *tweets* por captura: 20 *tweets*;
- Diferença entre os instantes de tempo atual e de criação do *tweet*: 30 minutos.

Com isso, temos os 20 últimos *tweets* que foram publicados pelo @UbiBus\_PE. Agora, eles serão formatados para, em seguida, a informação ser extraída e armazenada na tabela *InformacoesTransito*, vista no modelo do banco de dados.

A formatação é basicamente um conjunto de funções que irão processar a tupla que foi passada pela API do *Twitter*, que é o próprio *tweet*, e transformar em outra tupla que será mais facilmente entendida pela outra parte do sistema. A tupla de entrada possui o formato (*conteúdo\_do\_tweet*, *latitude\_longitude*, *data\_do\_tweet*), com ela, iremos extrair do conteúdo o trecho e a situação do trânsito que está sendo falado.

Para a extração do trecho e da situação do trânsito, utilizamos expressões regulares e a função nativa de Python de *stringmatching*. Esta função é redefinida por nós, criando uma gramática para ela:

<HORA> h <MIN> min Transito <AGRAVANTE> <SITUACAO>  
<PREPOSICAO> <LOCAL> <TRECHO>

<HORA> → número de 00 a 23;

<MIN> → número de 00 a 59;

<AGRAVANTE> →  **muito | pouco | quase | bastante | meio**

<SITUACAO> →  **parado | lento | ruim | bom | livre | triste | fácil |  
difícil | moderado | péssimo | complicado | devagar | rápido |  
congestionado | engarrafado | fluindo | normal | tranquilo | leve**

<PREPOSICAO> →  **em | na | no**

<LOCAL> → nomes de ruas da base de dados

<TRECHO> → trechos cadastrados na base de dados

Após a extração, teremos uma lista de tuplas da forma (*trecho*, *situação\_transito*, *data\_hora*). A situação do trânsito é calculada a partir de uma função que foi criada para classificar o adjetivo <SITUACAO> combinado com o seu <AGRAVANTE>. Esta função retorna um número de 0 a 3 onde:

- 0 → trânsito livre
- 1 → trânsito moderado
- 2 → trânsito lento
- 3 → trânsito muito lento

Mais tarde, este número será utilizado pelo algoritmo de busca de rotas para influenciar na velocidade média (informação estática da aresta) e, conseqüentemente, no tempo total da viagem.

Para armazenarmos na base de dados, primeiro fazemos a verificação se esta informação já se encontra nela. Se não houver nada a respeito deste trecho ou se a situação estiver diferente, a informação é armazenada (ou atualizada). O que poderia ser feito no futuro, mas ainda não está sendo feito, é uma classificação da informação em termos de sua confiabilidade. Para cada informação que vem repetida, significa que a probabilidade de ser verdade é maior e, com isso, evitamos capturar informações postadas por má fé de usuários. Logicamente, falamos deste próximo passo considerando que as informações serão pegadas não só apenas do perfil do @UbiBus\_PE, mas sim de vários outros a mais.

#### **4.2.3. Detalhes da Comunicação – API REST**

REST significa Transferência de Estado Representacional, do inglês *Representational state transfer*. REST é uma técnica de engenharia de software para fornecer uma camada de comunicação em cima do protocolo de internet HTTP. REST não é um protocolo, não possui padrão, mas qualquer pessoa pode fornecer a API REST do seu site.

Isso só é possível, porque REST trabalha apenas com requisições de *url* no *browser*, logo, qualquer dispositivo que acesse a internet, poderá enviar requisições e receber respostas de um site utilizando a API REST dele. Com esta ferramenta, é possível fazer a abstração do dispositivo que está fazendo a requisição, podendo ele ser um *desktop*, um *tablet* ou um celular [REST 2011].

Com a utilização desta API fizemos a comunicação com o *Twitter* para pegar as informações postadas pelo perfil @UbiBus\_PE, mas além disso, utilizamos os mesmos conceitos para as requisições de rotas do nosso sistema. O usuário, ao passar suas paradas de origem e destino e o tipo de busca, está por trás da aplicação enviando uma requisição pela *url* ao servidor. O servidor faz os cálculos e retorna um dicionário para o usuário. Este dicionário é o estado do sistema para aqueles parâmetros, ele é interpretado pela aplicação do usuário e então o usuário vê o mapa com a rota retornada.

Em um dicionário retornado por uma requisição de um usuário, verificamos que existe uma origem, um destino, os passos e vários, ou nenhum, pontos de trajeto. A origem é um outro dicionário contendo Latitude, Longitude e Instrução. O destino é igual à origem, com a adição da mensagem final. Os passos são um dicionário com Latitude, Longitude e várias mensagens de texto indicando trocas de ônibus. Os pontos de trajeto são dicionários com Latitude e Longitude para que a rota seja desenhada corretamente no mapa.

### **4.3. Como foi feito?**

Nesta sessão, iremos descrever como foi feita a busca pelas rotas dando ênfase ao algoritmo utilizado citando seus prós e contras.

A busca pelas rotas foi feita com um único algoritmo, que se mostrou bastante eficiente para as buscas por distância e por tempo. Porém verificamos que na busca pelo preço, ele não é o mais indicado, pois ocorreram problemas quando havia rotas com trocas de ônibus.

Utilizamos o algoritmo de busca gulosa de Dijkstra, pois ele trata bem um grafo com pesos nas arestas para achar o caminho de menor custo [Dijkstra 2011]. Numa próxima versão deste projeto iremos utilizar o algoritmo A\*, que se encaixa melhor com a solução do nosso problema. O A\* é uma melhoria do Dijkstra, pois utiliza uma função heurística e encontra o caminho mais rapidamente e de uma forma global [A\* 2011].

Nosso grafo foi construído tendo seus nós como tuplas do tipo (*Parada, Linha*). Com isso, foi possível diferenciar o caminho, de uma parada para outra, realizado por mais de uma linha diferente. Os pesos das arestas do grafo são calculados de acordo com o tipo de busca. Para a busca por distância, temos armazenadas nas próprias arestas as distâncias de uma parada à outra. No caso da busca por tempo, temos armazenadas nas arestas as velocidades médias das ruas, logo, fazemos o cálculo do tempo de acordo com razão da distância pela velocidade. Para a busca por preço, temos armazenados nas arestas os preços das linhas de ônibus, porém encontramos problemas com as trocas de linhas.

Na busca por menor tempo, em cada repetição que o algoritmo dá, é chamada uma função que verifica se há informação de trânsito sobre o trecho que está sendo percorrido. No caso de haver tal informação, será feito o cálculo do tempo levando em consideração a velocidade média da aresta e o fator de afetação, número de 0 a 3. Para os valores, temos os seguintes resultados:

- 0 → trânsito livre, acrescenta-se 05 Km/h à velocidade média;
- 1 → trânsito moderado, diminui-se 20 Km/h da velocidade média;
- 2 → trânsito lento, diminui-se 50% da velocidade média;
- 3 → trânsito muito lento, diminui-se 50 Km/h da velocidade média.

Ao final desta mudança na velocidade, fazemos uma verificação para não deixar a velocidade média da via se tornar menor do que 05 Km/h. Porém estes valores foram postos sem um estudo efetivo, só estão aí para que verifiquemos ao final da busca uma alteração no tempo padrão de viagem.





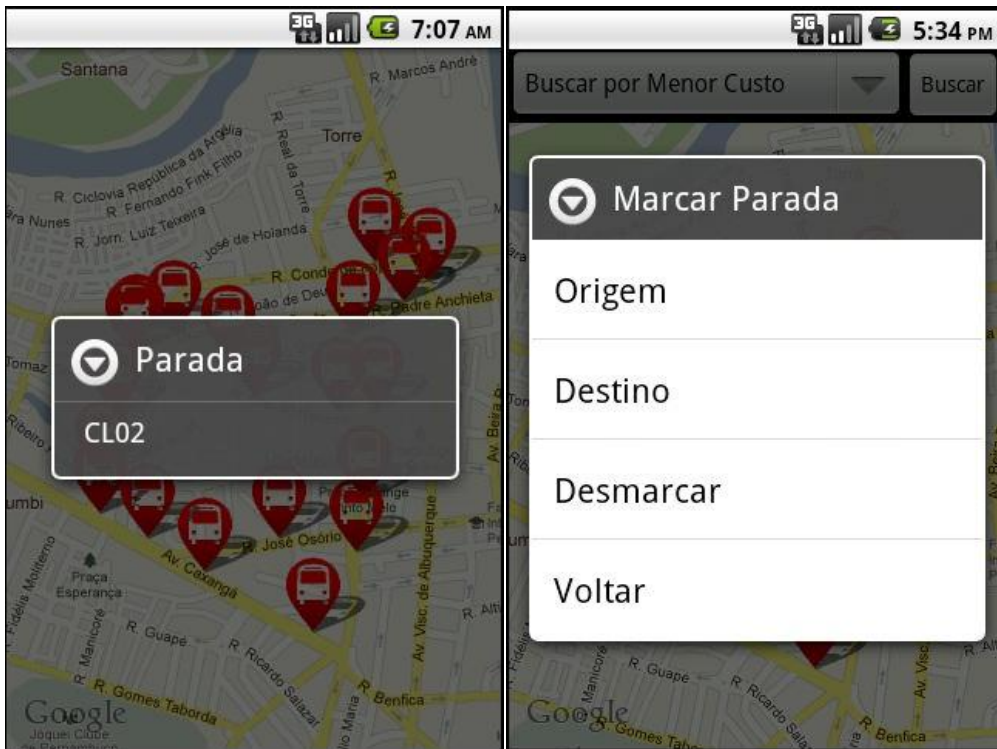


Figura 4.5 - Parada selecionada e opções de marcar como origem ou destino.

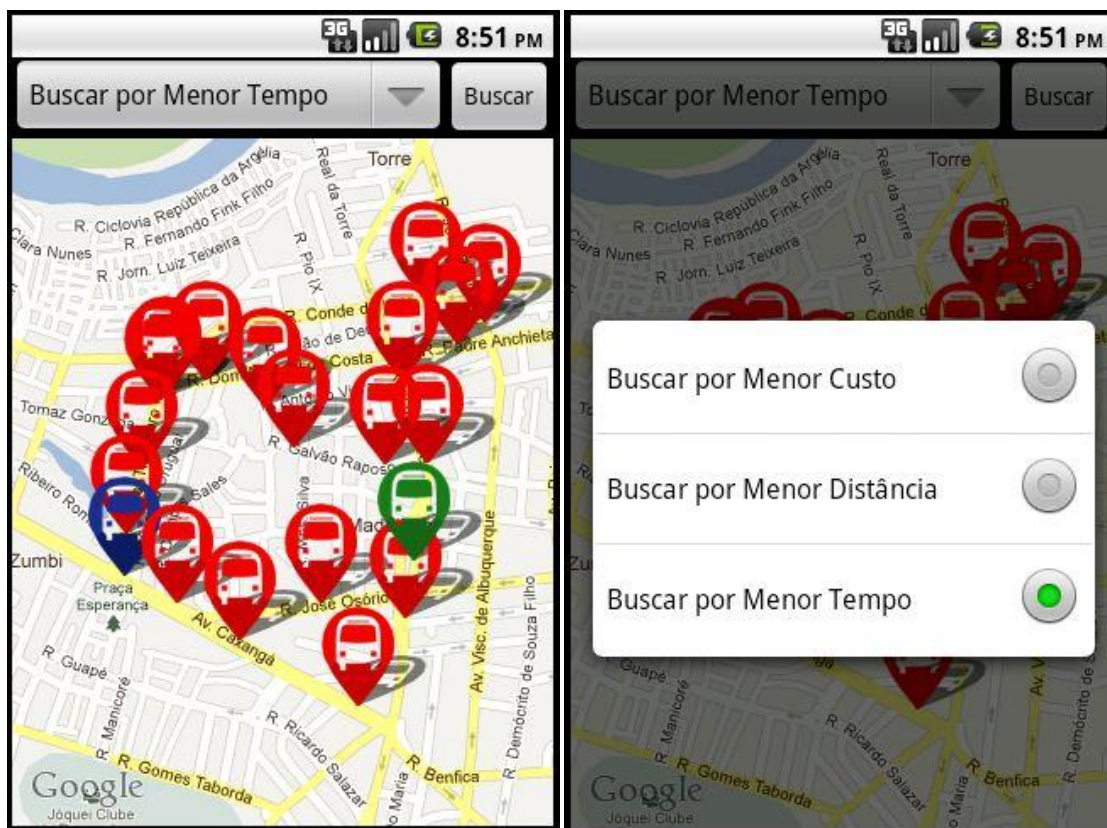


Figura 4.6 - Mapa com as paradas escolhidas em destaque.

Figura 4.7 - Possíveis critérios de escolha de rotas.



Figura 4.8 - Mapa com a rota indicada.

Figura 4.9 - Instruções a serem seguidas.

Vemos que o tempo total gasto na viagem foi de 11 minutos, pois estamos na situação em que não há trânsito em nenhuma via. Para uma melhor compreensão da influência da informação contextual, iremos publicar *tweets* de trânsito lento em uma das vias que a rota passa, a Rua José Osório (Figura 4.10).

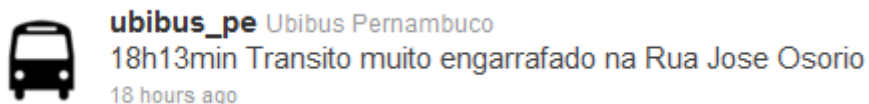


Figura 4.10 – Tweet de Informação de Trânsito

Ao refazer a busca da rota acima, verificamos que o tempo total gasto na viagem é de 16 minutos (Figura 4.11). Caso houvesse mais postagens falando sobre engarrafamento nas ruas que estão nesta rota, o sistema iria sugerir uma rota alternativa (Figura 4.12).



Figura 4.11 – Mudança no tempo total gasto na viagem



Figura 4.12 – Mudança na rota sugerida

## 5. Conclusão

O objetivo deste trabalho foi atingido com sucesso, foi criada uma ferramenta que sugere rotas de transportes públicos de acordo com o local que o usuário está e para onde ele deseja ir. Para isto, foram aprendidos vários conceitos novos, desde os na área de ITS até os da área de desenvolvimento de serviços Web. Também foram estudadas e aprendidas várias tecnologias diferentes, tanto para a criação da ferramenta de sugestão de rotas, quanto para o módulo de captura de mensagens do *Twitter*.

Porém foram verificados alguns pontos de melhoria que não foram corrigidos para este projeto, e que ficam como sugestões para trabalhos futuros nesta área. Tais melhorias foram apontadas ao longo deste trabalho, porém, aqui, irei listá-las complementando-as com uma breve justificativa.

- **Implementar outro algoritmo para a busca por menor preço;**  
O algoritmo atual está apresentando problemas quando há trocas de ônibus, e já foi comprovado que ele não resolve este tipo de problema.
- **Transferir a base de paradas e linhas para uma base real;**  
O universo que estamos trabalhando está restrito, talvez não seja o ideal para validar testes de desempenho dos algoritmos de busca.
- **Expandir a captura de informações de trânsito para outros perfis e outros sites;**  
Só o *Twitter* pode não ser o bastante, já há comunidades no *Facebook* [Facebook 2011] que tratam deste problema de trânsito e, em particular, o *GoogleMaps* está com um serviço de mostrar as condições do trânsito em algumas vias, podendo ser tomado como entrada.
- **Evoluir o algoritmo de interpretação de informações de trânsito para um que analise a semântica;**  
A utilização de expressão regular para capturar a situação do trânsito é muito restrita, pois a informação deve vir em um padrão pré-definido. Ao

utilizar algoritmos de análise semântica, a informação poderá vir sem tanta organização, de forma não padronizada.

Além de todas estas mudanças propostas, várias mudanças ocorreram durante a implementação do projeto. Verificamos falhas em alguns aspectos e também outras funcionalidades que deveriam existir, e mudamos várias coisas. Tais funcionalidades são, por exemplo, a utilização como serviço, para que qualquer dispositivo possa acessar; a utilização de coordenadas geográficas, para uma melhor comunicação com o *GoogleMaps*; entre outras.

## Referências

- [A\* 2011] Algoritmo A\* <http://www.policyalmanac.org/games/aStarTutorial.htm>  
(acessado em dezembro de 2011)
- [API REST 2011] API REST do Twitter <https://dev.twitter.com/docs/api>  
(acessado em dezembro de 2011)
- [Caulfield *et al.* 2007] B. Caulfield, M. O'Mahony, "An Examination of the Public Transport Information Requirements of Users", In: IEEE Transactions on Intelligent Transportation Systems, Vol. 8, No. 1, March 2007.
- [Dijkstra 2011] Algoritmo de Dijkstra  
<http://www.cs.auckland.ac.nz/~jmor159/PLDS210/dijkstra.html> (acessado em dezembro de 2011)
- [Django 2011] Django <https://www.djangoproject.com/> (acessado em novembro de 2011)
- [DoT NY 2011] Departamento de Transporte do Estado de Nova Iorque  
<https://www.dot.ny.gov/divisions/operating/oom/transportation-systems/systems-optimization-section/ny-moves/what-is-its> (acessado em novembro de 2011)
- [Facebook 2011] Rede social Facebook <https://www.facebook.com/facebook>  
(acessado em dezembro de 2011)
- [GoogleMaps 2011] GoogleMaps <http://code.google.com/apis/maps/index.html>  
(acessado em dezembro de 2011)
- [Gothenburg 2002] Gothenburg Traffic Information Centre, "Effects of real-time information in Gothenburg", In: 19th GoTiC Rep., GOTIC, Gothenburg, Sweden, 2002.

- [Kern Concil 2011] Kern Concil of Governments  
<http://www.kerncog.org/cms/transportation/its> (acessado em dezembro de 2011)
- [Magalhães 2008] Magalhães, L. H. (2008) “Uma análise de ferramentas para mineração de conteúdo de páginas Web”, COPPE/UFRJ, M.Sc., Engenharia Civil.
- [Missouri S&T 2011] Universidade de Ciência e Tecnologia de Missouri  
<http://web.mst.edu/~ff/IV/AHSpaper221.html> (acessado em novembro de 2011)
- [Pak *et al.* 2010] Pak, A., Paroubek, P. (2010) “Twitter as a Corpus for Sentiment Analysis and Opinion Mining”, Université de Paris-Sud, Laboratoire LIMSI-CNRS.
- [Peters *et al.* 2004] A. Peters, S. von Klot, M. Heier, I. Trentinaglia, A. Hörmann, H. E. Wichmann, and H. Löwel, “Exposure to traffic and the onset of myocardial infarction”, In: New England J. Med., vol. 351, no. 17, pp. 1721–1730, Oct. 2004.
- [REST 2011] REST <http://en.wikipedia.org/wiki/REST> (acessado em dezembro de 2011)
- [Shawe-Taylor *et al.* 2006] J. Shawe-Taylor, T. de Bie, and N. Cristianini, “Data mining, data fusion and information management”, In: Proc. Inst. Elect. Eng.—Intell. Transp. Syst., vol. 153, no. 3, pp. 221–229, Sep. 2006.
- [Sipser 2007] M. Sipser “Introdução à Teoria da Computação”, Tradução da segunda edição norte-americana, pp. 31-140, 2007.
- [SQLite 2011] SQLite <http://www.sqlite.org/> (acessado em novembro de 2011)
- [Twitter 2011] Twitter <http://www.twitter.com/> (acessado em novembro de 2011)

[U.S. DOT 2001] U.S. DOT Volpe National Transportation Systems Center  
[http://wtiwebdev.coe.montana.edu:50001/Documents/RepositoryDocuments/State\\_of\\_ITS\\_in\\_NP.pdf](http://wtiwebdev.coe.montana.edu:50001/Documents/RepositoryDocuments/State_of_ITS_in_NP.pdf) (acessado em dezembro de 2011)

[Vieira *et al.* 2007] Vieira, V., Tedesco, P., Salgado, A. C., Brézillon, P. (2007) "Investigating the Specificities of Contextual Elements Management: The CEManTIKA Approach", In: Proc. of the 6<sup>th</sup> International and Interdisciplinary Conference on Modeling and Using Context (CONTEXT'07), LNAI 4635, pp. 493-506, Roskilde, Denmark.

[Vieira *et al.* 2009] Vieira, V., Tedesco, P., Salgado, A. C. (2009) "Modelos e Processos para o Desenvolvimento de Sistemas Sensíveis ao Contexto", In: Atualizações em Informática, Editora PUC-Rio, p. 381-431.

[Zhang *et al.* 2011] J. Zhang, F. Wang, K. Wang, W. Lin, X. Xu, C. Chen, "Data-Driven Intelligent Transportation Systems: A Survey", In: IEEE Transactions on Intelligent Transportation Systems 1, March 2011.