

# MAC - 5853

## Um serviço simples de controle de rotas

Paulo J. S. Silva, Flávio Soares Silva e Alfredo Goldman

Primeiro Semestre de 2009

### O Enunciado

Para os moradores da cidade de São Paulo o trânsito é um pesadelo. Horas e horas infindáveis são gastas. Mas a sina de alguns pode se transformar em oportunidade para outros. Esse é o caso da empresa *Rotas*. Rotas é uma empresa que deseja vender o serviço de sugerir rotas alternativas para seus usuários fugirem do trânsito e chegarem mais rápido ao destino.

A idéia é bem simples. Cada usuário, usando o GPS que diz exatamente onde ele está, consulta a empresa pedindo uma rota para ir até seu destino. A empresa então, usando informação do tráfego na cidade disponibilizada pela CET, calcula a rota de tempo mínimo e a passa ao cliente. O cliente então tem uma rota adaptada que pode lhe economizar um bom tempo por dia. Além disso, como mesmo assim o tráfego é muito dinâmico na cidade, a cada novo entrocamento o cliente volta a consultar a empresa para ver se a melhor rota mudou e se adapta então à resposta.

Nesse projeto iremos implementar o protótipo do sistema da empresa Rotas. Com esse protótipo a empresa pretende avaliar a eficácia de sua idéia e também fazer algumas demonstrações para possíveis clientes.

O sistema é composto de três módulos que devem ser implementados como programas independentes:

**CET** : Esse programa vai simular o papel da CET. Ele deve manter uma imagem do estado do sistema viário da cidade, que será representado por um grafo mais um valor entre  $[0, 1]$  em cada aresta que representa a taxa de ocupação daquela via. Como o trânsito varia muito durante o dia, a CET possui vários perfis de trânsito (que são valores diferentes das arestas), um para cada intervalo e meia hora. Tipicamente o perfil associado a períodos do horário de pico possuem taxas de ocupações maiores do que perfis associados a horários de pouco movimento.

**Cliente** : Esse programa representa um cliente. Como pode haver vários deles (essa é a esperança da empresa), pode haver várias instâncias desse programa. O módulo cliente pode mandar para o módulo Rotas um par de

vértices do grafo que representam a sua posição atual e o seu destino. Ele espera então receber do sistema Rotas uma aresta, que representa uma via na cidade, para iniciar o seu deslocamento. Note que o sistema Rotas devolve apenas a primeira via que o cliente, já que esse vai consultar novamente o sistema Rotas assim que chegar ao próximo entrocamento (vértice).

**Rotas** : Esse é o sistema principal. Ele deve ser capaz de calcular as rotas de tempo mínimo e também manter o histórico das rotas usadas por cada cliente para gerar uma conta no final do mês. Cada cliente é representado por seu nome, endereço e CPF. As funcionalidades que devem estar presentes no sistema Rotas.

1. Pedir informação atualizada da taxa de ocupação das vias para a CET a cada meia hora.
2. Receber requisições dos clientes e responder com a primeira aresta do caminho de tempo mínimo ligado a origem e o destino desejados. O sistema deve guardar a rota completa que o cliente usou para realizar o seu deslocamento. Essas rotas devem ser descartadas quando tiverem mais do que 30 dias. *Por simplicidade, considere que o cliente sempre tomará as rotas sugeridas pelo sistema Rotas.*
3. O sistema deve ser capaz de gerar uma conta por cliente cobrando pelo número de arestas sugeridas durante o dia (que é equivalente ao número de consultas).
4. O sistema deve ser capaz de recuperar as rotas completas utilizadas pelos clientes em seu deslocamento nos últimos 30 dias.

## 1 Representação do sistema viário

O sistema viário da cidade deve ser representado por um grafo em que os vértices representam entrocamento de vias (cruzamentos) e as arestas as vias. Note que todos os componentes dos sistemas devem conhecer esse grafo. Logo é razoável mantê-lo em disco e lê-lo ao inicializar cada componente.

O sistema considera que tempo que leva para percorrer uma aresta (ou via), deve ser  $1/(1 - \text{taxa de ocupação})$ , sendo que “taxa de ocupação” da cada via é dada pelo componente CET a cada meia-hora. O componente CET deve então manter uma tabela com 48 valores de taxa de ocupação das vias para cada intervalo de meia-hora. Para simplificar, vamos assumir que as taxas de ocupação se repetem todos os dias.

Para calcular o caminho entre dois vértices, a partir de uma requisição de um cliente, o sistema deve usar um algoritmo como o de Dijkstra, descrito por exemplo em [http://en.wikipedia.org/wiki/Dijkstra%27s\\_algorithm](http://en.wikipedia.org/wiki/Dijkstra%27s_algorithm). Lembre que o Cliente irá pedir atualizações do caminho a medida que se desloca. Porém o resultado só muda se a informação de tráfego mudou desde a última consulta (devido a novas informações terem chegado da CET para o sistema Rotas). Seu

sistema deve apenas calcular uma nova rota nesse caso. O sistema pode assumir que o cliente vai continuar sua viagem até o chegar ao seu destino. Ou seja, não há desistência.

Por fim, toda a comunicação entre os componentes deve ocorrer usando mensagens em XML. Ainda, os arquivos de inicialização devem também usar XML. Ficará a seu critério definir tanto o formato utilizado nos arquivos XML, na troca de mensagens, quanto o modelo de dados no banco de dados que armazena as informações das rotas usadas pelo cliente. Não há necessidade de uma GUI. Aceitamos que o sistema tenha apenas uma interface por linha de comando, desde que todos os requisitos descritos nesse documento estejam sendo observados.

A implementação deve ser realizada preferencialmente nas linguagens C++, Java, Smalltalk ou Python. Se você preferir desenvolver o sistema em outra linguagem, consulte os professores. O trabalho deve ser desenvolvido em duas fases.

### **Fase 1 — Projeto (entrega: 16/02/2009)**

Nesta primeira fase você deve apresentar o projeto de seu sistema. Este projeto deverá conter:

- (a) Para a descrição do controle de portaria acima crie o modelo ER-X e seu respectivo modelo físico em banco de dados relacional. Detalhando a ativação do sistema, os formatos de entrada e um esboço da aparência da interface (por linha de comando ou gráfica).
- (b) Um conjunto de exemplos e cenários de operação do sistema, mostrando, para cada um dos cenários, quais os comandos SQL serão enviados ao banco de dados e qual a resposta será gerada pelo sistema.
- (c) Uma descrição de quais tipos de entradas serão aceitáveis pelo seu sistema e quais entradas serão consideradas incorretas. Neste último caso, deve-se especificar qual será o comportamento do sistema.
- (d) Um (ou mais) diagrama(s) detalhando a arquitetura prevista para o sistema (UML é bem-vinda).

### **Fase 2 — Implementação (entrega: 16/03/2009)**

Na segunda fase você deverá implementar o sistema completo. Ele deverá implementar os três componentes e mostrar que estes se comunicam de forma correta. O sistema Rotas deve ser capaz de gerar a conta diária de um cliente e também de gerar um relatório com todas as rotas que um cliente usou nos últimos 30 dias. Lembre que o sistema deve permitir a criação de múltiplas instâncias de Cliente.

Você deve entregar todo o código implementado bem como documentação que descreve como compilá-lo e usá-lo. Por fim, o sistema deve ser apresentado funcionando para a banca em data a ser marcada depois da entrega.

Lembramos as datas importantes:

- 16/02/2007: entrega da fase 1;
- 16/03/2007: entrega da fase 2;
- Apresentação dos sistemas: data e horário a serem agendados para cada aluno.

As entregas deverão ser feitas em um único arquivo comprimido, no email [pjssilva@ime.usp.br](mailto:pjssilva@ime.usp.br). Se o arquivo exceder 2 Megabytes, o sistema de emails não realiza a entrega. Neste caso, coloque o arquivo em um site e envie o email dando o endereço do arquivo.

### **Bancos de Dados Livres**

Bancos de dados livres e gratuitos podem ser encontrados nos seguintes endereços:

**MySQL:** [www.mysql.com](http://www.mysql.com)

**PostgreSQL:** [www.postgresql.org](http://www.postgresql.org)

Informações sobre instalação do PostgreSQL podem ser encontradas a partir da seção de Links da página da disciplina MAC439 (Laboratório de Banco de Dados): <http://clinmaldb.usp.br/mac439/index.htm>.