ROTA | **R**eal-Time **O**ptimum **T**raffic **A**nalyzer

Plano de Gerenciamento de Projeto de Software – PGPS

**Versão 1.00**

01/04/2009

# Responsáveis

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Antônio Cláudio Goméz de Sousa

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Fernando Seabra Chirigati

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Rafael Shinji Aoki Kikuchi

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Talita Lopes Gomes

# Relatório de Mudanças

ROTA – **R**eal-Time **O**ptimum **T**raffic **A**nalyzer

**Versão 1.00 – 01/04/2009**

Criação do Documento

# Prefácio

Este documento tem por objetivo formalizar o planejamento do software ROTA – **R**eal-Time **O**ptimum **T**raffic **A**nalyzer –, e destina-se principalmente a toda a equipe de desenvolvimento do programa.

Neste plano, estão incluídos o sumário do projeto, a organização da equipe, os prazos de entrega e todas as outras informações relevantes que dizem respeito ao produto em questão.

# Índice

[1. Apresentação 1](#_Toc226113724)

[1.1 Sumário do Projeto 1](#_Toc226113725)

[1.1.1 Finalidades, Escopo e Objetivos 1](#_Toc226113726)

[1.1.2 Postulados e Restrições 1](#_Toc226113727)

[1.1.3 Liberações Parciais 1](#_Toc226113728)

[1.1.4 Sumário de Cronograma e Orçamento 1](#_Toc226113729)

[1.2 Evolução do Plano 2](#_Toc226113730)

[2. Referências 3](#_Toc226113731)

[3. Definições 4](#_Toc226113732)

[4. Organização do Projeto 5](#_Toc226113733)

[4.1 Interfaces Externas 5](#_Toc226113734)

[4.2 Estrutura Interna 5](#_Toc226113735)

[4.3 Papéis e Responsabilidades 5](#_Toc226113736)

[5. Processos de Gerenciamento 6](#_Toc226113737)

[5.1 Partida no Projeto 6](#_Toc226113738)

[5.1.1 Previsões 6](#_Toc226113739)

[5.1.2 Equipe 6](#_Toc226113740)

[5.1.3 Plano para a Aquisição de Recursos 6](#_Toc226113741)

[5.1.4 Plano de Treinamento da Equipe 6](#_Toc226113742)

[5.2 Plano de Trabalho 6](#_Toc226113743)

[5.2.1 Atividades 6](#_Toc226113744)

[5.2.2 Prazos 7](#_Toc226113745)

[5.2.3 Alocação de Recursos 7](#_Toc226113746)

[5.2.4 Alocação de Orçamento 7](#_Toc226113747)

[5.3 Planos de Controle 7](#_Toc226113748)

[5.3.1 Controle dos Requisitos 7](#_Toc226113749)

[5.3.2 Controle dos Prazos 7](#_Toc226113750)

[5.3.3 Controle do Orçamento 7](#_Toc226113751)

[5.3.4 Controle de Qualidade 7](#_Toc226113752)

[5.3.5 Plano de Relatórios 7](#_Toc226113753)

[5.3.6 Plano de Medidas 8](#_Toc226113754)

[5.4 Plano de Gerenciamento de Riscos 8](#_Toc226113755)

[5.5 Plano de Encerramento 8](#_Toc226113756)

[6. Processos Técnicos 9](#_Toc226113757)

[6.1 Modelo dos Processos 9](#_Toc226113758)

[6.2 Métodos, Ferramentas e Técnicas 9](#_Toc226113759)

[6.3 Infraestrutura 9](#_Toc226113760)

[6.4 Plano para a Aceitação do Produto 9](#_Toc226113761)

[7. Planos para os processos de Suporte 10](#_Toc226113762)

[7.1 Gerenciamento de Configuração 10](#_Toc226113763)

[7.2 Plano de Verificação e de Validação 10](#_Toc226113764)

[7.3 Documentação 10](#_Toc226113765)

[7.4 Plano para Assegurar a Qualidade 10](#_Toc226113766)

[7.5 Revisões e Auditorias 10](#_Toc226113767)

[7.6 Plano para a Resolução de Problemas 10](#_Toc226113768)

[7.7 Gerenciamento de Subcontratações 10](#_Toc226113769)

[7.8 Plano de Aperfeiçoamento 11](#_Toc226113770)

[8. Planos Adicionais 12](#_Toc226113771)

# Apresentação

## Sumário do Projeto

### Finalidades, Escopo e Objetivos

O intenso fluxo de automóveis nas metrópoles é um dos maiores desafios modernos. Não é incomum associar as grandes cidades a congestionamentos quilométricos, que já se tornaram rotina no cotidiano dos habitantes – São Paulo, por exemplo, conta com um acréscimo de 500 novos veículos diariamente.

Estudos mostram que congestionamentos podem causar problemas de saúde, como o estresse crônico. Além disso, eles contribuem para o aumento da poluição sonora e da poluição atmosférica. A primeira é causada pela combinação dos ruídos de motores e do uso insistente e excessivo das buzinas, e a segunda pela permanência demasiada de automóveis nas ruas.

Com base neste quadro, o ROTA – **R**eal-Time **O**ptimum **T**raffic **A**nalyzer – se propõe a simular o fluxo de automóveis em uma dada região. Com isso, informações importantes podem ser obtidas e analisadas acerca do tráfego, e soluções que procurem melhorar os congestionamentos podem ser tomadas de forma mais rápida e consciente.

### Postulados e Restrições

O produto apresenta restrições relacionadas ao prazo, à equipe e à tecnologia:

* Prazo de entrega: este é previsto para 01/07/2009;
* Equipe: ela é composta por três alunos da graduação que não podem dedicar tempo integral ao desenvolvimento do produto;
* Tecnologia: foi escolhida a plataforma Microsoft .NET.

### Liberações Parciais

O produto seguirá a seguinte programação:

* 01/04/2009: Plano para o Gerenciamento de Projeto de Software (PGPS);
* 18/05/2009: Especificação de Requisitos de Software (ERS) e Manual do Usuário (1ª versão);
* 08/06/2009: Projeto, Plano de Testes e Manual do Usuário (2ª versão);
* 01/07/2009: Versão Alfa do Sistema.

### Sumário de Cronograma e Orçamento

O projeto está dividido em quatro atividades: planejamento, análise, *design* e codificação. Esta divisão segue o Modelo de Processo em Cascata, também conhecido como *Ciclo de Vida Clássico*.

Posto que se trata de um projeto de graduação – em que os requisitos são fixos e o trabalho deve seguir até o fim de maneira linear – este é o modelo mais adequado para o produto.

No cronograma do projeto, estão previstos aproximadamente 16 dias para o planejamento, 33 dias para a análise, 15 dias para o *design* e 17 dias para a codificação. O cronograma completo pode ser encontrado no Apêndice I deste plano.

## Evolução do Plano

O planejamento a ser seguido é um dos fatores que contribuem para o sucesso do projeto. Ou seja, controlar a evolução do plano, a fim de implementar melhorias, significa assegurar a qualidade do produto final.

Dessa forma, o plano será analisado minuciosamente por todos os integrantes da equipe, assim como pelo gerente de qualidade, podendo sofrer modificações. Todas as alterações a serem feitas serão analisadas com cuidado por todos, e aquelas que sejam significativas e que contribuam para um melhor planejamento do produto serão consideradas.

# Referências

1. Pressman, R. S.: “Engenharia de Software”, 6ª edição, 2006, São Paulo, Editora McGraw-Hill
2. Microsoft Visual Studio 2008: *http://msdn.microsoft.com/en-us/vstudio/default.aspx*
3. Microsoft Visual C# 2008: *http://msdn.microsoft.com/en-us/vcsharp/default.aspx*
4. Microsoft .NET: *http://msdn.microsoft.com/en-us/netframework/default.aspx*
5. Microsoft Office Project 2007: *http://office.microsoft.com/pt-br/project/FX100487771046.aspx*
6. COCOMO II: *http://sunset.usc.edu/csse/research/COCOMOII/cocomo\_main.html*
7. Subversion: *http://subversion.tigris.org/*
8. TortoiseSVN: *http://tortoisesvn.tigris.org/*
9. Multithreading no Visual Studio: *http://msdn.microsoft.com/en-us/library/eed6swsx(VS.71).aspx*
10. Tutorial C#: *http://msdn.microsoft.com/en-us/library/aa288436.aspx*

# Definições

As siglas utilizadas no PGPS encontram-se listadas abaixo.

* DEL – Departamento de Engenharia Eletrônica e de Computação;
* PET – Programa de Engenharia de Transportes;
* UFRJ – Universidade Federal do Rio de Janeiro;
* COPPE – Instituto Alberto Luiz Coimbra de Pós-Graduação e Pesquisa de Engenharia;
* IDE – Integrated Development Environment;
* SCM – Software Configuration Management;
* RMMM – Risk Mitigation Monitoring and Management Plan.

# Organização do Projeto

## Interfaces Externas

Por se tratar de um projeto de graduação, não há, no momento, nenhuma relação com uma organização externa. Porém, o produto pode ser usado por empresas relacionadas a controle de tráfego e fluxo de automóveis, de maneira a melhorar o desempenho dos seus serviços.

## Estrutura Interna

A estrutura organizacional do projeto segue um modelo democrático: não há diferença de nível hierárquico entre os membros participantes – todos possuem o mesmo poder de decisão.

O professor Antônio Cláudio Goméz de Sousa, o qual faz parte do corpo docente do DEL/UFRJ, será responsável pelo controle de qualidade do produto, assim como pelo SCM.

Além disso, o professor Paulo Cezar Martins Ribeiro, que pertence ao corpo docente do PET/UFRJ, irá acompanhar o desenvolvimento da ferramenta, verificando a consistência da simulação, através de sua experiência na área de transportes; sendo assim, ele também participa do controle de qualidade do produto.

## Papéis e Responsabilidades

Todas as etapas do desenvolvimento e do processo de suporte serão feitas pela equipe, sem uma divisão específica das tarefas. Desta forma, cada integrante estará ciente de todo o processo – com isso, espera-se que o desempenho e o desenvolvimento sejam mais eficientes.

# Processos de Gerenciamento

## Partida no Projeto

### Previsões

As previsões do projeto foram feitas usando o software COCOMO II [6], baseando-se no modelo do primeiro estágio do projeto. Foi obtida uma estimativa de 2000 linhas de código em linguagem de alto nível, equivalente a um esforço nominal de 5,5 pessoas-mês e 3 pessoas-mês após o uso dos multiplicadores (ver Apêndice II).

A previsão do tempo de desenvolvimento fornecida pelo software foi de quatro meses. Usando um valor de R$5.000,00 para o salário mensal dos membros, estima-se um custo total de R$15.000,00 para o projeto.

### Equipe

A equipe necessária para o desenvolvimento do produto deve ser capaz de gerenciar e analisar o planejamento estipulado, seguindo o cronograma elaborado. Além disso, como não há etapas específicas para cada integrante, todos devem possuir as seguintes habilidades:

* Análise de Requisitos;
* Programação em C#;
* Familiarização com o ambiente de desenvolvimento Microsoft Visual C# 2008;
* Desenvolvimento de aplicações concorrentes e/ou multithreads;
* Capacidade de trabalhar em grupo.

### Plano para a Aquisição de Recursos

Não se aplica.

### Plano de Treinamento da Equipe

Todos os membros da equipe possuem as habilidades descritas na seção 5.1.2. Caso algum integrante sinta dificuldade em algumas delas, os seguintes passos podem ser seguidos:

* Caso a dificuldade seja na linguagem de programação C#, o tutorial disponível em [10] serve de apoio;
* Caso o obstáculo seja em relação ao ambiente de desenvolvimento escolhido, Microsoft Visual C# 2008, o site [3] possui excelentes referências;
* Informações sobre aplicações concorrentes e/ou multithreads, usando o ambiente de desenvolvimento escolhido, podem ser encontradas em [9].

É importante ressaltar que, devido à integração da equipe, as dúvidas podem ser tiradas dentro do próprio grupo de desenvolvimento.

## Plano de Trabalho

### Atividades

As atividades do projeto dizem respeito ao Modelo de Processo em Cascata [1]:

* Planejamento: atividade que estabelece um plano para o trabalho, incluindo o cronograma, os riscos prováveis, os recursos necessários e todas as informações relevantes referentes ao projeto e ao produto;
* Análise: atividade responsável por realizar uma modelagem que permita aos desenvolvedores entender melhor os requisitos de software; ou seja, a especificação de requisitos e a confecção do manual do usuário estão incluídas nessa atividade;
* *Design*: atividade responsável pela definição da infraestrutura e da arquitetura do sistema; um detalhamento maior das interfaces e dos módulos de apoio é necessário;
* Codificação: atividade que engloba a geração do código do programa e os testes necessários.

### Prazos

Os prazos estão detalhados no cronograma do projeto, o qual se encontra no Apêndice I deste plano.

### Alocação de Recursos

Em relação aos recursos humanos, três pessoas foram alocadas. No que diz respeito aos recursos materiais, os integrantes da equipe usarão seus computadores pessoais para desenvolver o produto.

### Alocação de Orçamento

A partir da utilização do software COCOMO II, considerando que os integrantes da equipe possuam um salário mensal de R$5.000,00, o orçamento do projeto foi estimado de, aproximadamente, R$17.000,00.

Todo esse orçamento se concentra nos recursos humanos, já que não são necessários outros recursos além daqueles já previstos para a equipe.

## Planos de Controle

### Controle dos Requisitos

Os requisitos do projeto e do produto serão verificados por toda a equipe durante as etapas de planejamento e análise, a fim de que eles sejam escolhidos corretamente.

Caso esses requisitos tenham que ser alterados, eles serão analisados por todos os membros da equipe, assim como pelos responsáveis pelo controle de qualidade.

### Controle dos Prazos

Os prazos serão respeitados, tendo como base o cronograma (Apêndice I). A ferramenta Microsoft Office Project 2007 [5] ajudará a equipe a seguir e controlar esses prazos.

### Controle do Orçamento

O controle do orçamento será feito automaticamente através do controle das atividades no tempo, conforme o cronograma especificado no Apêndice I.

### Controle de Qualidade

Para o controle de qualidade, o professor Antônio Cláudio fará revisões sobre os marcos, as quais serão realizadas durante os seminários do projeto (Apêndice I). Além disso, reuniões semanais serão realizadas com o professor Paulo Cezar, a fim de que as simulações sejam condizentes com a realidade.

### Plano de Relatórios

Além desse PGPS e da Especificação de Requisitos de Software, esse último a ser entregue ao final da atividade *Análise*, outros relatórios serão redigidos mensalmente dentro da própria equipe, para que os integrantes possam acompanhar o andamento do projeto, o estado dos requisitos e a qualidade do produto.

### Plano de Medidas

As horas de trabalho de cada integrante da equipe serão medidas, de forma a cumprir com os prazos estipulados. Além disso, outras medidas, como o esforço, os pontos de função e o número de linhas de código, serão calculadas, de maneira a serem validadas com a ajuda da ferramenta COCOMO II [6].

## Plano de Gerenciamento de Riscos

Todos os envolvidos no projeto de software participam da análise e gestão de riscos. A tabela de riscos, assim como os planos RMMM para cada um deles, encontra-se a seguir.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Risco | Categoria | Probabilidade | Impacto | RMMM |
| Aumento exagerado do número de linhas de código do simulador | Tamanho do produto | 50% | 3 | R1 |
| Prazo não atendido | Definição do processo | 40% | 2 | R2 |

**R1:**

* Como mitigar: Todo o desenvolvimento será o mais modular possível, para que, com uma maior organização, linhas desnecessárias sejam evitadas.
* Como monitorar: Toda a estrutura do código, assim como os algoritmos utilizados, serão analisados antes de serem implementados, para que linhas de código a mais sejam evitadas.
* Como gerenciar: Uma análise sobre o código será realizada, de forma a tentar reescrever o código, evitando o exagero de linhas.

**R2:**

* Como mitigar: Todos os integrantes da equipe planejarão seus horários, de modo que os prazos sejam cumpridos seguindo o cronograma definido.
* Como monitorar: Reuniões semanais serão realizadas para que o desenvolvimento possa ser monitorado e, assim, os prazos possam ser atendidos.
* Como gerenciar: Os integrantes realizarão uma análise sobre seus horários, de maneira a entregar o produto o mais rápido possível, mantendo a qualidade.

## Plano de Encerramento

O encerramento do projeto se dará pela entrega da versão alfa do produto aos usuários, que irão realizar testes no software, e pela revisão final feita pelo controle de qualidade. Além disso, a versão final do Manual do Usuário será entregue.

# Processos Técnicos

## Modelo dos Processos

Como especificado na seção 5.2.1, as principais atividades do projeto seguem o *Ciclo de Vida Clássico*, onde, ao final de cada uma das etapas, será realizado um seminário.

A data de realização do seminário corresponde ao prazo especificado para a realização da atividade à qual ele se refere. Além disso, após cada um desses marcos, serão feitas revisões para que as medidas necessárias à melhoria do projeto sejam tomadas.

## Métodos, Ferramentas e Técnicas

A metodologia de desenvolvimento será baseada no *Ciclo de Vida Clássico*, como já citado anteriormente.

A linguagem de programação escolhida foi C# (metodologia orientada a objetos). O ambiente de desenvolvimento será a IDE Microsoft Visual Studio 2008 [], que possui as ferramentas necessárias à fase de codificação.

## Infraestrutura

Além da infra-estrutura indicada no item anterior, a equipe usará a ferramenta de controle de versões Subversion [7] assim como a interface TortoiseSVN [8] para esse controle, com a finalidade de evitar sobreposição de arquivos e possíveis retrabalhos.

## Plano para a Aceitação do Produto

Para que a aceitação do produto seja maximizada, os seguintes passos devem ser seguidos:

* Lançamento da versão alfa no prazo esperado;
* Confecção de uma manual do usuário que seja claro e sucinto;
* Suporte aos usuários no caso de erros e/ou falhas.

# Planos para os processos de Suporte

## Gerenciamento de Configuração

Uma das ferramentas usadas para o gerenciamento da configuração, já citada na seção 6.3, é o Subversion. Com essa ferramenta, todas as versões e alterações do código-fonte são armazenadas, podendo-se voltar a um ponto anterior a qualquer instante, caso seja necessário. Além disso, toda a documentação e as referências usadas também foram incluídas nesse repositório, podendo ser acessadas a qualquer instante através da interface TortoiseSVN, instalada nos computadores pessoais dos integrantes da equipe.

É válido ressaltar que todos os documentos, como esse PGPS, possuem um controle de versões no início de cada arquivo, que inclui uma enumeração, os autores das alterações e as alterações propriamente ditas.

## Plano de Verificação e de Validação

A verificação do produto será feita pelo controle de qualidade, através das revisões dos marcos do cronograma, os quais se referem aos seminários do projeto.

Para a validação do produto, é necessário verificar se os requisitos pré-estabalecidos foram atendidos. Ou seja, não só as revisões sobre os marcos são importantes, como os Planos de Testes também são.

## Documentação

A documentação é composta dos seguintes itens:

* Itens não-liberáveis: PGPS e Plano de Testes;
* Itens liberáveis: Projeto do Sistema, Especificação dos Requisitos de Software e Manual do Usuário.

A elaboração da documentação é de responsabilidades dos membros da equipe e segue as normas disponíveis no site da disciplina.

## Plano para Assegurar a Qualidade

As principais medidas para assegurar o controle da qualidade baseiam-se nas revisões realizadas pelo professor Antônio Cláudio após a finalização de cada uma das atividades descritas na seção 5.2.1 e no acompanhamento em reuniões semanais com o professor Paulo Cezar, com experiência na área de transportes e na utilização de softwares com a finalidade proposta.

## Revisões e Auditorias

Para cada artefato produzido, Revisões Técnicas Formais serão realizadas pelo gerente de controle de qualidade do projeto. Essas revisões são feitas ao final de cada atividade, ou seja, durante os seminários.

## Plano para a Resolução de Problemas

Para que os erros sejam reportados, todos os integrantes da equipe de desenvolvimento estarão disponíveis para contato após a disponibilização da versão alfa. Com o *feedback* dos usuários, os *bugs* podem ser analisados e solucionados, de acordo com a prioridade dada pelos membros do projeto.

## Gerenciamento de Subcontratações

Não se aplica.

## Plano de Aperfeiçoamento

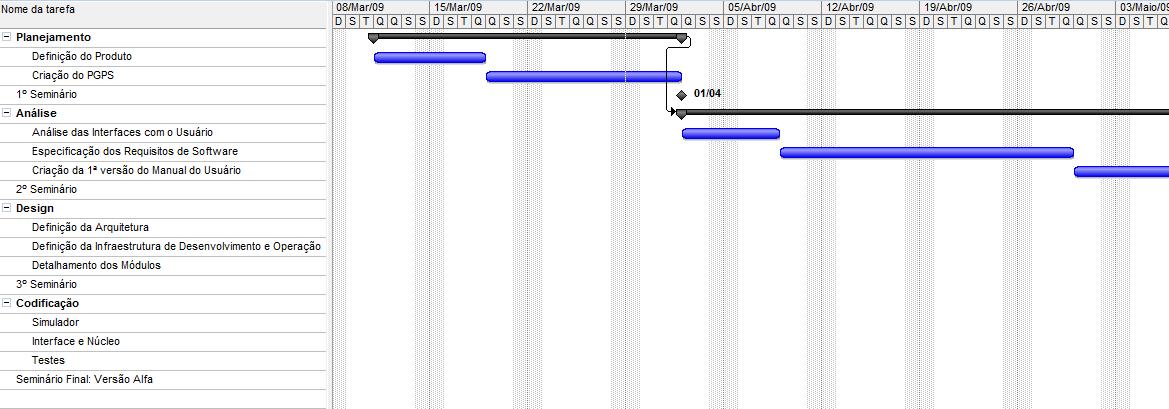
Para o aperfeiçoamento do projeto, novas técnicas e ferramentas poderão ser escolhidas durante o desenvolvimento do produto. Todavia, elas precisam passar por uma avaliação prévia feita pelos integrantes da equipe.

# Planos Adicionais

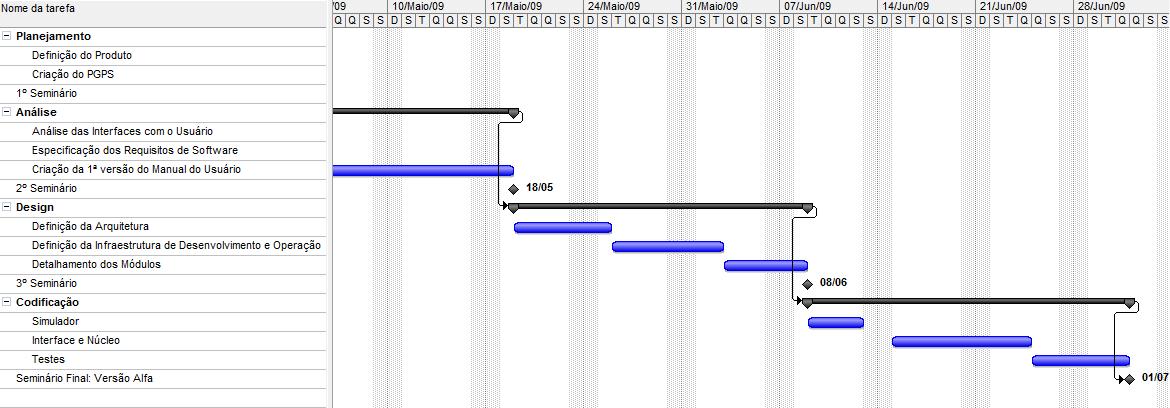
Não há planos a serem acrescentados neste documento.

# Apêndice I: Cronograma

## 11/03/09 a 06/05/09



## 06/05/09 a 01/07/09



# Apêndice II: Previsões baseadas no COCOMO II