TEN | Traffic Engine

Plano de Testes

**Versão 1.00**

06/07/2009

# Responsáveis

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

Antônio Cláudio Goméz de Sousa

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

Fernando Seabra Chirigati

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

Rafael Shinji Aoki Kikuchi

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

Talita Lopes Gomes

# Relatório de Mudanças

TEN – Traffic Engine

**Versão 1.00 – 06/07/2009**

Criação do Documento

# Índice

[1. Introdução 1](#_Toc234157461)

[1.1 Finalidade 1](#_Toc234157462)

[1.2 Definições, Acrônimos e Abreviaturas 1](#_Toc234157463)

[1.3 Referências 1](#_Toc234157464)

[2. Descrição Geral 2](#_Toc234157465)

[2.1 Itens 2](#_Toc234157466)

[2.1.1 Simulador 2](#_Toc234157467)

[2.1.2 Interface 2](#_Toc234157468)

[2.1.3 Atualizador 2](#_Toc234157469)

[2.1.4 Dados do Mapa e Dados dos Veículos 2](#_Toc234157470)

[2.2 Requisitos 2](#_Toc234157471)

[2.3 Visão geral 2](#_Toc234157472)

[2.4 Suspensão ou conclusão 3](#_Toc234157473)

[2.5 Ambiente 3](#_Toc234157474)

[2.6 Tarefas e cronograma 3](#_Toc234157475)

[2.7 Riscos e gerenciamento 3](#_Toc234157476)

[3. Especificações dos testes 4](#_Toc234157477)

[3.1 Confecção do Mapa 4](#_Toc234157478)

[3.1.1 Identificador 4](#_Toc234157479)

[3.1.2 Características 4](#_Toc234157480)

[3.1.3 Refinamento 4](#_Toc234157481)

[3.1.4 Identificador de caso de teste 4](#_Toc234157482)

[3.2 Definição de Parâmetros 5](#_Toc234157483)

[3.2.1 Identificador 5](#_Toc234157484)

[3.2.2 Características 5](#_Toc234157485)

[3.2.3 Refinamento 5](#_Toc234157486)

[3.2.4 Identificador de caso de teste 5](#_Toc234157487)

[3.3 Simulação 5](#_Toc234157488)

[3.3.1 Identificador 5](#_Toc234157489)

[3.3.2 Características 5](#_Toc234157490)

[3.3.3 Refinamento 6](#_Toc234157491)

[3.3.4 Identificador de caso de teste 6](#_Toc234157492)

[4. Casos de teste 7](#_Toc234157493)

[4.1 Caso de teste 1 7](#_Toc234157494)

[4.1.1 Identificador 7](#_Toc234157495)

[4.1.2 Itens 7](#_Toc234157496)

[4.1.3 Entradas e Saídas 7](#_Toc234157497)

[4.1.4 Ambiente 7](#_Toc234157498)

[5. Procedimentos de teste 8](#_Toc234157499)

[5.1 Procedimento 1 8](#_Toc234157500)

[5.1.1 Identificador 8](#_Toc234157501)

[5.1.2 Finalidade 8](#_Toc234157502)

[5.1.3 Necessidades especiais 8](#_Toc234157503)

[5.1.4 Ações 8](#_Toc234157504)

[5.1.5 Relatórios 8](#_Toc234157505)

# Introdução

## Finalidade

O TEN – Traffic Engine – é um software que simula o tráfego de veículos em uma região determinada pelo usuário. Dessa forma, análises podem ser realizadas para que congestionamentos freqüentes de uma grande cidade, por exemplo, possam ser amenizados. A região determinada pelo usuário deve ser confeccionada pelo mesmo através do próprio aplicativo.

A finalidade do plano de testes diz respeito a verificar a existência de erros, falhas e inconsistências no software, para que estes sejam evitados, garantindo a qualidade do sistema desenvolvido. Ele está dirigido para a equipe de desenvolvimento do aplicativo e para o gerente de qualidade do projeto, o professor Antônio Cláudio Goméz de Sousa.

## Definições, Acrônimos e Abreviaturas

* *Nó*: Ponto que define uma posição na área delimitada para a confecção de um mapa. Ele corresponde a uma extremidade de uma rua.
* *Aresta*: Conexão entre dois nós que representa uma rua do mapa.

## Referências

* *Plano de Gerenciamento de Projeto de Software – PGPS*

Versão: 1.00

Data: 01/04/2009

Responsável: Equipe de desenvolvimento do projeto

* *Especificação de Requisitos de Software – ERS*

Versão: 1.10

Data: 17/06/2009

Responsável: Equipe de desenvolvimento do projeto

* *Manual do Usuário*

Versão: 2.01

Data: 06/07/2009

Responsável: Equipe de desenvolvimento do projeto

* *Projeto de Software*

Versão: 1.10

Data: 06/07/2009

Responsável: Equipe de desenvolvimento do projeto

* *Implementando um Simulador de Tráfego Urbano para uma Interseção com Semáforos*

Glleddson Fryttys Menezes Leite, Antônio César Baleeiro Alves

Departamento de Computação – Universidade Católica de Goiás – UCG

Site da referência: *http://wsmartins.net/ermacs/trabalho\_18.pdf*

# Descrição Geral

## Itens

O sistema do TEN foi dividido em cinco módulos, os quais foram descritos no Projeto de Software do aplicativo. Assim, os testes a serem realizados têm o objetivo de identificar erros e falhas em cada um desses módulos, que estão descritos abaixo, juntamente com seus critérios de falha e correção.

### Simulador

Este módulo é responsável pelos algoritmos (cujo modelo encontra-se referenciado na Seção 1.3) e por todas as estruturas necessárias à simulação. Dessa forma, é imprescindível que a simulação ocorra de acordo com os algoritmos desenvolvidos. Caso alguma inconsistência seja encontrada, como, por exemplo, dois automóveis não respeitarem a distância de segurança determinada, a implementação dos algoritmos deve ser reavaliada.

### Interface

Este módulo é responsável por apresentar a interface gráfica para o usuário, por onde haverá todas as interações com o software. Assim, a interface deve estar consistente com os comandos acessados pelo usuário. Caso o usuário clique no botão de adicionar uma nova rua, que se encontra especificado no Manual do Usuário, esta ferramenta deve ser apresentada; caso isto não aconteça, o módulo apresenta uma falha em sua codificação.

### Atualizador

Este módulo é responsável por enviar pedidos de atualização periódicos à interface enquanto uma simulação está em andamento. Com isso, pretende-se manter uma sensação de continuidade para o usuário. Caso uma simulação se apresente ao usuário de forma não-contínua, ou seja, com intervalos de tempo significativos entre dois redesenhos, há indícios de que houve alguma falha no módulo.

### Dados do Mapa e Dados dos Veículos

Estes dois módulos são responsáveis por manter as estruturas de dados que contêm as informações referentes ao mapa e aos veículos da simulação, respectivamente. Dessa forma, assim como os parâmetros determinados pelo usuário, as informações obtidas durante a simulação não podem ser perdidos enquanto a seção estiver aberta.

## Requisitos

Os testes a serem realizados dizem respeito em verificar se os casos de uso do sistema, apresentados na Especificação de Requisitos de Software (ERS), funcionam sem falhas. Dessa forma, pretende-se garantir que todos os requisitos estabelecidos anteriormente sejam atendidos pelo software sem a presença de erros.

Devido à grande quantidade de casos de uso existentes, optou-se por não se considerar alguns deles; particularmente, não foram selecionados os Casos de Uso 3 (Aumentar Zoom) e 4 (Diminuir Zoom), uma vez que eles não apresentam alterações crucias nos resultados do software.

## Visão geral

As atividades de testes serão realizadas pelos integrantes da equipe que não estiveram demasiadamente ligados à codificação do sistema. Não será usada nenhuma ferramenta específica, uma vez que não há necessidade.

Para que alguns testes sejam realizados, arquivos *log* serão criados, de modo que informações importantes sejam coletadas internamente ao sistema.

## Suspensão ou conclusão

Haverá suspensão dos testes caso alguma falha seja encontrada. Neste caso, um dos integrantes da equipe de desenvolvimento deve corrigi-la para que os testes sejam retomados. Depois de realizados todos os testes e corrigidos todos os erros que o software eventualmente apresentou, pode-se concluir a etapa dos testes.

## Ambiente

Os testes serão realizados no mesmo ambiente em que o sistema estiver instalado, isto é, nos computadores pessoais da equipe de desenvolvimento do software.

## Tarefas e cronograma

Segundo o Plano de Gerenciamento de Projeto de Software (PGPS), a etapa de testes é composta por 5 dias. Dessa forma, foram identificadas três tarefas principais, cada uma delas relacionada a determinados requisitos. A tabela a seguir mostra as tarefas identificadas, juntamente com seus requisitos associados e seu prazo de duração estipulado.

|  |  |  |
| --- | --- | --- |
| Nome | Requisitos Associados | Prazo de Duração |
| Confecção do Mapa | Casos de Uso 1, 2, 7, 8 e 10 | 2 dias |
| Definição de Parâmetros | Casos de Uso 5, 6 e 9 | 1 dia |
| Simulação | Casos de Uso 11, 12, 13, 14, 15 e 16 | 2 dias |

Os prazos de duração envolvem não só a realização dos testes, como também a identificação e solução dos erros encontrados, caso haja.

## Riscos e gerenciamento

O principal risco existente diz respeito à presença de um erro que não foi identificado através dos testes realizados. Dessa forma, os usuários do sistema que encontrarem este erro devem reportá-lo à equipe de desenvolvimento do software para que ele seja corrigido imediatamente. Caso seja um erro que comprometa o uso e a eficiência de muitas funções do software, a correção do mesmo será enviada aos usuários o mais rápido possível; caso contrário, essa correção estará presente em uma próxima versão do aplicativo.

Além disso, há o risco do prazo estipulado para os testes não ser suficiente para a realização dos mesmos. Esse risco foi apresentado considerando o projeto como um todo no Plano de Gerenciamento de Projeto de Software (PGPS), na Seção 5.4. O plano de Risk Mitigation Monitoring and Management (RMMM) **R2** apresenta as diretrizes que devem ser tomadas caso o prazo se mostre insuficiente.

# Especificações dos testes

## Confecção do Mapa

### Identificador

O identificador único para o teste Confecção do Mapa é **T1**.

### Características

Neste teste, serão avaliados os seguintes casos de uso, os quais estão associados à confecção da região sobre a qual a simulação irá rodar:

* Caso de Uso 1 (Criar Rua)
* Caso de Uso 2 (Remover Rua)
* Caso de Uso 7 (Semaforizar um Ponto)
* Caso de Uso 8 (Remover Semáforo)
* Caso de Uso 10 (Criar Novo Mapa)

### Refinamento

Para que o teste **T1** tenha sucesso, alguns requisitos principais devem ser atendidos nos seguintes módulos do sistema:

* Interface: este módulo deve apresentar as ferramentas da interface correspondentes aos comandos acessados pelo usuário; à medida que o mapa é confeccionado, ele deve ser apresentado sem erros relacionados ao seu desenho, como, por exemplo, bordas das ruas e semáforos bem delimitados e intersecções das ruas bem representadas; por fim, quando um novo mapa for criado, a área de desenho e simulação deve ser limpa para que um novo mapa possa ser confeccionado.
* Dados do Mapa: quando os Casos de Uso 1 e 7 são executados, janelas de configuração são apresentadas ao usuário, caso ele queira modificar alguns parâmetros; se esses parâmetros são modificados, e os mesmos são válidos, eles devem ser imediatamente alterados internamente no sistema pelo módulo Dados do Mapa; um erro é identificado se essa alteração dos parâmetros não é capturada pelo sistema.

### Identificador de caso de teste

Os casos de teste associados a esta especificação são:

* C01 – Criação de uma rua
* C02 – Criação de uma rua com um número válido de vias
* C03 – Criação de uma rua com um número inválido de vias
* C04 – Criação de uma rua conectada a outra
* C05 – Criação de uma intersecção de mais de duas ruas
* C06 – Remoção de uma rua
* C07 – Criação de um semáforo em uma rua
* C08 – Criação de um semáforo em uma rua com uma temporização válida
* C09 – Criação de um semáforo em uma rua com uma temporização inválida
* C10 – Remoção de um semáforo em uma rua
* C11 – Criação de um novo mapa

## Definição de Parâmetros

### Identificador

O identificador único para o teste Definição de Parâmetros é **T2**.

### Características

Neste teste, estão associados os casos de uso relacionados à definição de parâmetros gerais da simulação e à alteração de outros parâmetros específicos. Os casos de uso estão listados abaixo.

* Caso de Uso 5 (Alterar Velocidade Máxima da Rua)
* Caso de Uso 6 (Definir Parâmetros Gerais)
* Caso de Uso 9 (Alterar Temporização do Semáforo)

### Refinamento

O teste **T2** terá sucesso caso os seguintes módulos seguirem alguns requisitos, como exposto abaixo.

* Interface: as janelas de configuração dos parâmetros devem ser apresentadas aos usuários sem que haja erros, para esses parâmetros possam ser alterados corretamente.
* Dados do Mapa: este módulo deve capturar a alteração dos parâmetros para que eles sejam modificados internamente no sistema; além disso, caso esses parâmetros sejam inválidos, sua alteração não seve ser permitida.

### Identificador de caso de teste

Os casos de teste associados ao teste **T2** encontram-se listados abaixo.

* C12 – Definição de uma nova velocidade máxima, válida, de uma rua
* C13 – Definição de uma nova velocidade máxima, inválida, de uma rua
* C14 – Definição de parâmetros gerais válidos
* C15 – Definição de parâmetros gerais, em que um deles seja inválido
* C16 – Definição de parâmetros gerais inválidos
* C17 – Definição de uma nova temporização, válida, de um semáforo
* C18 – Definição de uma nova temporização, inválida, de um semáforo

## Simulação

### Identificador

O identificador único para o teste Simulação é **T3**.

### Características

Neste teste, estão associados os casos de uso que estão relacionados com a simulação do tráfego de automóveis na região desenhada pelo usuário. Eles encontram-se listados abaixo.

* Caso de Uso 11 (Iniciar Simulação)
* Caso de Uso 12 (Pausar Simulação)
* Caso de Uso 13 (Retomar Simulação)
* Caso de Uso 14 (Parar Simulação)
* Caso de Uso 15 (Reiniciar Simulação)
* Caso de Uso 16 (Gerar Relatório)

### Refinamento

Alguns pontos devem ser verificados, em relação aos módulos do sistema, para que o teste **T3** tenha sucesso. Esses pontos estão listados abaixo.

* Simulador: este módulo deve garantir que a simulação em andamento esteja consistente com os algoritmos desenvolvidos; ou seja, os carros devem seguir as ruas, um carro não pode bater em outro, caso haja um semáforo, este deve ser respeitado, e assim por diante; além disso, ele deve garantir quando um comando de simulação está apto a ser executado (por exemplo, uma simulação não pode ser retomada se ela não está pausada).
* Interface: este módulo deve estar sempre se atualizando, através de pedidos do módulo Atualizador, enquanto uma simulação está em curso, para que a mesma se apresente de forma correta ao usuário, sem falhas; ou seja, não deve haver erros no desenho do mapa e dos veículos durante o curso de uma simulação.
* Atualizador: para que o módulo Interface apresente a simulação com uma sensação de cadência ao usuário, é necessário que o Atualizador esteja em constante atividade, já que ele indica à Interface a necessidade de atualizar os dados da simulação; assim, caso haja erros nesse módulo, a visualização da simulação pode ficar demasiadamente prejudicada.
* Dados do Mapa: a simulação em curso usa os parâmetros definidos pelos usuários, como a distância de segurança, por exemplo; logo, este módulo deve garantir a integridade desses parâmetros, de modo que o Simulador não obtenha dados inconsistentes para a simulação; além disso, grande parte desses dados é usada para a geração do relatório.
* Dados dos Veículos: os dados referentes aos veículos, como velocidade, não são determinados pelos usuários; porém, eles são indiretamente determinados por outros parâmetros, como distância de segurança, presença de semáforos, entre outros; logo, o módulo Dados dos Veículos não pode possuir erros no acesso a esses parâmetros; os dados dos veículos também são usados para a geração do relatório, que não pode estar inconsistente.

### Identificador de caso de teste

Os casos de teste associados a esta especificação são:

* C19 – Início de uma simulação
* C20 – Retomada de uma simulação em andamento
* C21 – Pausa de uma simulação
* C22 – Retomada de uma simulação pausada
* C23 – Término de uma simulação em andamento
* C24 – Término de uma simulação pausada
* C25 – Reinício de uma simulação em andamento
* C26 – Reinício de uma simulação pausada
* C27 – Geração de um relatório durante uma simulação em andamento
* C28 – Geração de um relatório durante uma simulação pausada
* C29 – Geração de um relatório sem a existência de uma simulação anterior

# Casos de teste

## Caso de teste 1

### Identificador

Identificador único do caso de teste.

### Itens

Itens e requisitos a serem testados no caso de teste.

### Entradas e Saídas

Especificação das entradas necessárias para o caso de teste, e das saídas esperadas.

### Ambiente

Necessidades de hardware,  software e outras necessidades para o caso de teste.

# Procedimentos de teste

## Procedimento 1

### Identificador

Oi!

### Finalidade

Indicar os casos de teste incluídos neste procedimento.

### Necessidades especiais

Necessidades especiais para a execução do procedimento.

### Ações

Ações necessárias para executar o procedimento: iniciar, executar, medir, suspender, recomeçar, parar ou concluir.

### Relatórios

Como serão apresentados e guardados os resultados dos testes (inclui um log).