

# Token: Uma Ferramenta para o Controle de Alterações em Projetos de Software em Desenvolvimento

LEONARDO GRESTA PAULINO MURTA  
MÁRCIO DE OLIVEIRA BARROS  
CLÁUDIA MARIA LIMA WERNER

COPPE / UFRJ – Departamento de Sistemas e Computação  
Caixa Postal: 68511 – CEP 21945-970 – Rio de Janeiro – RJ  
Fax / Telefone: 5521 590-2552  
{murta, marcio, werner}@cos.ufrj.br

## Abstract

This paper presents Token, a tool developed at COPPE/UFRJ to support change control in concurrent software development projects. Token synchronizes the access to components that compose a software project, inhibiting multiple users to change a specific component simultaneously. Token was designed to be accessible by the Internet. Its server side runs on top of Linux operating system, using the PHP3 scripting language, MySQL relational database and Apache web server.

**Palavras-Chave:** Controle de alterações, Gerência de configuração e desenvolvimento para Internet

## 1. Introdução

Os projetos de desenvolvimento de software contemporâneos têm progressivamente aumentado de tamanho e complexidade, sendo cada vez mais comum sua realização por equipes de médio porte (entre 10 e 20 desenvolvedores) e grande porte (acima de 20 desenvolvedores). Com as facilidades de comunicação proporcionadas pela Internet, a necessidade de experiência em diversas áreas de conhecimento e a pressão por cronogramas mais restritos, alguns projetos são realizados por diversas equipes trabalhando concorrentemente. Ainda mais, as dificuldades de reunir os especialistas necessários em um mesmo local físico e a delegação do desenvolvimento de determinados componentes para outras empresas, são exemplos de fatores que podem exigir que as equipes participantes de um projeto estejam geograficamente distribuídas.

Entretanto, a existência de mecanismos eficientes de comunicação, como a Internet, não soluciona os problemas de desenvolvimento concorrente de projetos de software. Ao contrário, o novo cenário traz novos problemas para o processo de desenvolvimento. Por exemplo, o trabalho concorrente de equipes geograficamente distribuídas dificulta o controle de alterações nos componentes de um projeto em desenvolvimento [PRE95]. Mesmo quando precedida de uma precisa definição das interfaces entre os componentes, a realização de um projeto pode exigir que diversos desenvolvedores alterem simultaneamente os mesmos componentes. Estas situações exigem a adoção de políticas para manter a consistência entre os componentes da versão atual do projeto ou permitir que essa consistência seja posteriormente restituída.

A ferramenta *Token* foi desenvolvida para apoiar o desenvolvimento concorrente de projetos de software, auxiliando na resolução dos problemas de controle de alterações nos componentes do projeto. Suas principais funcionalidades são: o cadastramento dos desenvolvedores participantes do projeto, a troca de informações entre estes desenvolvedores e o controle de alterações nos componentes do projeto. Token foi desenvolvida em ambiente **Linux**, utilizando a linguagem de script **PHP3**, o banco de dados **MySQL** e o servidor web **Apache**. A ferramenta é acessível via um navegador Internet, sendo independente da plataforma cliente.

Algumas das vantagens em utilizar o *Token*, ao invés de produtos comerciais, como por exemplo **Code Co-op**, **Visual SourceSafe**, **+1CM**, etc. [CMT00], são: o seu custo reduzido, por ser baseado em plataforma gratuita (**Linux**, **Apache**, **PHP3** e **MySQL**), ter como meio de comunicação a Internet, permitindo um desenvolvimento totalmente distribuído, e ser independente de plataforma, tanto no lado cliente como no lado servidor.

Este artigo está organizado em três seções, além desta introdução. A segunda seção trata do cadastramento dos desenvolvedores que participam de um projeto. A terceira seção apresenta os mecanismos de comunicação presentes na ferramenta *Token*. Em seguida, descrevemos os mecanismos de controle de alterações concorrentes. Finalmente, a seção 5 apresenta as perspectivas futuras para este trabalho.

## 2. Cadastramento de Desenvolvedores

*Token* fornece suporte para o cadastramento dos desenvolvedores participantes de um projeto. Este cadastramento é necessário para que os desenvolvedores tenham acesso aos recursos de comunicação oferecidos pela ferramenta (seção 3). Além disso, somente os desenvolvedores cadastrados terão acesso aos componentes do projeto, acessados através dos mecanismos de controle de alteração da ferramenta (seção 4).

O início da página de cadastramento de desenvolvedores lista as informações de todos os desenvolvedores previamente cadastrados como participantes do projeto. O restante da página permite a troca da senha do desenvolvedor corrente, o cadastramento de um novo desenvolvedor e a remoção de um desenvolvedor previamente cadastrado.

## 3. Mecanismos de Comunicação entre Desenvolvedores

Para facilitar a comunicação entre os desenvolvedores, *Token* oferece um quadro de mensagens. Para controlar um grande número de mensagens, estas são indexadas por assuntos. A página de mensagens, apresentada na Figura 1, permite visualizar as mensagens recebidas, enviar novas mensagens em determinados assuntos, adicionar e remover assuntos.

Sempre que uma mensagem é enviada através da ferramenta *Token*, os desenvolvedores cadastrados recebem uma notificação por e-mail. A notificação indica o assunto da mensagem, seu tipo e contém um *link* para a página de mensagens do *Token*. Os tipos de mensagem utilizados no *Token* são: pergunta, resposta, informação e urgente. Na versão atual da ferramenta, os tipos de mensagem são meramente informativos, permitindo ao leitor uma rápida identificação do objetivo da mensagem e a criação de filtros em seu e-mail.

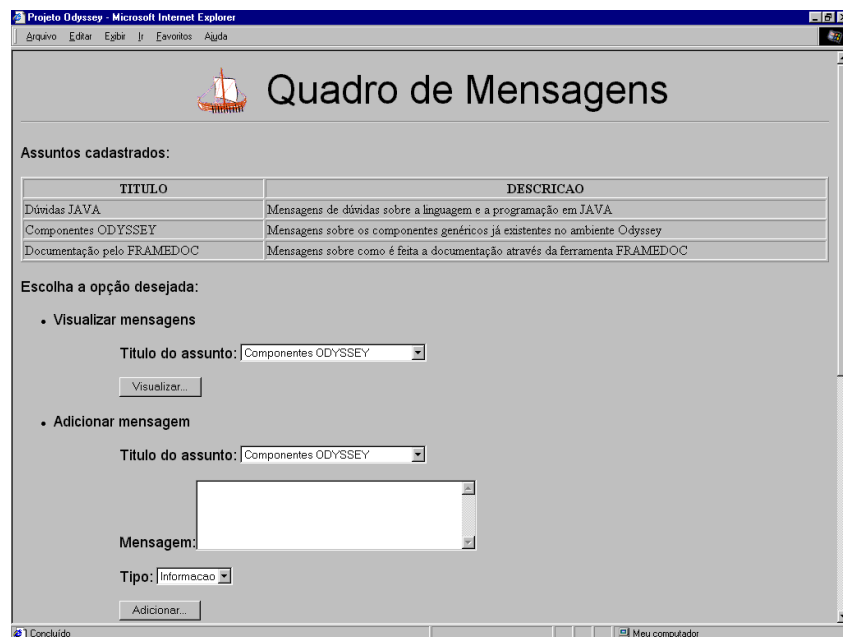


Figura 1: Página de comunicação entre usuários

## 4. Mecanismos de Controle de Alteração

As páginas de cadastramento de desenvolvedores e de mensagens servem como suporte para a principal funcionalidade oferecida pela ferramenta *Token*: permitir que um projeto seja

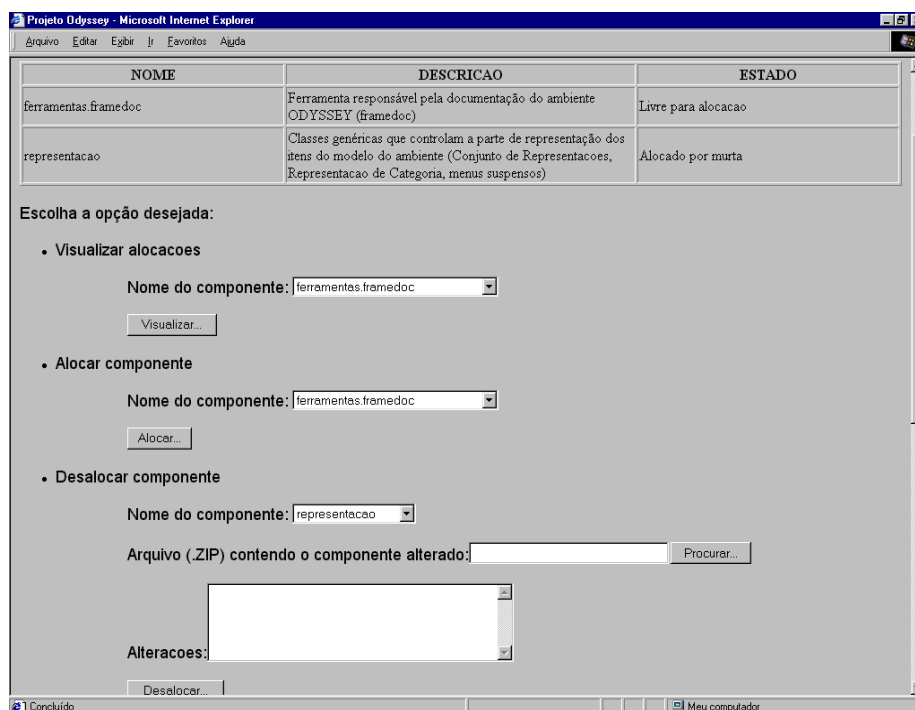
implementado simultaneamente por diversas pessoas geograficamente distribuídas.

A abordagem adotada consiste na divisão de um projeto em diversos componentes atômicos de software. Os componentes são acessíveis pelos desenvolvedores através da ferramenta, que permite que apenas um desenvolvedor altere um determinado componente a cada instante. Um componente é o elemento de menor granularidade controlado pela ferramenta.

*Token* segue a mesma abstração dos protocolos de comunicação das redes de computadores **Token Ring** e **Token Bus** [TAN96]. Nestas redes, diversos computadores compartilham um meio de comunicação comum, que deve ser utilizado por um computador de cada vez. Para sincronizar os acessos à rede, um pacote de dados padronizado, denominado *token*, permanece em tráfego na rede enquanto nenhuma mensagem é transmitida. Quando um computador deseja transmitir uma mensagem, ele retira o *token* da rede, envia sua mensagem, e, por fim, restaura o *token*, permitindo que outros computadores enviem suas mensagens.

Da mesma forma, através da ferramenta *Token*, os desenvolvedores podem alocar ou desalocar componentes. Cada componente possui um *token*, que se encontra disponível quando o componente não está reservado para nenhum desenvolvedor. Quando um desenvolvedor decide alterar um componente, ele deve pegar o *token* do componente. Sendo o *token* único por componente, apenas um desenvolvedor poderá alterar um determinado componente por vez. Se o componente desejado foi previamente reservado para um desenvolvedor, este deve liberar seu *token* para que outro desenvolvedor tenha permissão para alterá-lo.

A página de controle de alterações do *Token*, apresentada na Figura 2, exibe o estado de todos os componentes que estão sob custódia do *Token*. A página oferece comandos para visualizar as alocações efetuadas sobre um componente, alocar, desalocar, adicionar e remover componentes. *Token* controla a alocação de componentes no nível lógico, limitando o acesso aos componentes à presença de seus *tokens*, e no nível físico, disponibilizando os arquivos com o código fonte dos componentes.



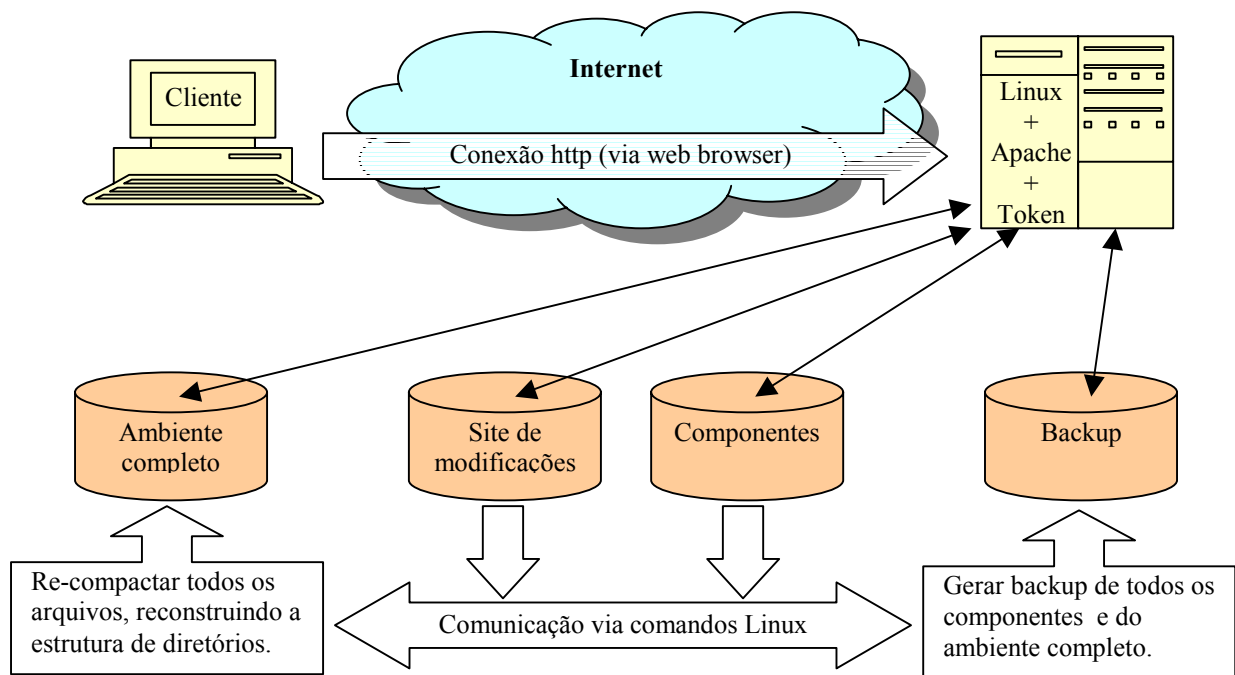
**Figura 2: Página de controle de alterações em componentes**

A desalocação de componentes é um processo complexo, onde *Token* reconstrói a estrutura do projeto a partir dos componentes existentes em sua base e do componente recentemente liberado. Para que um componente seja desalocado, é necessário que ele tenha sido previamente alocado pelo desenvolvedor corrente e que esse desenvolvedor informe o nome do componente, o arquivo

contendo seu código fonte e uma descrição das alterações que o componente sofreu. Os passos executados pela ferramenta na desalocação de um componente são:

- Fazer um *backup* do arquivo que contém o código fonte do componente, antes das alterações realizadas pelo desenvolvedor corrente;
- Atualizar uma página que indica todas as atualizações sofridas pelo componente, mantendo um histórico das alterações realizadas por cada desenvolvedor;
- Reconstruir o arquivo que contém o código fonte completo do projeto, a partir dos arquivos com o código fonte de seus componentes.

O diagrama apresentado na figura 3 exibe a forma de iteração do *Token* com as suas bases de dados e com o usuário, representado por uma conexão http vinda de uma máquina cliente.



0Figura 3: Diagrama de iteração do *Token*

## 5. Conclusões e Perspectivas Futuras

*Token* está sendo aplicado, com resultados satisfatórios, no desenvolvimento do Projeto Odyssey [WER99], cujo objetivo é a construção de uma infra-estrutura de desenvolvimento de software baseada na reutilização de modelos de domínio. O projeto está em desenvolvimento pelo grupo de reutilização de software da COPPE/UFRJ, contando com cerca de 10 pessoas, entre docentes, alunos de doutorado, mestrado e iniciação científica. O projeto possui cerca de cem mil linhas de código, em trezentas classes de objetos, escritas na linguagem Java. Recentemente, a ferramenta foi adotada pelo Projeto GOA++ [MAU97], em desenvolvimento pelo grupo de banco de dados da mesma instituição.

## Referências Bibliográficas

- [PRE95] Pressman, R. S.; "Engenharia de Software"; MAKRON Books; 1995
- [WER99] Werner, C.; Mattoso, M.; Braga, R.; Barros, M.; Murta, L.; Dantas, A; "Odyssey: Infra-estrutura de Reutilização baseada em Modelos de Domínio"; XIII SBES, Caderno de Ferramentas, Florianópolis, Outubro 1999; pp. 17-20
- [MAU97] Mauro, R. et al.; "GOA++: Tecnologia, implementação e extensões aos serviços de gerência de objetos"; XII SBBB, Fortaleza, Outubro 1997; pp. 272-286
- [TAN96] Tanenbaum, A.S.; "Computer Networks"; 3rd edition; Prentice Hall; March; 1996;
- [CMT00] Site na Internet em: <http://www.cmtoday.com/yp/commercial.html>