

Universidade Federal de Pernambuco  
Graduação em Ciência da Computação  
Centro de Informática

---



# VSFocus: Orientação a Tarefas para Microsoft Visual Studio

---

Proposta de Trabalho de Graduação

**Aluno:** João Augusto B. C. Alves ([jabca@cin.ufpe.br](mailto:jabca@cin.ufpe.br))

**Orientador:** André Luis Medeiros dos Santos ([alms@cin.ufpe.br](mailto:alms@cin.ufpe.br))

**Co-orientador:** Ryan Leite Albuquerque ([ryan@cesar.org.br](mailto:ryan@cesar.org.br))

Recife, Maio de 2007

## 1. Contexto

Enquanto sistemas de *software*, *frameworks*, e ferramentas continuam a se expandir indefinidamente, a habilidade de um desenvolvedor para processar informação tem um limite prático. A maioria dos ambientes de desenvolvimento (*IDEs*) modernos tratam esta questão com recursos para compilação incremental, visualizadores de estrutura, e busca textual. Apesar de estes mecanismos facilitarem a navegação em sistemas complexos, eles não dão nenhuma ajuda aos desenvolvedores para gerenciar o enorme volume de informação que precisa ser processado para concluir uma atividade de programação (1). Como resultado, as interfaces dessas *IDEs* tornam-se cada vez mais sobrecarregadas, dificultando o trabalho e diminuindo a produtividade.

Além disso, vale mencionar o importante papel da gerência de configuração no contexto das metodologias ágeis de desenvolvimento. Frequentemente gerência de configuração e controle de versão são consideradas pelos desenvolvedores atividades complexas em um processo de desenvolvimento, que podem atrapalhar o “trabalho de verdade” da programação. Na realidade, é um bom processo de gerência de configuração, aliado a um conjunto de ferramentas adequado, o que permite um trabalho em equipe efetivo (2).

Neste âmbito, é crucial notar a importância da integração entre ferramentas para o sucesso de uma equipe de desenvolvimento ágil. A integração mantém o ambiente de software mais padronizado e pode reduzir substancialmente o nível de mudanças de contexto, promovendo assim um maior desempenho em geral (3). Pondo em termos práticos: abrir um navegador web apenas para marcar um *bug* como corrigido pode se tornar algo desestimulante, além de ser um convite à perda de tempo (4).

Com as recentes variações de preço, as novas edições do *Visual Studio Team System*<sup>®</sup> (VSTS), um dos ambientes de desenvolvimento mais utilizados no mercado mundial de software (5), estão se tornando cada vez mais inacessíveis para pequenas empresas, além de fornecer recursos muitas vezes desnecessários para a realidade dessas organizações. Porém, versões mais restritas, como o *Visual Studio Standard Edition*<sup>®</sup> (VSSE), não apresentam recursos para trabalho em equipe, nem mesmo serviços mais básicos como controle de versão, rastreamento de tarefas (*issue tracking*) e testes unitários.

Por outro lado, muitos desses serviços podem ser providos através de componentes de software livre/gratuito (6), que em sua maior parte não apresentam integração com *IDEs* comerciais. Exemplos populares desses sistemas são: o *Subversion* (7), que possui certo nível de integração com o VSSE; o *Trac* (8); e o *NUnit* (9), que também possui *plug-ins* para integração com o VSSE.

## 2. Escopo e Objetivos

Neste contexto, o projeto propõe o desenvolvimento de um protótipo de ferramenta para integração entre o ambiente *Visual Studio 2005*<sup>®</sup> (VSSE 2005) e sistemas de *issue tracking*,

tomando como inspiração o projeto *Mylar* (10). Possibilitando aos desenvolvedores um melhor gerenciamento de suas atividades e mais agilidade na comunicação, além de ajudar a manter o foco no contexto das tarefas em andamento (4) (11).

O protótipo proposto deve incluir as seguintes características:

- Gestão de tarefas
  - Gestão de repositórios
  - Painel de tarefas
  - Busca e filtragem (desejável)
  - Contexto (desejável)
- Suporte a *issue trackers*
  - Trac
    - Consultas
    - Modo *offline*
    - Notificações
  - Bugzilla (desejável)
  - Google Code Hosting (desejável)
  - Mantis (desejável)

### 3. Cronograma

Atividades	Maio	Junho	Julho	Agosto
Levantamento e análise de material bibliográfico	■	■		
Design da aplicação proposta		■		
Implementação		■	■	
Testes			■	■
Redação do trabalho		■	■	■
Preparação da apresentação				■

### 4. Referências

1. **Kersten, Mik.** Task Focused Programming with Mylar, Part 1: Integrated task management for Eclipse. *IBM developerWorks*. [Online] 10 de Outubro de 2006. <http://www-128.ibm.com/developerworks/java/library/j-mylar1/>.
2. **Berczuk, Steve e Appleton, Brad.** *Software Configuration Management Patterns: Effective Teamwork, Practical Integration*. s.l. : Addison-Wesley, 2003. 0201741172.
3. **Spolsky, Joel.** Human Task Switches Considered Harmful. *Joel on Software*. [Online] 12 de Fevereiro de 2001. <http://www.joelonsoftware.com/articles/fog0000000022.html>.

4. **Coté, Michael.** More on Mylar: Tasks, Picking, Agile ALM. *People Over Process*. [Online] 12 de Junho de 2006. <http://redmonk.com/cote/2006/12/06/more-on-mylar-tasks-picking-agile-alm/>.
5. **Evans Data Corp.** *Developers' Choice IDE Scorecard*. 2006.
6. **Gunderloy, Mike.** Three must-have tools: Version control, issue tracking, unit testing. [Online] 23 de Maio de 2005. [http://whatis.techtarget.com/tip/0,289483,sid8\\_gci1091076,00.html](http://whatis.techtarget.com/tip/0,289483,sid8_gci1091076,00.html).
7. *Site do Subversion*. [Online] <http://subversion.tigris.org/>.
8. *Site do Projeto Trac*. [Online] <http://trac.edgewall.org/>.
9. *Site do NUnit*. [Online] <http://www.nunit.com/index.php>.
10. *Site do Projeto Mylar*. [Online] <http://www.eclipse.org/mylar/>.
11. **Kersten, Mik.** Task Focused Programming with Mylar, Part 2: Automated context management for Eclipse. *IBM developerWorks*. [Online] 10 de Outubro de 2006. <http://www.ibm.com/developerworks/java/library/j-mylar2/>.

## 5. Data e Assinaturas

Recife, 18 de maio de 2007

---

André Luis Medeiros dos Santos (Orientador)

---

João Augusto de Britto Cavalcanti Alves (Proponente)

---

Ryan Leite Albuquerque (Co-orientador)