

3

Estudo de Ferramentas

Existem diferentes abordagens para automatizar um processo de desenvolvimento. Um conjunto de ferramentas pode ser utilizado para aperfeiçoar o trabalho, mantendo os desenvolvedores focados no próprio desenvolvimento e não em ações paralelas.

Não é objetivo deste trabalho criar, do zero, uma ferramenta de gerência de projeto, mas sim estudar o funcionamento de um workflow dinâmico. Esta seção apresenta o estudo sobre alguns dos sistemas de workflow existentes hoje e difundidos no mercado de trabalho, estudando os fluxos de seus workflows e a flexibilidade existente neles. Após o estudo, foi selecionada a ferramenta que se adéqua melhor às necessidades do projeto. Também foram estudadas outras funções existentes como a possibilidade de criar *wikis*, fóruns, ou outra funcionalidade que agreguem valor ao trabalho.

3.1.

JIRA

JIRA é uma aplicação J2EE de gerência de projetos com acompanhamento e gestão de problemas. Suas funcionalidades são apresentadas abaixo.

Gestão de problemas, tarefas e melhorias.

O Jira permite a gerência de pendências (*issues*) em um projeto. A pendência pode ser relativa a um *bug*, nova funcionalidade, tarefa ou melhoria. O Jira permite criar novos tipos de pendências caso seja necessário para o projeto.

Create Issue

Enter the details of the issue...

Project: Projeto do Mestrado

Issue Type: New Feature

* Summary:

Priority: **Major**

Due Date:

Component/s: Unknown

Affects Version/s: Unknown

Fix Version/s: Unknown

Assignee: **- Automatic -** [Assign to me](#)

* Reporter: **admin**
Start typing to get a list of possible matches.

Environment:
For example operating system, software platform and/or hardware specifications (include as appropriate for the issue).

Description:

Concluído

Ilustração 4 - Tela de Criação de Nova Pendência

Depois de criadas, as pendências podem possuir os estados de Aberta, Em Progresso e Resolvida. A pendência pode ser resolvida das seguintes maneiras: terminada, não será terminada, duplicada, incompleta e não foi possível reproduzir.

O Jira é bastante personalizável, os tipos das pendências (Ilustração 5), assim como várias outras informações do sistema, também podem ser alterados, permitindo maior adaptação da ferramenta ao usuário.

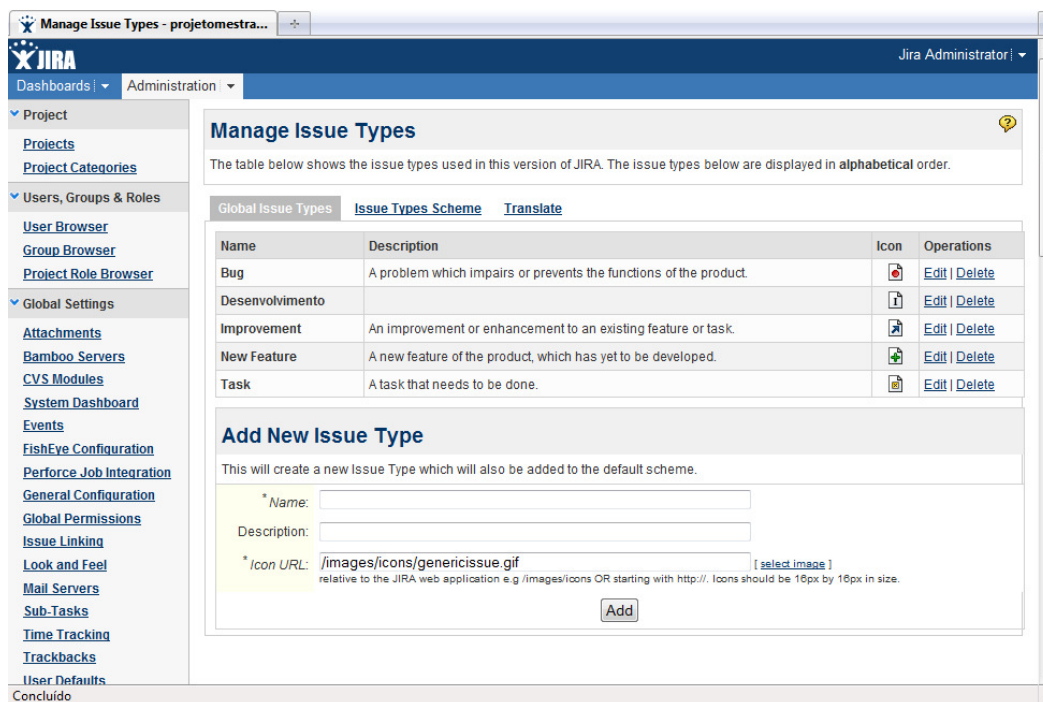


Ilustração 5 - Tela de Gerência de Tipos de Pendências

Realiza o acompanhamento de componentes e versões

O Jira é totalmente integrado com sistemas de gerência de versão como o CVS ou *Subversion*. Através do próprio sistema é possível acompanhar o histórico de alterações dos artefatos do projeto e mapeá-los com as pendências que motivaram suas alterações.

Suporte a arquivos anexos.

Dentro de um projeto ou da própria pendência é possível criar anexos de documentos e arquivos, permitindo que toda documentação do projeto seja armazenada

Busca textual e filtros.

A Ilustração 6 mostra a tela de definição dos critérios de filtro de pendências. Os filtros permitem a seleção de quais pendências aparecerão na listagem principal. O sistema oferece a possibilidade de salvar diversos filtros para serem utilizados a qualquer momento.

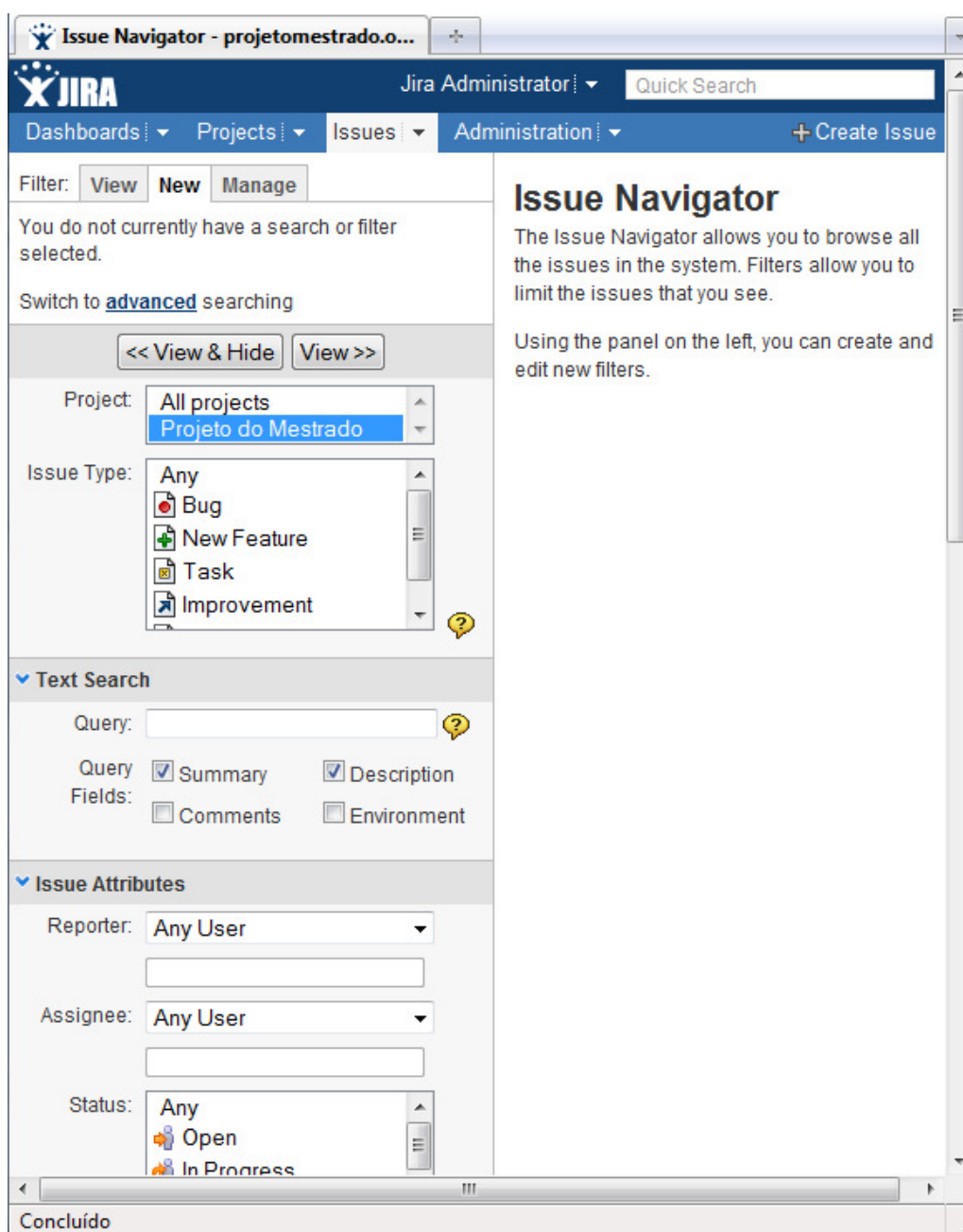


Ilustração 6 - Tela de criação de filtro

Notificações por e-mail.

É possível manter os usuários constantemente informados sobre as alterações no projeto através do envio de emails automaticamente a cada modificação no projeto.

O envio de emails é bastante configurável. A Ilustração 7 apresenta a tela com a configuração padrão de envio de email. O usuário pode criar para seu projeto novas configurações de envio de emails, definindo quais usuários recebem quais informações.

Edit Notifications — Default Notification Scheme

On this page you can edit the notifications for the "Default Notification Scheme" notification scheme.

☐ [Add notification](#)

☐ [View all notification schemes](#)

Event	Notifications	Operations
Issue Created (System)	<input type="checkbox"/> Current Assignee (Delete) <input type="checkbox"/> Reporter (Delete) <input type="checkbox"/> All Watchers (Delete)	<input type="checkbox"/> Add
Issue Updated (System)	<input type="checkbox"/> Current Assignee (Delete) <input type="checkbox"/> Reporter (Delete) <input type="checkbox"/> All Watchers (Delete)	<input type="checkbox"/> Add
Issue Assigned (System)	<input type="checkbox"/> Current Assignee (Delete) <input type="checkbox"/> Reporter (Delete) <input type="checkbox"/> All Watchers (Delete)	<input type="checkbox"/> Add
Issue Resolved (System)	<input type="checkbox"/> Current Assignee (Delete) <input type="checkbox"/> Reporter (Delete) <input type="checkbox"/> All Watchers (Delete)	<input type="checkbox"/> Add
Issue Closed (System)	<input type="checkbox"/> Current Assignee (Delete) <input type="checkbox"/> Reporter (Delete) <input type="checkbox"/> All Watchers (Delete)	<input type="checkbox"/> Add
Issue Commented (System)	<input type="checkbox"/> Current Assignee (Delete) <input type="checkbox"/> Reporter (Delete) <input type="checkbox"/> All Watchers (Delete)	<input type="checkbox"/> Add
Issue Comment Edited (System)	<input type="checkbox"/> All Watchers (Delete) <input type="checkbox"/> Reporter (Delete) <input type="checkbox"/> Current Assignee (Delete)	<input type="checkbox"/> Add
Issue Reopened (System)	<input type="checkbox"/> Current Assignee (Delete) <input type="checkbox"/> Reporter (Delete) <input type="checkbox"/> All Watchers (Delete)	<input type="checkbox"/> Add

Localizar: mail [Próxima](#) [Anterior](#) [Realçar tudo](#) ☐ Diferenciar maiúsculas/minúsculas

Concluído

Ilustração 7 - Tela de Configuração de Envio de Emails

Compatível com praticamente qualquer banco de dados relacional.

Os bancos de dados suportados pelo Jira são:

- SQL Server 2005
- SQL Server 2008
- MySQL
- PostgreSQL
- Oracle
- HSQLDB

Fácil de estender e integrar com outros sistemas.

O Jira permite a inclusão de *plugins* dentro do seu projeto. Um *plugin* é uma extensão da ferramenta que provê novas funcionalidades. A Ilustração 8 apresenta a interface gráfica para a inclusão de novos *plugins* na ferramenta.

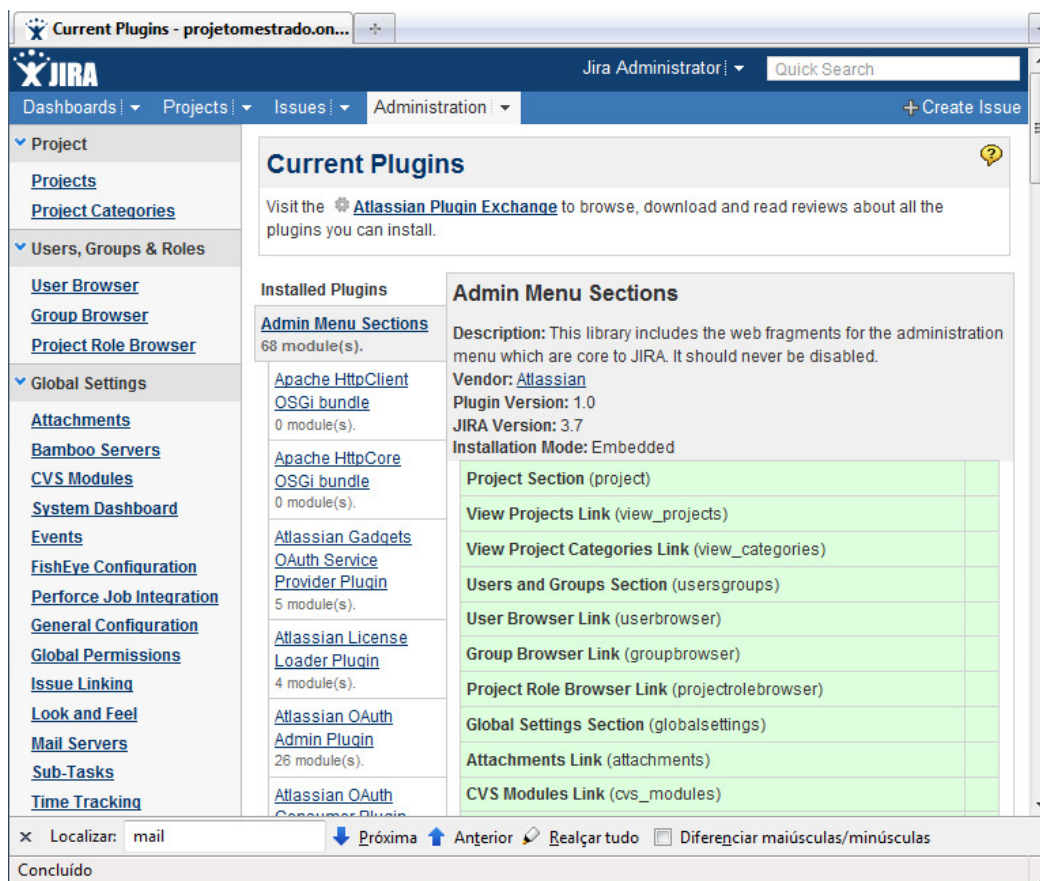


Ilustração 8 - Tela de Configuração de Plugins

O Jira também possui conectores que permitem a integração com outros sistemas como o *Confluence* ou o *MS Project*.

3.1.1.

Workflow

O Jira permite criar vários workflows diferentes definindo o fluxo de suas pendências. Um workflow padrão já vem em sua instalação definindo o fluxo dos status das pendências.

View Workflow Steps — jira

This shows all of the steps for jira.

Not editable because jira is a system workflow.

☐ View all [workflows](#).

☐ View all [statuses](#).

Step Name (id)	Linked Status	Transitions (id)	Operations
Open (1)	Open	Start Progress (4) >> In Progress Resolve Issue (5) >> Resolved Close Issue (2) >> Closed	View Properties
In Progress (3)	In Progress	Stop Progress (301) >> Open Resolve Issue (5) >> Resolved Close Issue (2) >> Closed	View Properties
Resolved (4)	Resolved	Close Issue (701) >> Closed Reopen Issue (3) >> Reopened	View Properties
Reopened (5)	Reopened	Resolve Issue (5) >> Resolved Close Issue (2) >> Closed Start Progress (4) >> In Progress	View Properties
Closed (6)	Closed	Reopen Issue (3) >> Reopened	View Properties

Ilustração 9 - Visão Geral de um Workflow no Jira

Um workflow possui as seguintes informações:

- Passos (steps): definindo os estados do workflow
- Quais os status das pendências que estão associados a um passo,
- As transições definidas para estes. Cada transição altera o status da pendência para o status alvo da transição.

No momento da criação de uma transição, é possível definir qual o status alvo da pendência e se será utilizada uma tela de transição no momento de sua execução.

Add Workflow Transition

Create a transition from **Open** to another step.

Transition Name:

Description:

Destination Step:

Transition View:

The screen that appears for this transition (if any).

Ilustração 10 - Criação de Nova Transição no Workflow

Na tela de transição o usuário pode inserir informações adicionais, caso sinta necessidade.

Para associar um workflow a um projeto, primeiro é necessário criar um *Scheme*. Dentro de um *Scheme*, vários workflows podem ser associados aos tipos de pendências diferentes. Permitindo que cada um possua um workflow diferente.

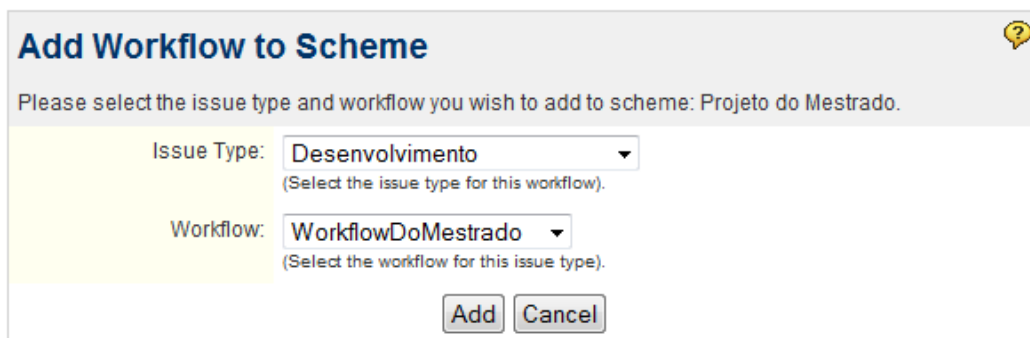


Ilustração 11 - Adicionar Workflow em um *Scheme*

3.1.2.

Avaliação

A ferramenta é bastante prática e completa. Alguns pontos negativos foram encontrados na ferramenta:

- É uma ferramenta paga, existindo apenas uma versão gratuita para teste.
- Não possui criação de caminhos críticos, associando tarefas que são pré-requisitos de outras.
- A versão em português da ferramenta não está muito bem internacionalizada. Existem muitos termos em português e inglês que aparecem misturados no texto.

O workflow da ferramenta é bastante flexível. É possível criar vários workflows diferentes para um mesmo projeto, definindo workflows específicos para pendências de tipos diferentes e utilizando o recurso de telas de transições para as transições específicas. O lado negativo do workflow é que ele está atrelado à mudança de status da ferramenta. Caso deseje-se realizar regras de transições para a mudança de outras informações de uma pendência, como alteração da sua prioridade, não é possível.

3.2.

Redmine

Redmine é uma ferramenta web, *open source*, voltada para gerência de projetos e *bug-tracker*. Ela inclui calendários e diagramas de *gant*, adicionando representações visuais de projetos e seus *deadlines*.

O design do *Redmine* é significativamente influenciado pelo *Trac*, um sistema similar. O *Redmine* foi escrito em *Ruby on Rails*.

O *Redmine* possui as seguintes funcionalidades:

- Suporte a múltiplos projetos
- Flexibilidade para controle de permissões
- Sistema de controle de tarefas flexível
- Diagrama de Gantt e calendário
- Novidades, documento e sistemas de gerência de arquivo
- Feeds e notificações por email.
- Wiki por projeto
- Fóruns por projeto
- Funcionalidades de controle de tempo simples
- Campos personalizados para tarefas, projetos e usuários
- Integração com sistemas de controle de versão (SVN, CVS, Git, Mercurial, Bazaar e Darcs)
- Suporte a múltiplas autenticações LDAP
- Suporte a auto-registro de usuários
- Suporte a várias línguas
- Suporte a vários bancos de dados
- Um projeto no Redmine pode possuir os seguintes módulos:
- Issue tracking: Controle de tarefas. Caso o bugtracker faça parte do projeto.
- Documents: Deve ser ativado quando o projeto possuir um repositório de documentos
- Repository: Permite associar o projeto a um repositório de versionamento.
- Time tracking: Disponibiliza um diagrama de gantt para o projeto.
- Files: Permite anexar arquivos ao projeto
- News: Apresenta uma área de novidades para o projeto

- Wiki: permite a criação de um wiki para o projeto

Para este trabalho, será criado um projeto com todos os módulos para permitir o estudo de toda a ferramenta.

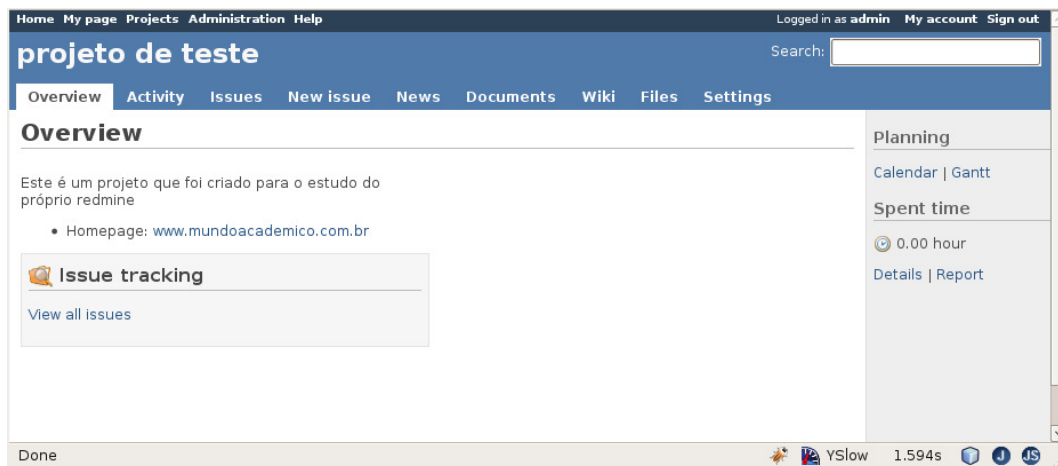


Ilustração 12 - Tela inicial de um projeto

A Ilustração 12 apresenta a tela inicial do projeto criado. Seus módulos são apresentados em abas. Na barra lateral é apresentado o diagrama de *gannt* e os relatórios de horas gastas no projeto.

A Página Inicial do projeto apresenta uma visão geral de todo o projeto. Na parte superior à esquerda no *Issue tracking* provê uma visão de quantas tarefas estão sendo executadas ou fechadas e de quais tipos. Na área *Members* pode-se ver quais os usuários que participam do projeto. Na parte de *Latest news* pode-se ver quais as ultimas noticias do projeto.

A aba *Activity* apresenta cada alteração realizada no projeto. Dentre elas alterações no wiki, mudanças nos status de tarefas, dentre outros.

A área de atividades provê um *log* do histórico de todas as atividades realizadas no projeto, incluindo:

- Bugs e Acertos
- Commits no repositório
- Novidades

Os itens acima são apresentados por padrão. Mas é possível ativar uma lista de mudanças realizadas no *wiki* e no fórum.

O módulo de Novidades apresenta a possibilidade de novidades serem inseridas de maneira rápida no projeto, mantendo seus participantes informados sobre mudanças realizadas. Os participantes podem incluir comentários nas novidades.

O *Redmine* também possui um módulo de estatísticas de repositório. Ele gera duas imagens a partir do sistema de versionamento, representando o número de mudanças (número de arquivos alterados) e de revisões (número de *commits* no repositório) do repositório configurado. Uma imagem mostra uma comparação entre meses de mudanças X revisões. A outra imagem mostra o número de mudanças e revisões por usuário do repositório.

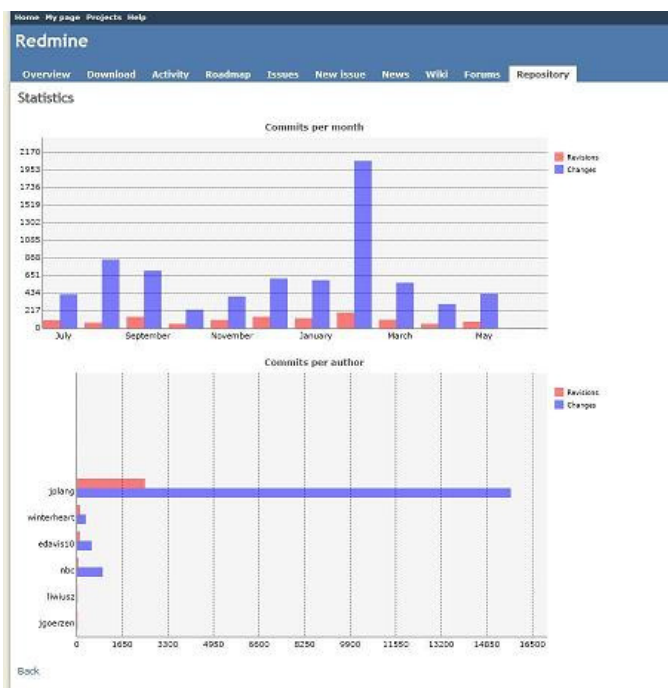


Ilustração 13 - Tela de estatísticas do repositório

3.2.1.

Workflow

Para o *Issue tracker* funcionar, é necessário realizar algumas configurações. Como a ferramenta é totalmente flexível, no primeiro *login* é necessário definir todo seu workflow.

Primeiro deve-se definir o *Tracker*, que é o container com o workflow. Para o workflow, devem-se definir os estados das tarefas e suas regras de transição. A vantagem do Redmine é que diferentes workflows podem ser definidos para diferentes projetos. Logo é possível criar perfis de projetos de acordo com suas necessidades.

Os workflows das tarefas na ferramenta são extremamente flexíveis. Cada estado e transição são configurados pelo próprio usuário. A Ilustração 14 apresenta a tela de cadastro de status das tarefas (encontrada na área administrativa da ferramenta). Todos os status podem ser inseridos na ordem

desejada. É possível definir quais status são padrões, ou seja, no momento da criação da tarefa, os status padrões podem ser atribuídos à tarefa. É possível definir também se um status representa uma tarefa fechada. Quando todas as tarefas de uma iteração possuem um status definido como “Tarefa Fechada”, então a iteração está concluída.

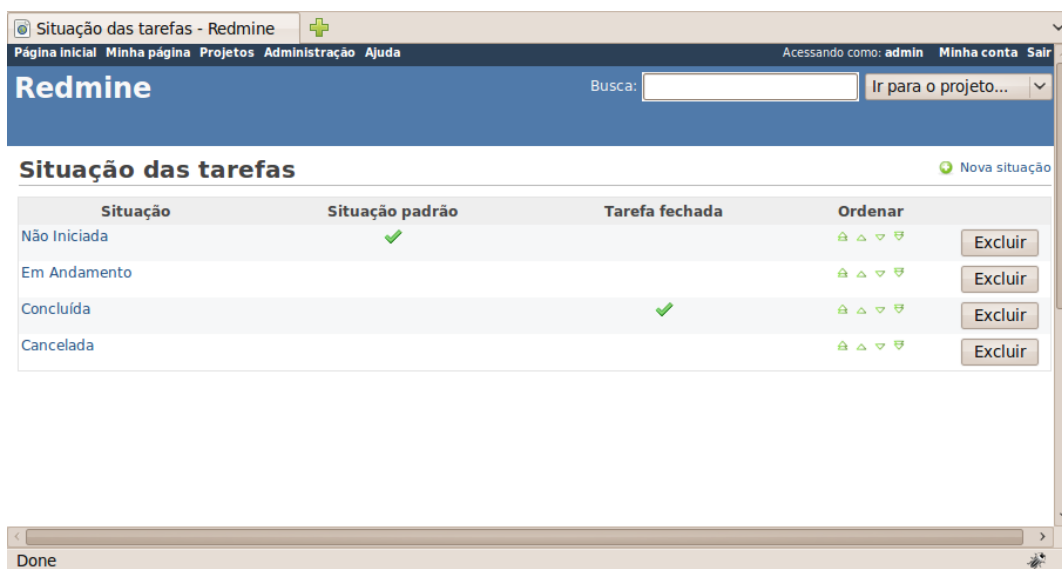


Ilustração 14 - Tela de Configuração dos Status das Tarefas

Com os status de uma tarefa definidos, o usuário pode entrar na tela de criação de fluxo de trabalho (*workflow*) para definir quais são as transições entre os status (Ilustração 15). É possível inclusive definir um workflow diferente para cada tipo de tarefa.

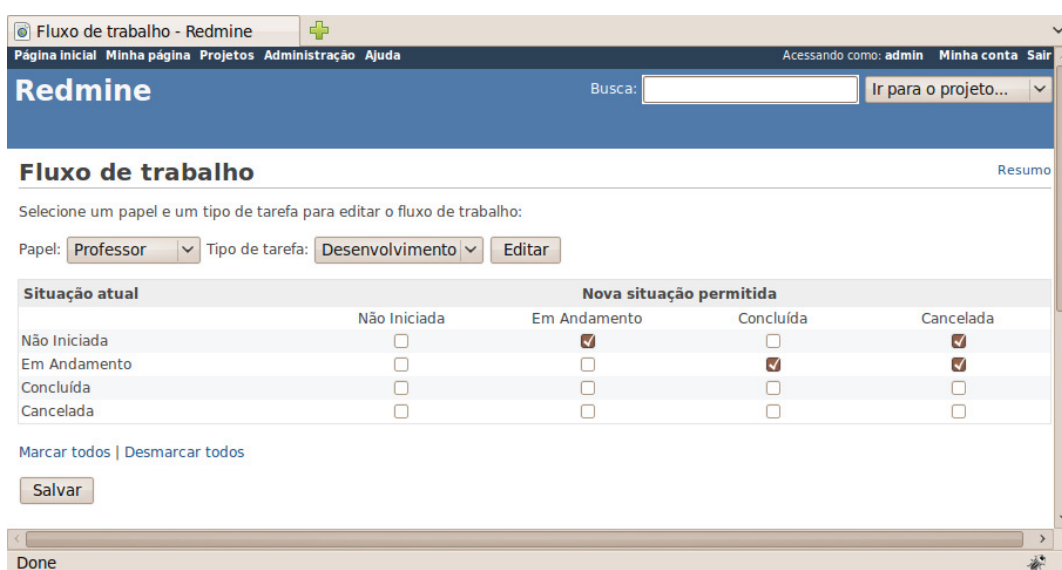


Ilustração 15 - Tela de Configuração do Workflow

Com o workflow criado, é necessário definir quais as categorias e prioridades das tarefas. Foram criadas as prioridades Alta, Média e Baixa e as categorias Desenvolvimento, Teste, Estudo e Documentação.

Ilustração 16 - Tela de criação de tarefa

No cadastro de uma nova tarefa, são dadas todas as informações necessárias ou disponíveis para a mesma. É definido o *Tracker*, título, descrição, status, prioridade etc.

3.2.2.

Avaliação

A ferramenta possui um alto grau de personalização, provendo cada projeto com alta flexibilidade. Também é facilmente extensível.

O *Redmine* tira proveito da arquitetura do framework *Ruby on Rails* e da sua facilidade de inclusão de *plugins*. Um *plugin* reflete uma funcionalidade extra que, através de uma linha de comando, é possível inserir na sua ferramenta. O *Redmine* também apresenta uma interface gráfica para a gerência dos mesmos na área administrativa. Dentre os vários *plugins*, existem inclusões de blogs em um projeto, criação de rastreamentos de ferramentas Google, criação de votos

para as tarefas, dentre outros. A facilidade de inclusão de *plugins* facilita ainda mais a extensão e personalização da ferramenta.

Por ser uma ferramenta open source, seu código pode ser utilizado e alterado livremente.

3.3.

Google Code

Google Code é uma ferramenta web gratuita disponibilizada pelo Google que permite o controle de projetos *open-source*. As funcionalidades da ferramenta são listadas abaixo e mais bem detalhada ao longo da seção.

- Criação de novos projetos em qualquer tópico.
- Espaço de um gigabyte para armazenamento de código versionado através dos gerenciadores de versão Subversion ou Mercurial.
- Ferramentas de integração de busca de diferentes revisões de código, para facilitar a visualização, contribuição com o projeto, e manutenção de código de alta qualidade.
- Um sistema de gerência de issues e wiki.

No *Google Code*, o usuário pode cadastrar *issues* (problemas) encontrados no projeto definindo sua categoria como defeito reportado por usuário ou por desenvolvedor (Ilustração 17). Quando um novo *issue* é cadastrado, o sistema apresenta um *template* padrão no campo de descrição, orientando na descrição do cenário que deve ser executado para reproduzi-lo.

Google Code **projetodeteste123**
Projeto para estudo da ferramenta

Project Home Downloads Wiki **Issues** Source Administer

New Issue | Search Open Issues for Search | [Advanced Search](#) | [Search Tips](#)

Template: Defect report from developer

Summary: Enter one-line summary

Description: What steps will reproduce the problem?

- 1.
- 2.
3. |

What is the expected output? What do you see instead?

Please use labels and text to provide additional information.

[Attach a file](#) [Notify me of issue changes, if enabled in settings](#)

Status: Assigned

Owner: bsiqueira

Cc:

Labels: Type-Defect Priority-Medium [Add a row](#)

[Submit Issue](#) [Discard](#)

Tip: Please search for an existing issue before reporting a problem as a new issue.

Remember: This report will be publicly visible. So, don't include passwords or other confidential information.

Ilustração 17 - Tela de cadastro de *Issues*

Depois de cadastrado, podem ser adicionados comentários e arquivos anexos no *issue*, registrando seu histórico.

A ferramenta disponibiliza um repositório para o código fonte do projeto. Cada atualização registrada no repositório pode receber comentários dos usuários e notas positivas ou negativas acerca da qualidade da alteração.

Outro recurso disponibilizado pelo *Google Code* é o *wiki*. Os usuários podem criar um *wiki* para organizar a documentação do projeto utilizando marcação específica para formatar o texto.

A Ilustração 18 apresenta a administração dos dados do projeto. A ferramenta possui duas categorias para os status de *issues*: abertas ou fechadas. É possível criar vários status diferentes para suas tarefas, mas não é possível criar um workflow para as tarefas existentes. Não existem restrições com relação às mudanças de estados e cada estado pode ser alterado para todos os outros.

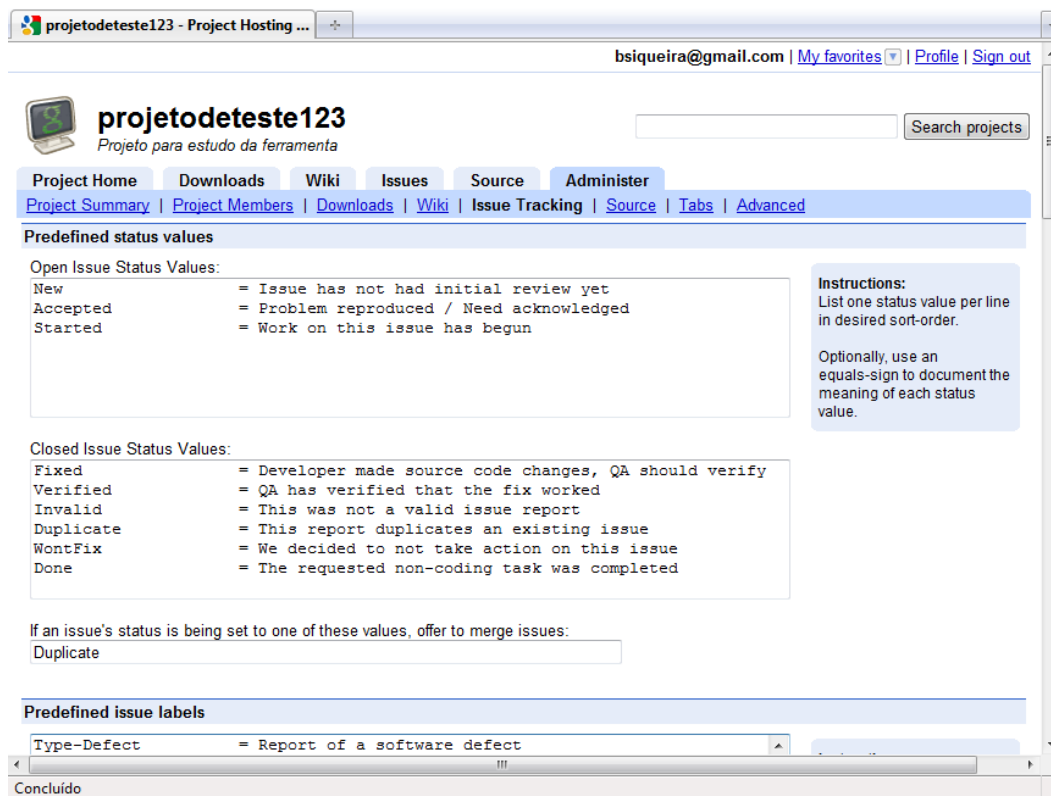


Ilustração 18 - Tela de Status de *Issues*

Como ferramenta de gerência de projeto, o *Google Code* é bastante completo, permitindo com facilidade o cadastro de *issues* e de documentação. Também provê um ambiente simples para o armazenamento e controle de todos os artefatos gerados ao longo de desenvolvimento.

A ferramenta não disponibiliza a organização dos *issues* em iterações ou algum organizador um nível acima para melhor controle em projetos de longo prazo. Também não permite cadastrar tipos novos de *issues*, limitando-os apenas a defeitos reportados por usuário, por desenvolvedor e revisão de funcionalidade. Embora permita algumas personalizações, não permite extensões e uso em outros ambientes que não o Google. Também não permite o desenvolvimento de projetos que não são *open source*, pois o código-fonte pode ser acessado e visualizado por todos.

3.4.

Bugzilla

Bugzilla é uma aplicação web de controle de alterações. É flexível e se encaixa bem em ambientes com vários projetos.

É possível gerenciar vários produtos, agrupando projetos em “classificações”. Para um produto específico, é possível definir componentes.

Para auxiliar a gerência de projetos e garantia de qualidade, a ferramenta permite a definição de *milestones* (marcos) e versões de lançamento.

Um *bug* cadastrado no Bugzilla serve para caracterizar tanto defeitos quanto novas funcionalidades. Cada *bug* é associado a um módulo (produto e componente) e à versão em que está presente.

Bugzilla - Bug 3551 terrible mess Last modified: 2009-10-07 08:49:30 PDT

Home | New | Browse | Search | Reports | My Requests | Preferences | Help | Log out biquetira@gmail.com

Bug List: (2 of 97) First Last Prev Next Show last search results

Bug 3551 - terrible mess (edit) Save Changes

Status: NEW (edit)

Product: FoodReplicator

Component: Digestive Goo

Version: unspecified

Platform: PC Windows 2000

Importance: P1 blocker

Target Milestone: ---

Assigned To: gerv@mozilla.org

QA Contact: marialuigina@gmail.com

URL: http://jide-oss.dev.java.net/ (edit)

Whiteboard:

Keywords:

Depends on:

Blocks:

Show dependency tree / graph

Reported: 2006-03-17 06:31 PST by FarnugA@cpwplc.com

Modified: 2009-10-07 08:49 PDT (History)

CC List: Add me to CC list 1 user (edit)

See Also: Add Bug URLs:

Large text box:

Free text:

A multiple-select box: Option 1 Option 2

Drop Down List:

Date Time:

Flags:

another-flag

another-flag2

blocker

regression

test

Orig. Est.	Current Est.	Hours Worked	Hours Left	%Complete	Gain	Deadline
0.0	0.0	0.0 + 0	0.0	0	0.0	(YYYY-MM-DD)

Ilustração 19 - Um *bug* exibido em um formulário Bugzilla

A Ilustração 20 apresenta o workflow do ciclo de vida de um *bug*. Embora seja bastante completo, este ciclo de vida não pode ser alterado para diferentes projetos ou diferentes tipos de *bugs*.

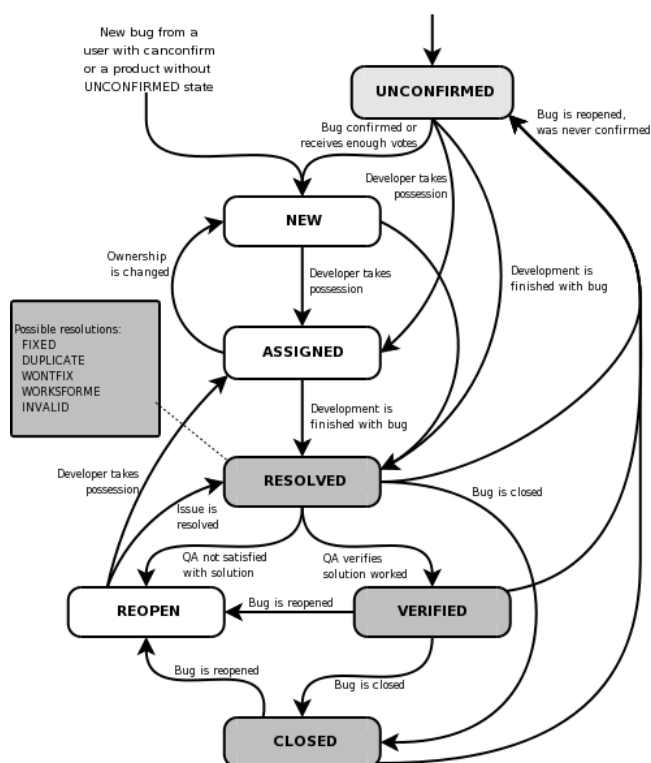


Ilustração 20 - Ciclo de vida de um *bug* no Bugzilla

Embora as funcionalidades da ferramenta sejam extensas, as operações principais oferecidas ao usuário são simples e diretamente ligadas às atividades de manutenção: a criação de um novo *bug*, sua triagem, revisão, escalonamento, discussão e fechamento e a obtenção de relatórios e consultas baseados em propriedades do *bug*. Para suportar estas operações, a ferramenta possui uma série de características particulares, que incluem:

- Contas de usuário: cada usuário possui uma conta própria pela qual é identificado, e que lhe permite acessar e alterar os atributos dos bugs.
- Cadastro de bugs: todo bug possui um conjunto completo de informações, incluindo uma descrição, estado atual, prioridade, versão, plataforma, e pessoas envolvidas.
- Comentários: cada bug armazena consigo uma lista de comentários e é possível acompanhar seu histórico.
- Anexos: podem ser anexados um ou mais arquivos a cada bug.
- Interface de consulta: é fornecida uma interface que permite buscar por bugs através das suas características (Ilustração 21). É possível, inclusive, efetuar alterações simultâneas em múltiplos bugs com base nos resultados desta busca.

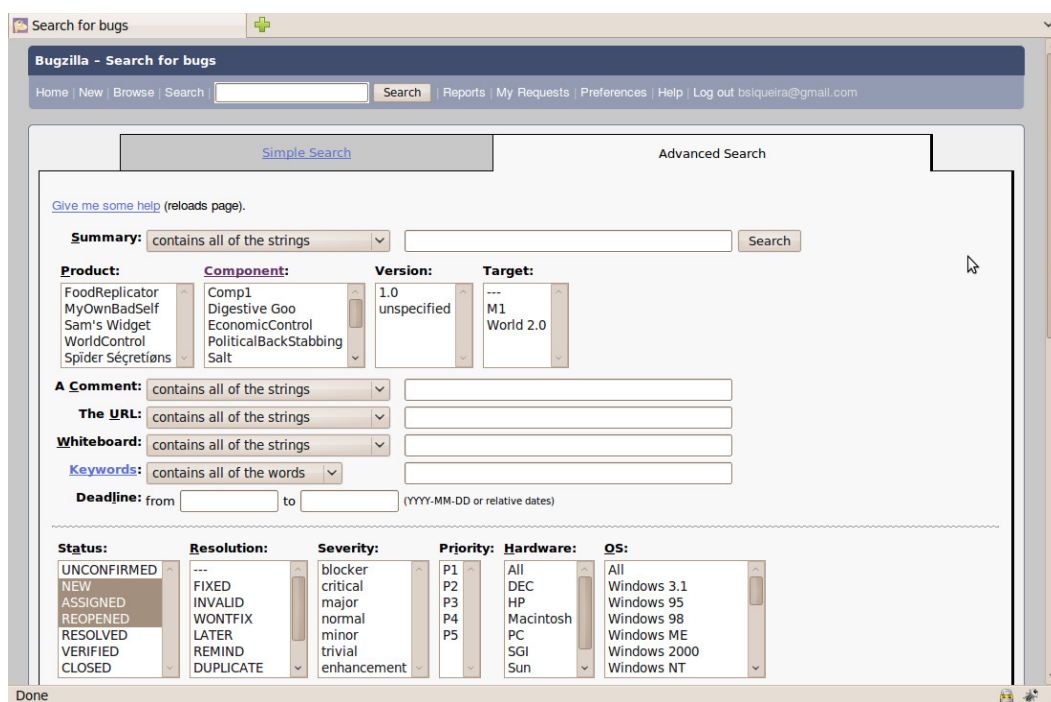


Ilustração 21 - Interface de busca do Bugzilla

- Integração com correio eletrônico: cada *bug* reportado e alteração efetuada podem ser enviados por email para um ou mais recipientes cadastrados.

Bugzilla é uma solução comprovada que apóia grandes projetos e bases de usuários. As funcionalidades de seu workflow são mais do que o suficiente para a maioria das organizações. Por outro lado, *Bugzilla* é particularmente complicado de instalar e gerenciar. E a interface do usuário não possui boa usabilidade.

3.5.Conclusão

Após análise das ferramentas acima, levando em consideração funcionalidades já implementadas, flexibilidade, tipo de licença de uso e se é facilmente extensível, foi selecionada a ferramenta *Redmine* como base para o desenvolvimento do workflow.

Um estudo mais aprofundado das ferramentas *Google Code* e *BugZilla* foi dispensado, pois alguns dos requisitos necessários para o projeto não eram atendidos pelas duas ferramentas e foram identificados com antecedência. Como a falta de simplicidade do *Bugzilla* e a impossibilidade de gerenciar projetos que não sejam open-source no *Google Code*.

Para desenvolver o workflow, é possível tirar proveito não só do *Redmine*, como do framework *Ruby on Rails* e de sua arquitetura simples e flexível. Maiores informações sobre a implementação estão na seção 4.