

## **Comparação de Ferramentas de Controle e Gestão de Bugs: Redmine, GitHub Issues, e Bugzilla**

Equipe: Antonio Favarin, Felipe Beppler, Filipe Richter, Kelvin Katsuo e Pedro Henrique Dias

### **1. Visão Geral das Ferramentas**

#### ***1.1 Redmine***

Redmine é um software livre, gerenciador de projetos baseados na web e ferramenta de gerenciamento de bugs.

#### ***1.2 GitHub Issues***

GitHub Issues é o sistema nativo de rastreamento de bugs e gerenciamento de tarefas da plataforma GitHub.

#### ***1.3 Bugzilla***

Bugzilla é uma ferramenta baseada em Web e e-mail que dá suporte ao desenvolvimento do projeto Mozilla, rastreando defeitos e servindo também como plataforma para pedidos de recursos.

## 2. Uso das Ferramentas em Projetos Open Source

Para realizar uma avaliação completa das ferramentas selecionadas (**Redmine**, **GitHub Issues**, e **Bugzilla**), é necessário explorar como cada uma é utilizada em projetos de software open source.

### 2.1 Redmine: Projeto Open Source Avaliado – ChiliProject

- **Descrição:** O **ChiliProject** é um fork do Redmine e também usa Redmine para seu próprio gerenciamento de bugs e tarefas. A ferramenta oferece suporte ao gerenciamento de múltiplos projetos, cada um com seus próprios bugs e tarefas.
- **Gerenciamento de Bugs e Tarefas:** A criação de bugs é bastante detalhada, com campos personalizados para adicionar descrições completas, como passos para reproduzir o problema e opções para definir prioridade e severidade. As tarefas podem ser atribuídas a diferentes usuários e ter status atualizado conforme o progresso.
- **Características Específicas:** O Redmine permite anexar arquivos e comentários em tickets, além de criar relatórios e gráficos baseados no progresso das tarefas. A flexibilidade é o ponto forte, possibilitando adaptar as funcionalidades às necessidades específicas de cada projeto.

### 2.2 GitHub Issues: Projeto Open Source Avaliado – Vue.js

- **Descrição:** O projeto **Vue.js** utiliza **GitHub Issues** para gerenciar suas tarefas e bugs. A simplicidade e integração nativa com o código facilitam o acompanhamento de bugs diretamente na plataforma GitHub.
- **Gerenciamento de Bugs e Tarefas:** Criar issues no GitHub é direto, com campos para descrição, passos para reproduzir e até links para o código associado. Labels e milestones ajudam a organizar as issues por prioridade, tipo de bug e etapa de desenvolvimento.
- **Características Específicas:** Um grande diferencial do GitHub Issues é sua integração com **GitHub Actions** e automações, permitindo que atualizações sejam automaticamente vinculadas a eventos de commit ou pull requests, além de permitir a adição de comentários e anexos diretamente na interface.

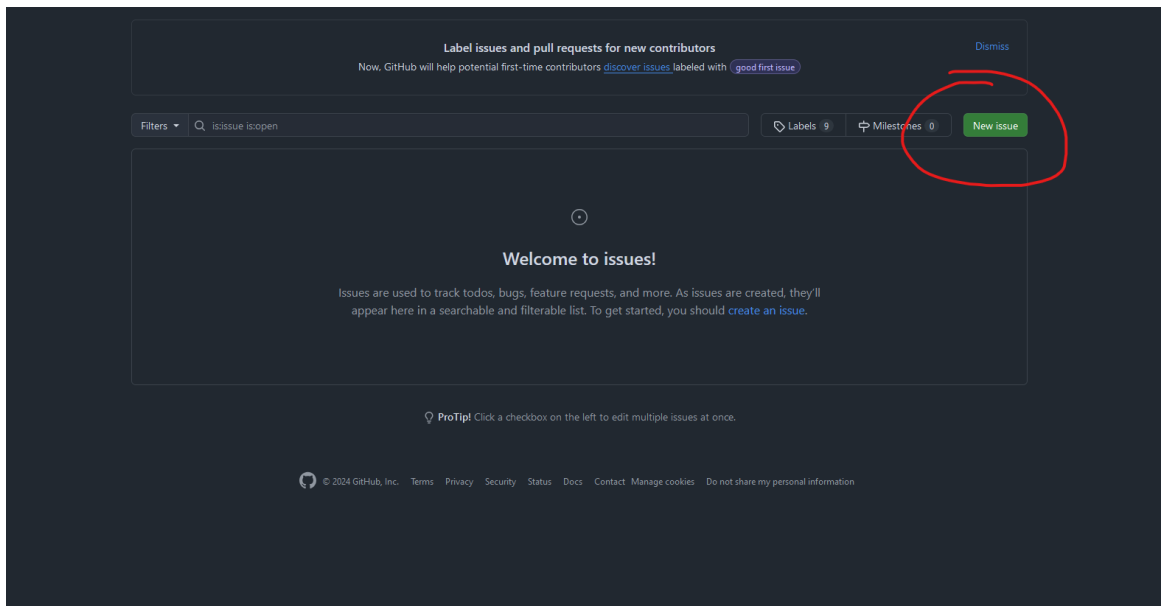
### **2.3 Bugzilla: Projeto Open Source Avaliado – Mozilla Firefox**


- **Descrição:** O navegador **Mozilla Firefox** é um dos principais projetos que utilizam o **Bugzilla** para rastreamento de bugs e gerenciamento de tarefas. A interface, embora mais técnica, oferece uma riqueza de opções para detalhamento e personalização de tickets.
- **Gerenciamento de Bugs e Tarefas:** A criação de tickets no Bugzilla envolve preencher campos detalhados como produto afetado, versão, sistema operacional, severidade e prioridade. É possível anexar arquivos e adicionar comentários relevantes para o desenvolvimento.
- **Características Específicas:** O Bugzilla oferece campos avançados para controle de bugs em grande escala, como dependências entre bugs e relatórios detalhados que ajudam a organizar e priorizar o trabalho.

## **3. Execução de Casos de Uso Comuns**

### **3.1 Criação de um Bug/Ticket**

- **Redmine:** O processo de criação de bugs no Redmine permite adicionar informações detalhadas, como descrição, passos para reproduzir o erro, responsável pela tarefa e prioridade. Além disso, é possível adicionar campos personalizados, o que é útil para diferentes tipos de projetos.
- **GitHub Issues:** A criação de uma issue no GitHub é simples. O usuário pode fornecer uma descrição, anexar arquivos relevantes, e adicionar labels e milestones para categorizar a issue. A simplicidade da interface torna o processo rápido e acessível.



 **Add a title**

Titulo do Issue

**Add a description**

Write | Preview

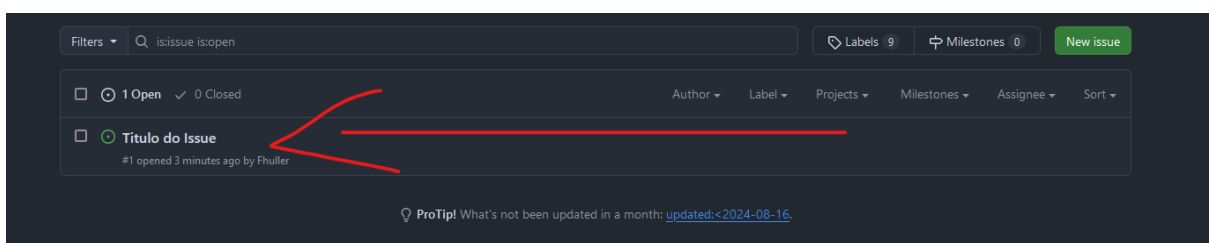
Descrição do issue

Markdown is supported | Paste, drop, or click to add files

**Submit new issue**

Remember, contributions to this repository should follow our [GitHub Community Guidelines](#).

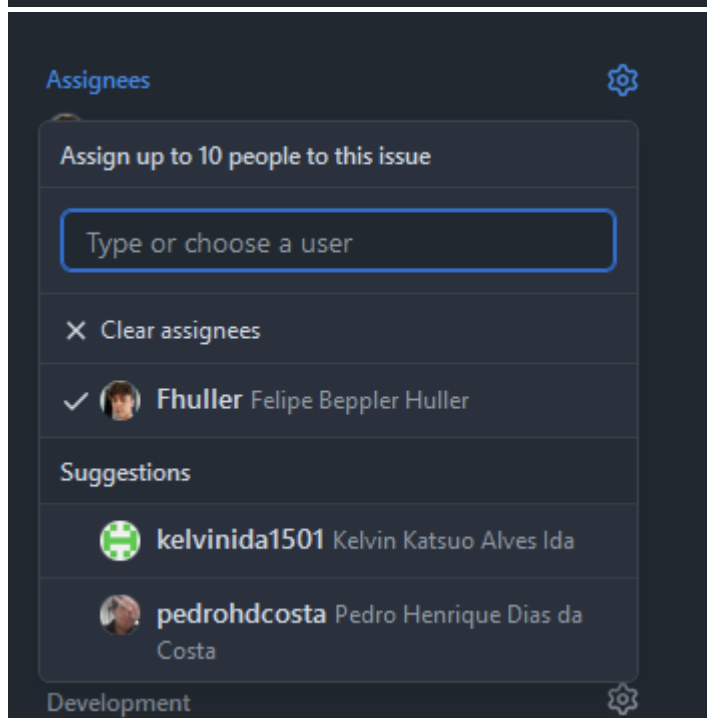
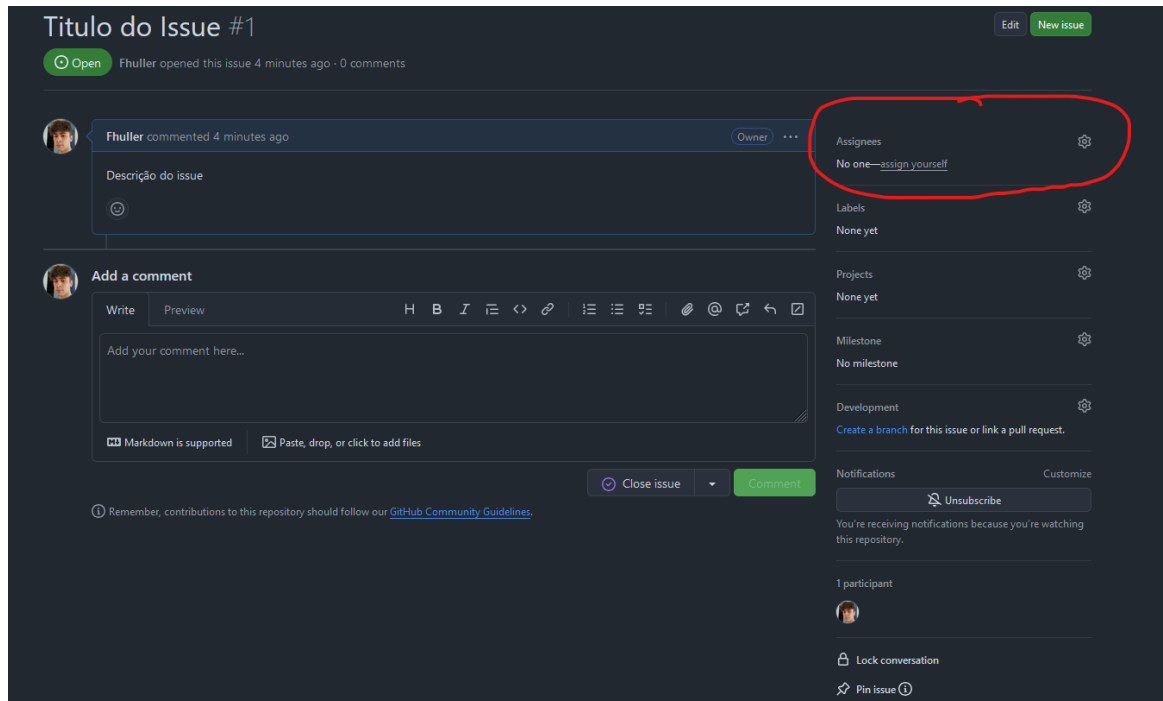
Após criar a issue irá aparecer para os outros usuários



- **Bugzilla:** A criação de tickets no Bugzilla é bem detalhada, com campos específicos para informações como gravidade, componente afetado e dependências entre bugs. Há uma clara separação entre campos obrigatórios e opcionais, o que garante que o ticket seja o mais completo possível.

### 3.2 Atribuição de Bug/Ticket

- **Redmine:** Permite atribuir bugs a desenvolvedores específicos e até criar grupos de usuários que podem compartilhar a responsabilidade por um bug. A interface de atribuição é flexível e permite mudar o responsável à medida que o status do bug evolui.
- **GitHub Issues:** No GitHub, a atribuição de issues é feita diretamente por meio da interface, onde um ou mais desenvolvedores podem ser associados à issue. O uso de labels ajuda a identificar quem está responsável por determinadas categorias de bugs.



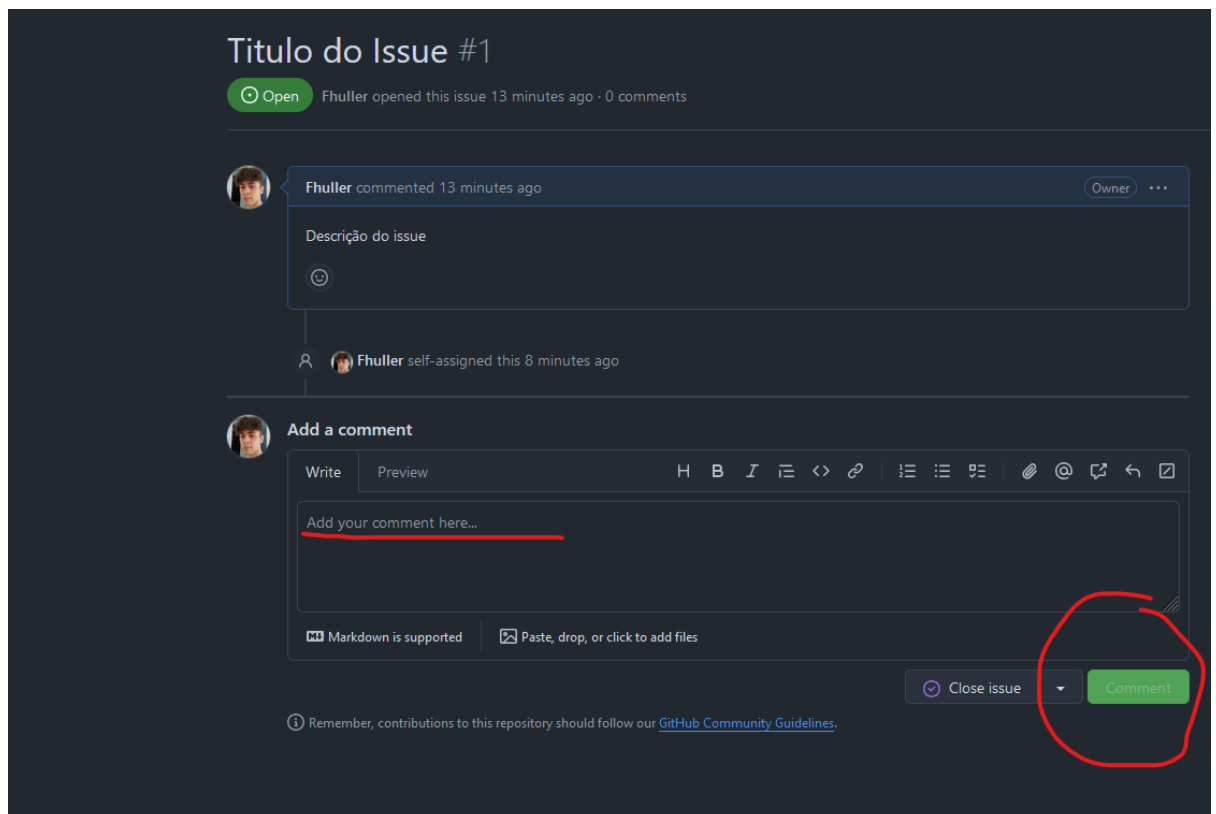
- **Bugzilla:** O Bugzilla também facilita a atribuição de bugs a desenvolvedores específicos. A ferramenta permite que os bugs sejam transferidos entre desenvolvedores conforme necessário, e permite a inclusão de um CC para que outros membros da equipe acompanhem as atualizações.

### 3.3 Atualização de Status

- **Redmine:** O status dos bugs pode ser atualizado manualmente conforme o progresso (por exemplo, de "Novo" para "Em andamento" ou "Fechado"). O Redmine permite a criação de status personalizados e fluxos de trabalho, permitindo que a equipe ajuste o processo conforme necessário.
- **GitHub Issues:** No GitHub Issues, o status de uma issue é alterado automaticamente quando o desenvolvedor fecha a issue ou vincula a uma pull request. O status também pode ser atualizado manualmente para refletir o andamento.
- **Bugzilla:** O Bugzilla tem um sistema robusto de estados de bugs (como "Novo", "Atribuído", "Resolvido", "Verificado"). Ele também permite atualizações detalhadas de status, especialmente em projetos maiores, onde cada mudança precisa ser registrada.

### 3.4 Adição de Comentários e Anexos

- **Redmine:** Comentários podem ser adicionados diretamente nos bugs, e arquivos relevantes (como logs ou capturas de tela) podem ser anexados. O histórico completo das interações e anexos é mantido.
- **GitHub Issues:** Comentários são uma parte essencial do fluxo de trabalho do GitHub. Eles permitem discussões em tempo real e são utilizados para fornecer feedback. Anexos podem ser incluídos, e há suporte para markdown, facilitando a formatação.



- **Bugzilla:** O Bugzilla permite que os desenvolvedores e usuários adicionem comentários detalhados e anexem arquivos aos tickets. Isso ajuda a manter um registro completo de todas as discussões e alterações relacionadas a um bug.

### 3.5 Criação de Relatórios e Métricas

- **Redmine:** O Redmine oferece relatórios avançados, incluindo gráficos de burndown e relatórios de progresso. É possível personalizar esses relatórios com base em diversos critérios, como tempo gasto, estado atual dos bugs e responsabilidades.
- **GitHub Issues:** O GitHub oferece métricas básicas de progresso através dos **GitHub Projects**, mas para relatórios mais detalhados, é necessário integrar com ferramentas de terceiros, como **ZenHub** ou **Waffle.io**.
- **Bugzilla:** O Bugzilla possui um sistema avançado de relatórios, permitindo a criação de consultas detalhadas para acompanhar bugs por status, prioridade, produto, e outros critérios. Esses relatórios podem ser exportados e visualizados em diferentes formatos.

## 4. Tabela Comparativa

Critério	Redmine	GitHub Issues	Bugzilla
Interface de Usuário	Flexível, porém mais técnica	Simples e intuitiva	Um pouco datada, mas funcional

<b>Funcionalidades</b>	Gestão de múltiplos projetos, relatórios avançados	Labels, milestones, integração Git	Campos avançados, workflows detalhados
<b>Customização</b>	Alta (campos personalizados e plugins)	Baixa (estruturado de forma simples)	Alta (configuração detalhada de workflows)
<b>Escalabilidade e Desempenho</b>	Boa para projetos grandes	Boa para projetos de pequeno a médio porte	Excelente para projetos grandes
<b>Relatórios e Métricas</b>	Relatórios detalhados e gráficos	Relatórios básicos via GitHub Projects	Relatórios customizáveis e detalhados
<b>Integração com Ferramentas</b>	Integra com Git, SVN e outras	Integra com GitHub Actions, CI/CD	Integra com várias ferramentas, porém exige configuração
<b>Comunidade e Suporte</b>	Ativa, com plugins disponíveis	Enorme, com vasta documentação	Comunidade consolidada, ampla documentação
<b>Custo e Implementação</b>	Gratuito, mas pode exigir instalação local	Gratuito, baseado na plataforma GitHub	Gratuito, mas requer instalação e manutenção
<b>Segurança e Controle de Acesso</b>	Permissões granulares	Permissões básicas	Controle de acesso detalhado

## 5. Análise Detalhada

### 5.1 Interface de Usuário e Usabilidade

- **Redmine** oferece uma interface flexível, porém mais técnica, o que pode exigir mais tempo de adaptação por parte dos usuários. Em compensação, permite a personalização de fluxos de trabalho e campos de acordo com as necessidades do projeto.
- **GitHub Issues** é muito simples e direto, com uma interface moderna e intuitiva, especialmente útil para quem já trabalha com GitHub. Suas funcionalidades de labels e milestones tornam o gerenciamento de tarefas fluido.
- **Bugzilla** possui uma interface mais antiquada, mas ainda assim funcional. Sua complexidade pode ser uma barreira inicial, mas para projetos grandes, essa abordagem mais técnica é vantajosa.

### 5.2 Funcionalidades e Customização

- **Redmine** se destaca pela sua capacidade de gerenciar múltiplos projetos e pela extensa personalização, seja com campos personalizados ou com a adição de plugins. Isso permite adaptá-lo a diferentes cenários de projetos, sejam eles pequenos ou grandes.



- **GitHub Issues**, embora mais simples, é extremamente eficaz para projetos de pequeno a médio porte. A falta de campos personalizados é compensada pela simplicidade e integração com GitHub Actions.
- **Bugzilla** tem uma das implementações mais avançadas de workflows de bugs. Ele permite a criação de campos personalizados e regras detalhadas, o que o torna ideal para grandes equipes e projetos complexos.

### 5.3 Escalabilidade e Desempenho

- **Redmine** se comporta bem em projetos de grande escala, com a vantagem de suportar múltiplos projetos e usuários simultaneamente. Seu desempenho depende da configuração do servidor, mas oferece estabilidade.
- **GitHub Issues** é ideal para projetos menores, com menos volume de dados, devido à sua estrutura simples e integração nativa com GitHub. A escalabilidade pode ser limitada em projetos de grande porte.
- **Bugzilla** foi projetado para lidar com grandes volumes de bugs e tarefas, sendo utilizado em projetos massivos como o Mozilla Firefox. Sua capacidade de lidar com muitos usuários e dados o torna a escolha certa para grandes organizações.

### 5.4 Relatórios e Métricas

- **Redmine** oferece funcionalidades avançadas de relatórios com gráficos e métricas detalhadas, tornando-o uma excelente escolha para quem precisa de uma visão clara do andamento do projeto.
- **GitHub Issues** possui opções básicas de relatório, que podem ser estendidas através de GitHub Projects, mas carece da profundidade que outras ferramentas oferecem.
- **Bugzilla** permite a criação de relatórios extremamente detalhados e customizáveis, com opções avançadas de filtragem de dados.

### 5.5 Integração com Ferramentas

- **Redmine** possui integrações nativas com sistemas de controle de versão, como Git e SVN, e pode ser expandido com plugins para CI/CD e outras ferramentas.
- **GitHub Issues** se integra diretamente com GitHub Actions, oferecendo automação fácil para pipelines de CI/CD.
- **Bugzilla** também integra bem com várias ferramentas, porém a configuração dessas integrações pode exigir mais esforço e conhecimento técnico.

## 6. Recomendações

- Para projetos que exigem **muita personalização**, **Redmine** é a escolha ideal. Ele oferece uma ampla gama de funcionalidades e pode ser adaptado a diversos cenários, desde pequenas startups até grandes organizações.

- **GitHub Issues** é recomendado para **projetos menores ou de médio porte** que já utilizam o GitHub como repositório. Sua interface simples e a integração com o ecossistema GitHub o tornam perfeito para equipes ágeis e menores.
- Para **projetos grandes e complexos**, **Bugzilla** é a melhor escolha. Sua capacidade de lidar com grandes volumes de dados e sua flexibilidade na criação de workflows avançados são indispensáveis em grandes organizações.

## 7. Conclusão

Com base na análise das três ferramentas, a escolha da melhor depende do contexto e do tipo de projeto. O **Redmine** se destaca pela sua flexibilidade e funcionalidades avançadas, sendo mais indicado para quem precisa de personalização e gestão de múltiplos projetos. O **GitHub Issues** é ideal para projetos de menor escala, com foco na simplicidade e integração rápida com o código. Já o **Bugzilla** é robusto e ideal para projetos de grande escala, especialmente aqueles que necessitam de rastreamento detalhado de bugs e controle refinado do fluxo de trabalho.