

## Estudo Autodidata Git e GitHub



Aula: 7

Tema: Issues, Forks e Saved Replies

## Issues no GitHub

O Issues é um sistema de rastreamento de tarefas, bugs e sugestões dentro de um repositório no GitHub.

Para que servem as Issues?

- ➔ Relatar bugs: Se o código está com erro, você pode abrir uma Issue para descrever o problema.
- ➔ Sugerir melhorias: Quer adicionar uma funcionalidade? Abra uma Issue e discuta com a equipe.
- ➔ Organizar tarefas: Times usam Issues para dividir o trabalho e acompanhar o progresso.
- ➔ Fazer perguntas: Pode ser usado para tirar dúvidas sobre o projeto.

## Componentes de uma Issue:

Título → Um resumo curto do problema.

Descrição → Explicação detalhada, pode incluir imagens, trechos de código e links.

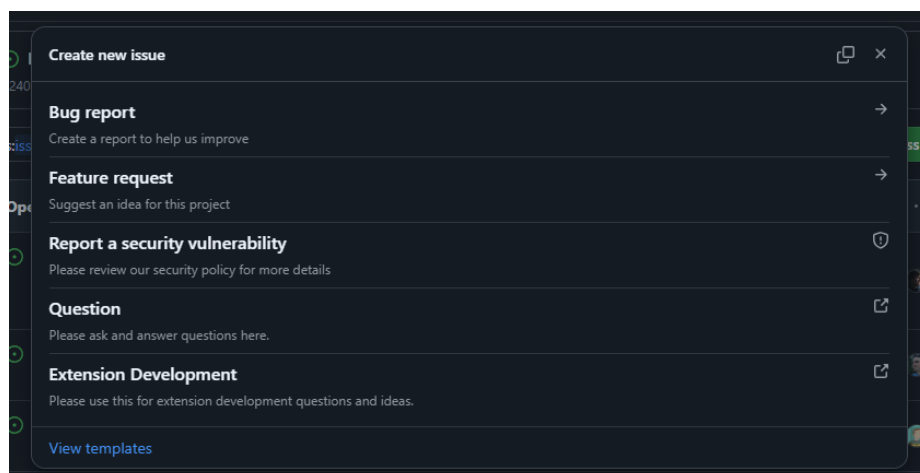
Labels → Categorias como "bug", "enhancement" (melhoria), "question", etc, classificações gerais do problema.

Assignees → Pessoas responsáveis por resolver a Issue.

Milestones → Agrupam Issues relacionadas dentro de um objetivo maior.

Comentários → Discussões entre os colaboradores.

Referências → É possível referenciar commits, pull requests e outras Issues.



## Como criar uma Issue?

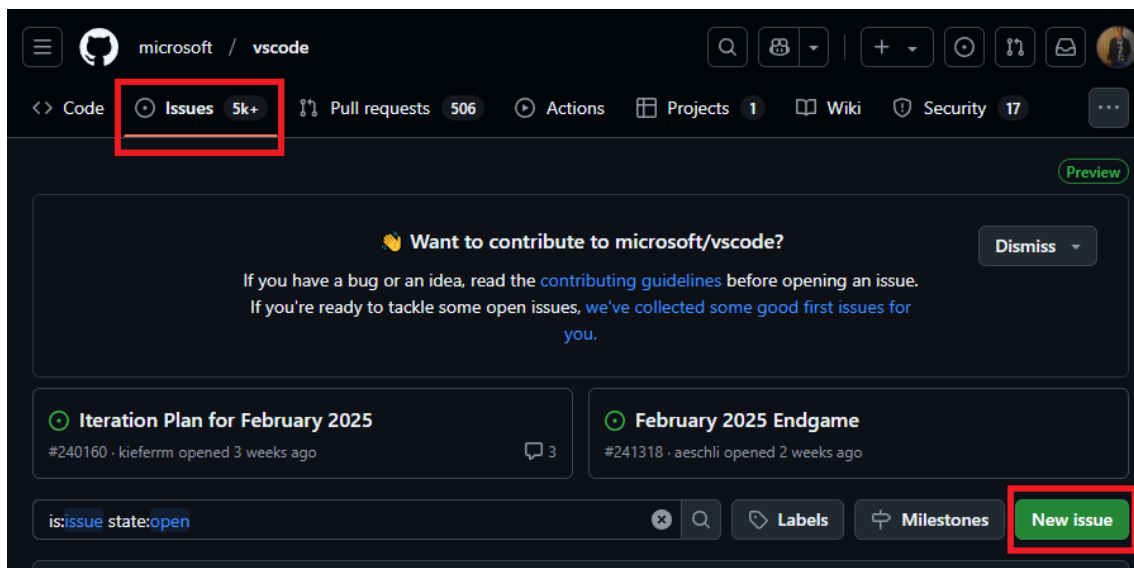
Vá até a aba Issues do repositório.

Clique em **New Issue**.

Preencha o título e a descrição.

Adicione Labels, Assignees e outros detalhes (se necessário para descrever melhor).

Clique em **Submit new issue**.



## Fechando uma Issue

Manualmente clicando em Close issue.

Automaticamente, ao usar palavras-chave em um commit/pull request (exemplo: Fixes #10 ou Resolves #10).

## Forks

O Fork é uma cópia de um repositório que fica na sua conta. Ele permite que você faça mudanças sem afetar o projeto original.

### Passo 1: Criar um Fork

No GitHub.com, vá até o repositório que deseja fazer um Fork.

No canto superior direito, clique no botão Fork.

O repositório será copiado para sua conta no GitHub.

### Passo 2: Clonar o Fork no GitHub Desktop

Abra o GitHub Desktop.

Clique em File > Clone repository.

Vá até a aba URL e escolha seu repositório forkado.

Escolha a pasta onde deseja salvar e clique em Clone.

### Passo 3: Criar uma branch para suas mudanças

No GitHub Desktop, clique no menu suspenso no canto superior.

Clique em New Branch.

Dê um nome para sua branch (exemplo: nova-feature).

Clique em Create Branch.

### Passo 4: Fazer modificações no código

Agora você pode abrir os arquivos no seu editor de código (VS Code, por exemplo) e fazer as alterações que quiser.

Depois de editar, volte para o GitHub Desktop para poder fazer o push e mandar as edições do repositório local para o repositório remoto.

### Passo 5: Fazer commit das mudanças

No GitHub Desktop, vá até a aba Changes.

Verifique os arquivos que foram modificados.

Escreva uma mensagem curta no campo Summary (exemplo: "Adicionei um botão novo").

Clique no botão Commit to [nome-da-branch].

**Passo 6: Enviar as mudanças para o GitHub (Push)**

No GitHub Desktop, clique em Push origin (canto superior direito).

Isso enviará suas mudanças para o seu Fork no GitHub.

**Passo 7: Criar um Pull Request (PR)**

Agora que suas mudanças estão no GitHub, você pode pedir para o dono do repositório original revisar e, se aprovar, aceitar suas mudanças.

Acesse o GitHub.com e vá até seu Fork.

Você verá uma mensagem sugerindo abrir um Pull Request.

Clique em Compare & pull request.

Adicione um comentário explicando suas mudanças.

Clique em Create pull request.

Agora é só esperar a revisão!

**Passo 8: Atualizar seu Fork quando o repositório original for atualizado**

Se o repositório original recebeu novas mudanças, você pode atualizar seu Fork pelo GitHub Desktop.

**Resumo Rápido**

- ✓ Forks permitem copiar um repositório para sua conta e modificá-lo sem afetar o original.
- ✓ GitHub Desktop facilita clonar, criar branch, fazer commits e enviar mudanças.
- ✓ Issues são criadas diretamente no site do GitHub para rastrear bugs e melhorias.
- ✓ Pull Requests permitem enviar suas mudanças para o projeto original.

**Diferença entre Fork e Clone**

Fork	Clone
Copia o repositório para sua conta no GitHub	Copia o repositório para sua máquina
Útil para contribuir em projetos open source	Útil para trabalhar localmente
Mantém um link com o repositório original	Não mantém esse link automaticamente

## Saved Reply

GitHub Issues, o Saved Reply (ou Resposta Salva) é um recurso que **permite salvar mensagens padrão para reutilização rápida ao responder a Issues e Pull Requests.**

Para que serve o Saved Reply?

Ele é útil para times e desenvolvedores que respondem frequentemente a perguntas ou comentários repetitivos. Com isso, você pode:

- ✓ Economizar tempo, evitando escrever a mesma resposta várias vezes.
- ✓ Padronizar respostas, garantindo consistência na comunicação.
- ✓ Melhorar a organização, usando mensagens prontas para diferentes situações.

Como criar um Saved Reply no GitHub?

1. Vá para uma Issue ou Pull Request no GitHub.
2. No campo de comentário, clique no ícone de três pontinhos `...` no canto inferior direito.
3. Selecione Saved replies.
4. Clique em New saved reply.
5. Dê um nome para identificar essa resposta.
6. Escreva a mensagem que deseja salvar.
7. Clique em Save reply para salvar.

Como usar um Saved Reply?

1. Quando for responder a uma Issue ou Pull Request, clique novamente nos três pontinhos `...`.
2. Selecione Saved replies.
3. Escolha a resposta salva que deseja usar.
4. Edite se necessário e clique em Comment para enviar.

### Exemplo de uso de Saved Reply

🔗 Situação: Você trabalha em um repositório open source e recebe muitas perguntas sobre como contribuir.

✉ Resposta salva:

- > Olá! Obrigado por seu interesse em contribuir. 🚀
- > Para começar, siga estes passos:
- > 1. Faça um Fork do repositório.
- > 2. Clone o projeto localmente.
- > 3. Crie uma nova branch para suas mudanças.
- > 4. Após editar, faça um Pull Request.
- > Mais detalhes no nosso guia de contribuição: [link para o guia].

Agora o melhor de tudo, parando pra pensar o GitHub pode ser visto como um tipo de rede social:

### **GitHub: A Rede Social dos Programadores**

Se você parar para pensar, o GitHub funciona como uma rede social para programadores, onde os desenvolvedores colaboram, discutem e contribuem para projetos de código aberto. Assim como redes sociais tradicionais, ele tem ferramentas que facilitam a interação entre os usuários. Vamos comparar!

#### **1 Issues + Saved Reply = Postagens e Comentários**

Nas redes sociais, quando alguém tem um problema ou dúvida, faz um post e outras pessoas comentam para ajudar. No GitHub, isso acontece com as Issues:

- ✓ Usuários criam Issues para relatar problemas, sugerir melhorias ou tirar dúvidas.
- ✓ Outros programadores comentam para oferecer soluções ou pedir mais detalhes.
- ✓ Saved Replies funcionam como respostas prontas (como aquelas mensagens automáticas que alguns perfis usam para responder perguntas frequentes).

🔗 Exemplo prático:

- ◆ No Twitter/Facebook: "Meu aplicativo está travando, alguém sabe como resolver?"
- ◆ No GitHub: Issue aberta → "O botão de login não funciona no Chrome. Alguém mais está com esse problema?"

#### **2 Forks = Compartilhamento e Criação de Conteúdo**

No Instagram ou TikTok, você pode pegar uma tendência e criar sua própria versão. No GitHub, os Forks fazem algo parecido:

- ✓ Permitem que você copie um repositório para sua conta e faça modificações.
- ✓ Se encontrar uma solução para um problema, você pode abrir um Pull Request para sugerir a mudança ao dono original.
- ✓ Se quiser personalizar um projeto para seu próprio uso, o Fork também é a solução.



🔗 Exemplo prático:

- ◆ No TikTok/Instagram: Você vê um vídeo interessante, baixa o áudio e cria sua própria versão.
- ◆ No GitHub: Você vê um projeto legal, faz um Fork, melhora alguma coisa e compartilha sua versão.

### 3 Pull Requests = Marcação de Amigos e Colaborações

Quando alguém quer colaborar com um post ou projeto nas redes sociais, pode ser marcado ou adicionado. No GitHub, isso acontece via Pull Requests (PRs).

- ✓ Você faz um Fork, melhora um código e propõe a mudança ao repositório original.
- ✓ O dono do repositório analisa, dá feedback e decide se aceita ou não sua contribuição.
- ✓ É como se você marcasse um amigo numa publicação sugerindo que ele usasse algo que você criou.

🔗 Exemplo prático:

- ◆ No Facebook: "Ei @amigo, achei essa matéria interessante, dá uma olhada!"
- ◆ No GitHub: "Fiz essa melhoria no código e abri um Pull Request para revisar e, se aprovado, integrar ao projeto!"

### Resumo: GitHub como Rede Social

GitHub	Rede Social
<b>Issues</b> → Reportam bugs e sugerem melhorias.	<b>Posts</b> → Compartilham problemas e pedem opiniões.
<b>Comentários nas Issues</b> → Usuários discutem soluções.	<b>Comentários nas postagens</b> → Pessoas ajudam e debatem.
<b>Saved Reply</b> → Respostas automáticas para perguntas frequentes.	Respostas automáticas de perfis nas redes sociais.
<b>Forks</b> → Copiar um repositório para personalizar ou melhorar.	Compartilhar/Remixar um post ou vídeo para fazer algo novo.
<b>Pull Requests</b> → Contribuir com melhorias para um projeto.	Marcar um amigo para colaborar em uma postagem.