

# Educação Profissional Paulista

Técnico em  
**Desenvolvimento  
de Sistemas**

# Fluxo de trabalho

## Trabalho em equipe com *branches* e *pull requests*

Aula 3

Código da aula: [SIS]C4U2S6A3

## Exposição



### Objetivo da aula

- Conhecer o **processo de integração das *branches* e dos *pull requests*** durante o fluxo de trabalho diário.



### Competências da unidade (técnicas e socioemocionais)

- Aplicar ***frameworks* de desenvolvimento ágeis**, utilizando tecnologias de **CI e CD** que trabalham para a segurança do ambiente funcional e entregas divididas em partes **que agregam valor ao negócio de forma rápida**.



### Recursos didáticos

- Recurso audiovisual para a exibição de vídeos e imagens;
- Acesso ao laboratório de informática e/ou à internet;
- Caderno para anotações.



### Duração da Aula

45 minutos.

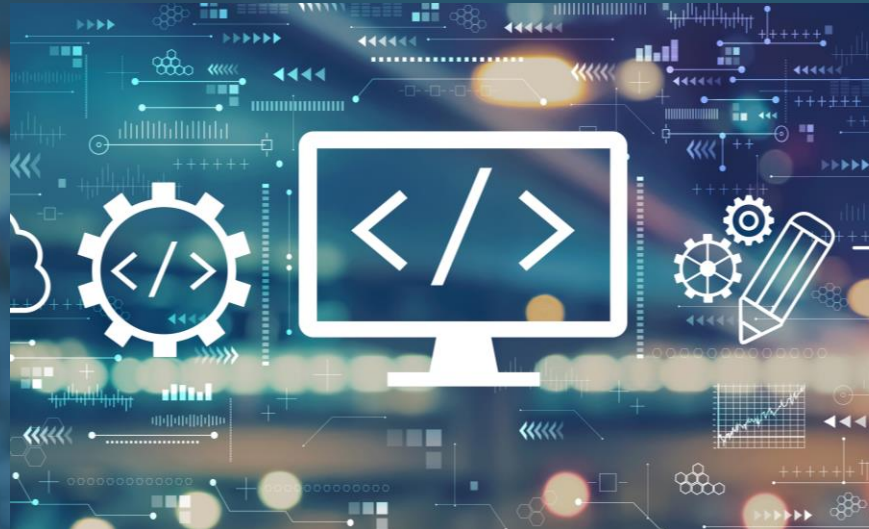
# Integrando *branches* e *pull requests* na rotina de trabalho

Integrar *branches* e *pull requests* nas rotinas diárias de desenvolvimento ajuda a manter a consistência, a qualidade e a colaboração ao longo do ciclo de vida do projeto.

- ✓ Compreender a **integração de *branches* e *pull requests*** em metodologias ágeis;
- ✓ Planejar estratégias para **minimizar e resolver conflitos de *merge***;
- ✓ Praticar o uso de **ferramentas de monitoramento dos repositórios**.



# Exposição



## Configurar pipelines CI/CD

Use ferramentas como Jenkins, Travis CI ou GitHub *actions* para configurar pipelines que automaticamente constroem, testam e implantam seu código a cada *pull request* ou *merge*.



## Testes automatizados

Escreva testes automatizados para validar as alterações de código. Isso garante que o novo código não quebre a funcionalidade existente.



## Revisões de código

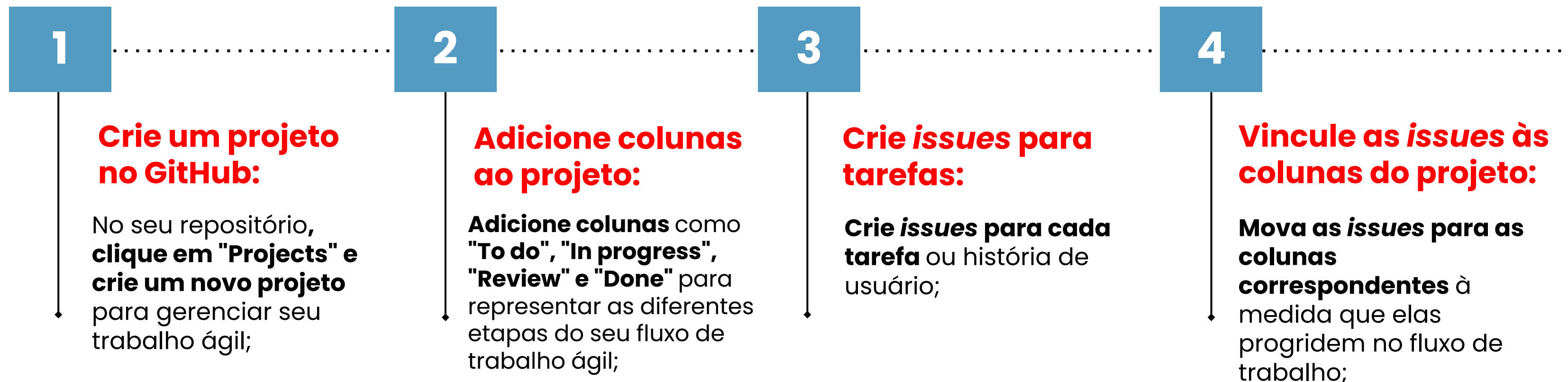
Implemente práticas de revisão de código dentro dos *pull requests*. Assegure-se de que pelo menos uma pessoa revise o código antes do *merge*.

**Fluxo de trabalho ágil:**  
como integrar *branches* e  
*pull requests* em  
metodologias ágeis

Passos práticos:

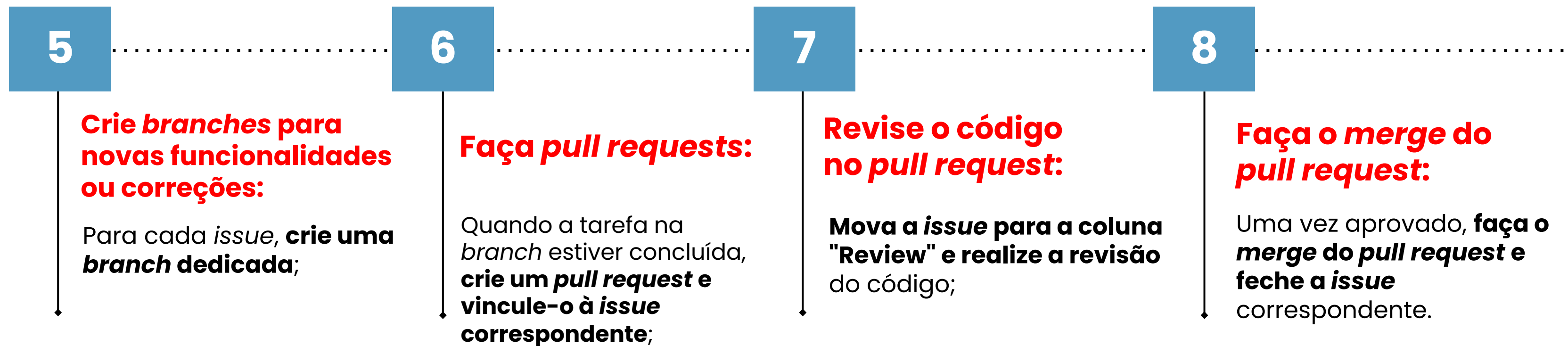
# Fluxo de trabalho ágil: como integrar *branches* e *pull requests* em metodologias ágeis

Para integrar **branches e pull requests** em um **fluxo de trabalho ágil no GitHub**, você pode **seguir estes passos**:





# Fluxo de trabalho ágil: como integrar *branches* e *pull requests* em metodologias ágeis



# Prevenção de conflitos: estratégias para minimizar e resolver conflitos de *merge*

**Conflitos de *merge*** podem ser problemáticos, mas **existem estratégias para minimizar sua ocorrência e resolver** quando ocorrem.

**Passos práticos:**

<b><i>Merge</i></b>	<b><i>Rebase</i></b>
Use o comando git merge para integrar alterações de outra <i>branch</i> .	Use git rebase para aplicar suas alterações em cima da <i>branch</i> principal atualizada.

**Resolução de conflitos:** quando conflitos ocorrem, o Git marca os arquivos afetados. **Abra esses arquivos, resolva os conflitos manualmente e depois faça *commit* das alterações.**



## Exposição



© Getty Images

# Prevenção de conflitos: estratégias para minimizar e resolver conflitos de *merge*

Para **prevenir conflitos e gerenciar *merges* no GitHub, você pode:**

### **Manter suas *branches* atualizadas**

Regularmente faça *merge* ou *rebase* da *branch* principal para a sua *branch* de *feature*.

### **Usar a função "Draft Pull Request"**

Quando começar a trabalhar em uma nova *branch*, crie um *draft pull request* para mostrar aos outros que você está trabalhando nessa parte do código.

### **Comunicar-se com a equipe**

Use a seção de comentários do *pull request* para manter a comunicação aberta com sua equipe sobre o progresso e os potenciais conflitos.

Elaborado especialmente para o curso.





Vamos conhecer na prática uma estratégia de *branching*?  
Assista ao vídeo “Pegando um *commit*” na plataforma Alura.



© Getty Images

Exposição



Acesso para link externo:  
ALURA. *Git e Github: estratégias de ramificação, conflitos e pull requests*. Pegando um commit. Disponível em: <https://cursos.alura.com.br/course/git-github-branching-conflitos-pull-requests/task/57927>. Acesso em: 5 fev. 2024.

Vamos  
fazer uma  
**atividade**

# Criação dos repositórios no GitHub

A *engineering manager* (EM) do seu *squad* solicitou **a criação de um fluxograma simples sobre a criação de um repositório no GitHub e suas configurações principais**, a fim de estimar o esforço e planejar as ações técnicas do seu time.

A partir da demanda solicitada, **criem um fluxograma simples que demonstre todos os passos vistos para a criação e a configuração do ambiente.**

Vamos verificar as orientações para a elaboração dessa atividade no próximo slide.



Vamos  
fazer uma  
**atividade**

## Criação dos repositórios no GitHub

 10 min

 **Atividade em grupos**



**1** Planejamento inicial  
Determinar quais são os fluxos necessários para efetivar a criação de um repositório e as configurações principais.



**2** A partir do estudo e do planejamento inicial, colocar no papel toda a sequência do fluxo e suas ações principais para garantir a assertividade.



**3** Criação do fluxograma conforme o levantamento das etapas necessárias.



**4** Envio do fluxograma pelo AVA.





Vamos  
fazer uma  
**atividade**

## Sugestão-padrão de resposta

**Resultado esperado:** relatório e fluxograma para criação de repositório:

1

### Documento de planejamento inicial:

- **Objetivo:** determinar os **fluxos necessários para a criação e a configuração de um repositório**.
- **Conteúdo:**
  - **Levantamento de requisitos básicos** para o repositório;
  - Identificação das **configurações principais**;
  - **Análise de melhores práticas** para gerenciamento de repositórios;
  - **Planejamento das políticas de acesso** e colaboração.

2

### Sequência do fluxo e ações principais:

#### Descrição:

- Uma descrição detalhada do fluxo de trabalho, incluindo etapas como a **criação do repositório, configuração de *branches*, definição de políticas de *merge*, adição de colaboradores, configuração de arquivos como README e .gitignore e implementação de ferramentas de CI/CD**.
- Cada etapa é acompanhada por **ações específicas a serem realizadas**, garantindo a assertividade do processo.

Vamos  
fazer uma  
**atividade**

# Continuação da sugestão-padrão de resposta

## 3 Criação de fluxogramas:

### Visualização gráfica:

- Um **fluxograma** claramente delineado **que mapeia todas as etapas** identificadas no documento de planejamento;
- O **fluxograma** demonstra a **ordem lógica das etapas e como elas se interconectam**;
- **Inclui decisões-chave**, como a escolha de licenças, configurações de segurança e integrações de ferramentas.

## 4 Envio do fluxograma por meio do ambiente virtual de aprendizagem (AVA):

### Formato de envio:

- O **fluxograma**, juntamente com o relatório, **é enviado** digitalmente pela plataforma **AVA**;
- **Instruções para acessar e interpretar o fluxograma são fornecidas**;
- **Espaço para feedback e discussão** sobre o fluxograma e o planejamento é disponibilizado no AVA.



# Hoje desenvolvemos:

- 1 A aprendizagem de **características processuais e práticas para a integração das *branches* de *pull requests*.**
- 2 A identificação de **pontos de atenção necessários para a prevenção de conflitos e estratégias para minimizar problemas com *merges*.**
- 3 O conceito **de utilização de ferramentas e práticas para monitorar a saúde dos repositórios.**

O que nós  
**aprendemos  
hoje?**

© Getty Images





# Saiba mais

Hoje **os conceitos de IA podem ser aplicados em diversas realidades** quando o assunto é relacionado à tecnologia. Dessa forma, **que tal explorar um pouco o relacionamento entre IA e GitHub assistindo ao vídeo abaixo?**

"GitHub Copilot X: tudo que você precisa saber da extensão que usa o GPT-4", do canal CÓDIGO FONTE TV. Disponível em:

<https://www.youtube.com/watch?v=BsMmhozMN5I>. Acesso em: 5 fev. 2024.

# Referências da aula

ALURA. Git e Github: estratégias de ramificação, conflitos e pull requests. Pegando um commit. Disponível em: <https://cursos.alura.com.br/course/git-github-branching-conflitos-pull-requests/task/57927> Acesso em: 5 fev. 2024.

CÓDIGO FONTE TV. GitHub Copilot X: tudo que você precisa saber da extensão que usa o GPT-4. Disponível em: <https://www.youtube.com/watch?v=BsMmhozMN5I> Acesso em: 5 fev. 2024.

GITHUB DOCS. Página inicial. Disponível em: <https://docs.github.com/pt>. Acesso em: 5 fev. 2024.

Identidade visual: Imagens © Getty Images.



# Educação Profissional Paulista

Técnico em  
**Desenvolvimento  
de Sistemas**