Educação Profissional Paulista

Técnico em
Desenvolvimento de Sistemas

Secretaria da Educação

SÃO PAULO
GOVERNO DO ESTADO

Fluxo de trabalho

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

Aula 1

Código da aula: [SIS]C4U2S6A1

Exposição

Objetivo da aula

- Conhecer estratégias para o **gerenciamento eficaz de *branches* por meio do trabalho em equipe.**

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.

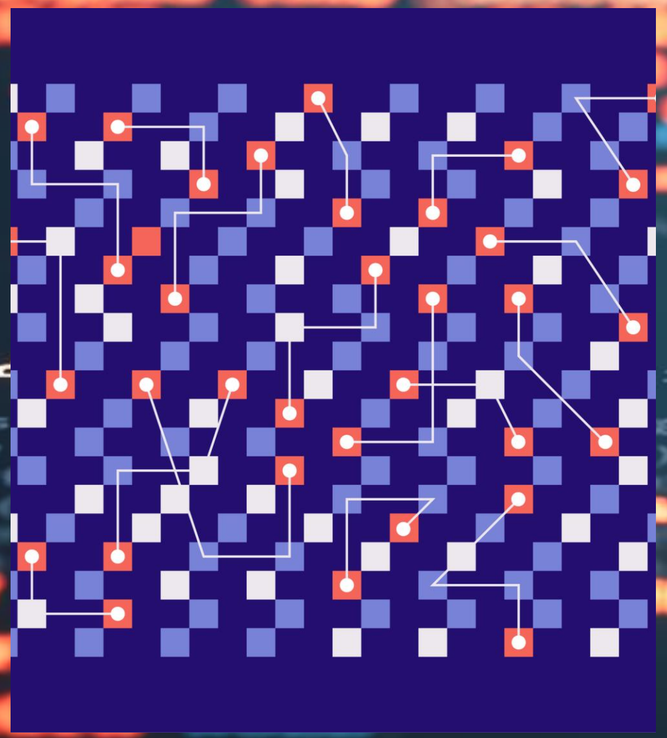
Exposição

Estratégias para o gerenciamento eficaz de *branches*

O gerenciamento eficaz de *branches* é crucial em um ambiente de desenvolvimento colaborativo para garantir que diferentes funcionalidades e correções possam ser trabalhadas simultaneamente sem interferir umas nas outras.

- **Criar acesso ao GitHub;**
- Compreender a **divisão de responsabilidades entre Git e GitHub**;
- Diferenciar **nomeação de organização**.

Exposição



© Getty Images

Recapitulando: Git

Git é um sistema de controle de versão distribuído, gratuito e de código aberto, projetado para lidar com tudo, desde projetos pequenos a muito grandes, com rapidez e eficiência.

O Git é notável por sua **flexibilidade, seu desempenho e sua segurança,** tornando-se uma escolha padrão para muitas equipes de desenvolvimento de software.

Exposição

Características principais do Git

Distribuído

Cada desenvolvedor tem uma **cópia completa do repositório**, incluindo todo o histórico de alterações. Isso torna as **operações mais rápidas**.

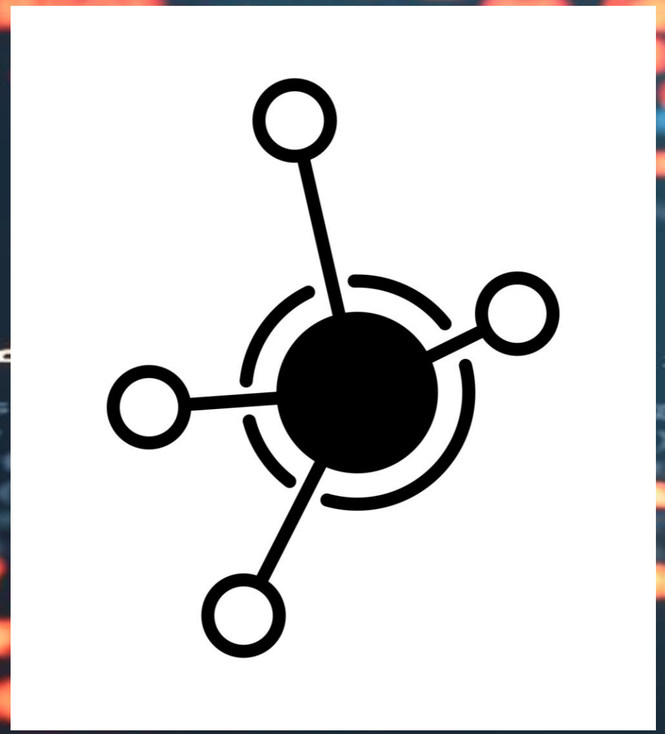
Controle de versão

Permite **rastrear alterações ao longo do tempo**, revertendo para estados anteriores quando necessário.

Branching e merging

Facilita o trabalho paralelo em diferentes recursos ou correções, **permitindo a fusão segura de alterações**.

Exposição



© Getty Images

Recapitulando: GitHub

GitHub é uma plataforma baseada na web para hospedagem de código usando o sistema de controle de versão Git.

Ele **fornece uma interface gráfica para o Git e ferramentas adicionais para a colaboração em equipe**, incluindo controle de acesso, gerenciamento de tarefas e rastreamento de problemas, além de recursos para automação de CI/CD (integração contínua e entrega contínua).

Exposição

Características principais do GitHub

Hospedagem de repositórios

Armazena repositórios Git on-line, proporcionando backup e fácil acesso.

Colaboração

Permite que equipes colaborem em projetos, com recursos como *pull requests*, revisão de código e discussões.

Integração contínua/delivery

Suporte para ferramentas de CI/CD, automatizando testes e implantação.

Exposição

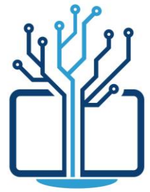
Fluxo típico de trabalho

Clone

Você **clona um repositório do GitHub para seu computador local**, criando uma cópia completa do projeto e seu histórico de versões.

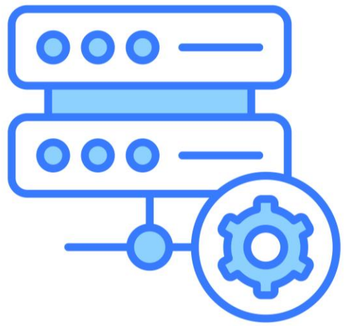
Branch

Você **cria *branches* para trabalhar em novos recursos ou correções** sem afetar o código principal.



Commit

Você **faz *commits* para salvar suas alterações localmente no Git.**



Push

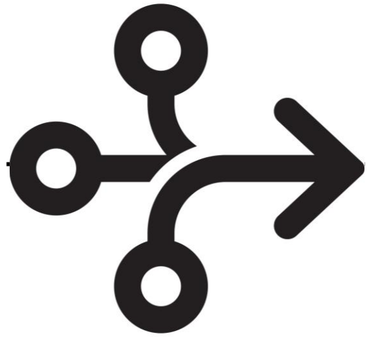
Você **envia suas alterações para o repositório no GitHub**, compartilhando seu trabalho com outros.

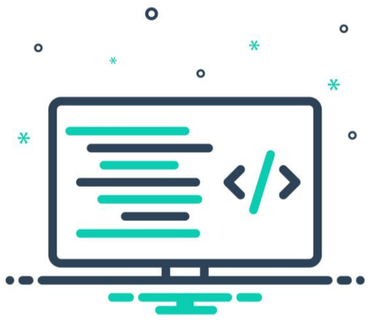
Exposição

Fluxo típico de trabalho

Merge

Após a revisão e a aprovação, suas alterações são mescladas no repositório no GitHub.





Pull request (PR)

Quando **suas alterações estão prontas**, você abre um PR no GitHub, solicitando que **suas alterações sejam revisadas e mescladas no código principal**.

Pull

Regularmente, você **puxa as alterações do repositório no GitHub para seu repositório local** a fim de manter seu código atualizado.

Exposição

Vamos realizar o cadastro no GitHub

Clique em “Sign up”

No canto superior direito da página, **clique em** “**Sign up**”.

Preencha o formulário de cadastro

Digite **seu e-mail**; **crie** um nome de **usuário**; escolha uma senha segura.

Verificação de e-mail

Acesse seu e-mail e **clique no link de verificação** enviado pelo GitHub.

Configurações adicionais e plano

<u>Atenção</u>: complete as configurações adicionais e **escolha o plano gratuito**.

Exposição

Vamos criar o primeiro repositório

Acesse sua página do GitHub.

Clique em “New repository”:

No canto superior esquerdo, ao lado do seu avatar, **clique no sinal de “+”** e **selecione “New repository”**.

Configure seu repositório:

- **Escolha um nome** para o seu repositório;
- **Adicione uma descrição** (opcional);
- **Selecione** se você quer que o repositório seja **público ou privado**;
- **Inicialize** o repositório com um **README** (opcional, mas recomendado);
- **Adicione** um .**gitignore** (se necessário);
- **Adicione uma licença** (se necessário).

Crie o repositório:

Clique em “**Create repository**”.

Exposição

© Getty Images

Vamos explorar um pouco mais o ambiente?

Settings gerais:

Options:	*Collaborators & teams*:	*Branches*:
Aqui, você pode **configurar aspectos básicos do repositório**, como seu **nome e sua descrição**, visibilidade (público ou privado) e **recursos como Wiki e Issues**. Você também pode **configurar o *branch* padrão** e os arquivos de README e .gitignore.	**Adicione colaboradores individuais ou equipes ao seu repositório**, definindo níveis específicos de acesso para cada um.	**Configure proteções de *branch*** para restringir quem pode fazer *push*, exigir revisões de código antes de mesclar e configurar status *checks* obrigatórios.

Elaborado especialmente para o curso.

Exposição

Funcionalidades de código e automação

1

Actions

Configure e gerencie GitHub *actions* para automação CI/CD e outros fluxos de trabalho.

2

Secrets & Variables

Armazene segredos e variáveis de ambiente que são usados em GitHub *actions*.

3

Webhooks

Configure *webhooks* para receber notificações HTTP sobre eventos específicos do repositório.

4

Deploy keys

Adicione chaves SSH que têm acesso de leitura ou gravação ao repositório.

Exposição

Funcionalidades de segurança e análise

Security & analysis:

Ative ou desative funcionalidades como Dependabot, que ajuda a identificar e corrigir vulnerabilidades de segurança nas dependências do seu projeto.

Code scanning alerts:

Configure ferramentas de análise de código para identificar possíveis problemas de segurança e erros de codificação.

Exposição

Que tal criarmos nossa primeira *branch*?

Localize e clique no seletor de *branch*:

No canto superior esquerdo, logo acima da lista de arquivos, você verá um **menu suspenso que mostra o nome da *branch* atual. Clique nele**.

Crie a nova *branch*:

- **Digite o nome da sua nova *branch*** no campo de texto.
- **Escolha a *branch*** a partir da qual você quer criar a nova (geralmente será a *branch* principal, como “main” ou “master”).
- **Clique em “Create branch: nome_da_nova_branch”.**

Importante

Agora, você criou uma nova ***branch*** e pode começar a fazer alterações nela, sem afetar a *branch* principal ou outras *branches*.

Isso é particularmente útil **quando você está trabalhando em uma nova funcionalidade ou correção**, pois **permite que você isole suas alterações até que estejam prontas para serem mescladas de volta ao código principal**.

Registro

Vamos fazer um ***quiz***

O que é Git?

Selecione a alternativa CORRETA:

- **Uma plataforma de hospedagem de código.**
- **Um sistema de controle de versão distribuído.**
- **Uma linguagem de programação.**
- **Um serviço de e-mail.**

Vamos fazer um ***quiz***

O que é Git?

✗ **Uma plataforma de hospedagem de código.**

RESPOSTA ERRADA! O GitHub é uma plataforma de hospedagem de código que utiliza o Git.

✓ **Um sistema de controle de versão distribuído.**

RESPOSTA CORRETA! O Git é um sistema de controle de versão distribuído que ajuda equipes de desenvolvimento a rastrear alterações no código ao longo do tempo.

✗ **Uma linguagem de programação.**

RESPOSTA ERRADA! Git não é uma linguagem de programação, mas uma ferramenta para gerenciar versões de código.

✗ **Um serviço de e-mail.**

RESPOSTA ERRADA! Git não tem relação com serviços de e-mail.

Registro

Vamos fazer um ***quiz***

O que você precisa fazer para criar uma nova *branch* e enviá-la para o repositório remoto usando a linha de comando?

Selecione a alternativa CORRETA:

- git branch nome_da_nova_branch seguido de git push origin nome_da_nova_branch
- git checkout nome_da_nova_branch
- git add nome_da_nova_branch
- git merge nome_da_nova_branch

Vamos fazer um ***quiz***

O que você precisa fazer para criar uma nova *branch* e enviá-la para o repositório remoto usando a linha de comando?

✓ **git branch nome_da_nova_branch seguido de git push origin nome_da_nova_branch**

RESPOSTA CORRETA! Primeiro você cria a *branch* localmente com git branch e, depois, a envia para o repositório remoto com git push.

✗ **git checkout nome_da_nova_branch**

RESPOSTA ERRADA! Esse comando só muda para uma branch existente; ele não cria uma nova.

✗ **git add nome_da_nova_branch**

RESPOSTA ERRADA! O comando git add é usado para adicionar alterações à área de *staging*, não para criar *branches*.

✗ **git merge nome_da_nova_branch**

RESPOSTA ERRADA! Git não tem relação com serviços de e-mail.

Registro

Vamos fazer um ***quiz***

Qual é o propósito de criar *branches* em um projeto de software?

Selecione a alternativa CORRETA:

- **Excluir versões antigas do código.**
- **Trabalhar em diferentes funcionalidades ou correções em paralelo com o código principal.**
- **Mudar o nome do repositório.**
- **Enviar e-mails para outros colaboradores.**

Vamos fazer um ***quiz***

Qual é o propósito de criar *branches* em um projeto de software?

- ✗ **Excluir versões antigas do código.**

  RESPOSTA ERRADA! *Branches* são usadas para desenvolver funcionalidades, correções ou experimentos em paralelo, não para excluir código.

- ✓ **Trabalhar em diferentes funcionalidades ou correções em paralelo com o código principal.**

  RESPOSTA CORRETA! *Branches* permitem que os desenvolvedores trabalhem em diferentes tarefas simultaneamente, sem interferir uns nos outros.

- ✗ **Mudar o nome do repositório.**

  RESPOSTA ERRADA! Muda-se o nome do repositório nas configurações do repositório, não por meio da criação de *branches*.

- ✗ **Enviar e-mails para outros colaboradores.**

  RESPOSTA ERRADA! Git e GitHub não têm funcionalidades de envio de e-mails para colaboradores por meio da criação de *branches*.

© Getty Images

O que nós **aprendemos hoje?**

Hoje desenvolvemos:

1. O conhecimento do **processo de configuração do GitHub para a criação de uma nova conta de acesso**.

2. A **configuração do ambiente** e a **exploração das ferramentas personalizáveis** dentro da plataforma.

3. A **criação da primeira *branch*** para versionamento dos arquivos de código.

Saiba mais

Você sabia que, **além de configurar o Git por meio do GitHub, também é possível instalar o software de versionamento diretamente em seu computador?**

“Como instalar e configurar Git / GitHub no Windows (passo a passo)”, do canal CÓDIGO FONTE TV. Disponível em: https://www.youtube.com/watch?v=mmcOw2ynWEs. Acesso em: 5 fev. 2024.

Esse tutorial explica o **processo de configuração da ferramenta em plataformas Windows**.

Referências da aula

BATAGINI, R. Como versionar usando o Git. **Medium**, 22 jul. 2020. Disponível em: https://medium.com/biblioteca-dos-devs/como-versionar-utilizando-o-git-1f5d8fe2afcd. Acesso em: 5 fev. 2024.

INFLUENTECH. **Como instalar e configurar Git/GitHub no Windows (passo a passo)**. Disponível em: https://www.youtube.com/watch?v=mmcOw2ynWEs. Acesso em: 5 fev. 2024.

MACHADO, B. Git e GitHub – Fundamentos. **Alura**, 1º dez. 2023. Disponível em: https://cursos.alura.com.br/meu-plano-de-estudos-brunamrs1-1679969205281-p564515. Acesso em: 5 fev. 2024.

Identidade visual: Imagens © Getty Images

Educação Profissional Paulista

Técnico em
Desenvolvimento de Sistemas

Secretaria da Educação

SÃO PAULO
GOVERNO DO ESTADO