

Educação Profissional Paulista

Técnico em
**Desenvolvimento
de Sistemas**

Fluxo de trabalho

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

Aula 2

Código da aula: [SIS]C4U2S6A2

Exposição



Objetivo da aula

- Compreender as **boas práticas de colaboração e comunicação efetiva por meio de *pull requests*** durante etapas de desenvolvimento.



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.

Maximizando a eficácia dos *pull requests*

Pull requests são uma **ferramenta essencial para revisão de código, discussão sobre alterações e garantia de qualidade** antes que o código seja integrado ao *branch* principal.

- ✓ Conhecer as **práticas de publicação de código**;
- ✓ Utilizar **técnicas de utilização dos comentários** para melhoria da qualidade;
- ✓ Identificar **automação e integração contínua**.

Exposição

Vamos praticar a publicação de um código no GitHub

Configurar o **repositório** no **GitHub**:



Acesse o GitHub e **faça login** na sua conta.



Crie um novo repositório e nomeie-o, por exemplo, simple-python-calculator.



Inicialize o repositório com um README e escolha um .gitignore apropriado para Python.

Exposição



© Getty Images

Criar uma nova *branch*

Dentro do repositório **no GitHub**, clique em “main” (ou o nome da sua branch principal) **para abrir o menu *dropdown de branches***.

No campo de texto, **digite o nome da sua nova *branch*** (por exemplo, feature-calculadora) **e pressione “Enter”**.

Adicionar o código do programa

- 1 Na nova *branch*, **clique em “Add file” e selecione “Create new file”**;
- 2 **Nomeie o arquivo como calculadora.py.;**
- 3 No editor de texto que aparece, **digite o código do seu programa.**
Por exemplo:

```
def soma(a, b):  
    return a + b  
  
if __name__ == "__main__":  
    num1 = float(input("Digite o primeiro número: "))  
    num2 = float(input("Digite o segundo número: "))  
    resultado = soma(num1, num2)  
    print(f"Resultado: {resultado}")
```


Adicionar o código do programa

4

Role para baixo e, **na seção "Commit changes", adicione uma mensagem de commit;**

5

Certifique-se de que a opção **"Commit directly to the feature-calculadora branch"** está selecionada;

6

Clique em "Commit changes".

Criação de um *pull request*

1

No repositório, **clique em “Pull requests”** e depois em **“New pull request”**.

2

Selecione base: main e compare: feature-calculadora.

3

Clique em **“Create pull request”**.

4

Adicione um título e uma descrição, se desejar, e clique em **“Create pull request”** novamente.

Exposição

Revisão do código e *merge*

Peça a um colega **para revisar o código** ou faça a revisão você mesmo.

Após a revisão, **clique em “Merge pull request” e confirme.**

Resultado esperado:

Fusão das alterações (*merge*)

Quando você clica em “Merge pull request”, está instruindo o sistema a combinar as mudanças da *branch* do *pull request* com a *branch* de destino (geralmente a *branch* principal ou master). O sistema tentará automaticamente mesclar as alterações. Se não houver conflitos, o *merge* será concluído com sucesso.

Atualização da *branch* de destino

Uma vez realizado o *merge* com sucesso, a *branch* de destino será atualizada com as alterações do *pull request*. Isso significa que as novas alterações agora fazem parte da *branch* principal do projeto.

Elaborado especialmente para o curso.

Vamos
fazer uma
atividade

Conhecer a plataforma GitHub

O objetivo desta atividade é
criar um novo repositório.

Para isso, leia e realize os
passos que se seguem:

 20 min



Será enviado
pelo AVA



1

Você acaba de sair de uma reunião da empresa em que trabalha, em que o novo software que será desenvolvido foi apresentado. Para adiantar o processo, seu *squad* ficou responsável por criar o repositório e configurar as políticas principais.



2

Crie o repositório da nova aplicação, chamada Tech Solutions, e parametrize os principais recursos de configuração.



3

Ao término do processo, todos os parâmetros definidos na configuração, assim como o link do novo repositório, devem ser enviados por e-mail, de forma que a equipe de desenvolvimento já possa iniciar sua utilização.



Preencha o arquivo completo da atividade prática, que será disponibilizado para os estudantes.

Vamos
fazer uma
atividade

Sugestão-padrão de resposta

Assunto: Configuração do novo repositório - Tech Solutions

Para: [Lista de e-mails da equipe de desenvolvimento]

Corpo do e-mail:

Olá, equipe!

Espero que estejam todos bem. Estou escrevendo para informar que o repositório para o nosso novo projeto, "Tech Solutions", foi criado e configurado com as políticas principais. Abaixo estão os detalhes do repositório e as configurações realizadas:

Nome do repositório: Tech Solutions

Link do repositório: [URL do Repositório]

Configurações iniciais:

Branches:

Branch principal: main

Branches de desenvolvimento: criadas conforme a necessidade de *features*.

Políticas de *merge*:

Proteção do *branch* principal: habilitada.

Revisão de código obrigatória: todos os *pull requests* devem ser revisados antes do *merge*.

Testes automatizados: configurados para rodar em todos os *pull requests*.

Colaboradores & equipes:



Vamos
fazer uma
atividade

Continuação da sugestão- -padrão de resposta

Desenvolvedores: acesso completo.

Testadores: acesso de leitura e criação de *issues*.

Gerentes de projeto: acesso administrativo.

Arquivos de configuração:

README.md: criado com informações básicas do projeto.

.gitignore: configurado para ignorar arquivos desnecessários.

Licença: licença MIT adicionada.

Integrações e ferramentas:

CI/CD: configuração inicial realizada.

Ferramentas de análise de código: integradas para manter a qualidade do código.

Próximos passos:

Solicitamos que todos os membros da equipe acessem o repositório, revisem as configurações e comecem a configurar seus ambientes de desenvolvimento locais. Quaisquer sugestões ou alterações adicionais nas configurações do repositório devem ser discutidas com a equipe.

Agradecemos pela colaboração de todos e estamos ansiosos para começar este emocionante projeto juntos. Por favor, não hesitem em entrar em contato para tirar quaisquer dúvidas ou obter esclarecimentos.

Atenciosamente,

[Seu nome]

[Seu cargo]

[Informações de contato]





O que nós
aprendemos
hoje?

Hoje desenvolvemos:

- 1** A aprendizagem **do processo de publicação de código e a criação de uma nova *branch* utilizando o GitHub.**
- 2** O conhecimento da **prática de revisão de código antes de efetivar uma publicação.**
- 3** Uma atividade de **criação de um novo repositório e de sua parametrização em equipes.**

© Getty Images



Saiba mais

Durante essa aula, citamos papéis dentro de um **squad** que **são muito importantes no quesito de agilidade para a gestão de projetos e entrega de soluções**. Que tal nos aprofundarmos um pouco mais no que é **gestão de projetos e organização de times**?

“Scrum // Dicionário do Programador”, do canal CÓDIGO FONTE TV. Disponível em:
https://www.youtube.com/watch?v=3aCww_1RnL0.
Acesso em: 5 fev. 2024.

Referências da aula

CÓDIGO FONTE TV. Scrum // Dicionário do Programador. Disponível em: https://www.youtube.com/watch?v=3aCww_IRnL0. Acesso em: 5 fev. 2024.

NEIVA, M. Como criar um repositório público no GitHub. **Medium**, 3 ago. 2022. Disponível em: <https://maribneiva.medium.com/como-criar-um-reposit%C3%B3rio-p%C3%ABlico-no-github-d2a4e1903049>. Acesso em: 5 fev. 2024.

Identidade visual: Imagens © Getty Images

Educação Profissional Paulista

Técnico em
**Desenvolvimento
de Sistemas**