
Atividade de Laboratório: Introdução à Colaboração com Git e GitHub

Disciplina: CMP2304

Objetivos:

- Aprender o fluxo de trabalho básico de um sistema de controle de versão distribuído.
- Praticar a criação e gerenciamento de repositórios no GitHub.
- Colaborar em um projeto simples, gerenciando contribuições individuais.
- Entender o uso de branches para desenvolvimento de funcionalidades paralelas.
- Familiarizar-se com os comandos essenciais do Git: `clone`, `pull`, `add`, `commit`, `push` e `merge`.

Pré-requisitos:

- Possuir uma conta ativa no GitHub.
- Ter o Git instalado e configurado em sua máquina local (com `user.name` e `user.email`).

Instruções Gerais

Para esta atividade, vocês se organizarão em grupos de 4 alunos. Um aluno será o "Líder do Projeto" e ficará responsável por criar o repositório e adicionar os outros três como colaboradores. Todos os membros do grupo, incluindo o líder, deverão executar as tarefas de colaboração.

Parte 1: Configuração do Repositório (Apenas para o Líder do Projeto)

O aluno designado como líder do grupo deve seguir os passos abaixo.

1.1. Criar um Repositório no GitHub:

1. Acesse sua conta no GitHub.
2. Clique no ícone de "+" no canto superior direito e selecione **"New repository"**.
3. Preencha os seguintes campos:
 - **Repository name:** `lab-git-colaborativo-cmp2304`
 - **Description:** Repositório para a atividade de laboratório da disciplina CMP2304.
 - **Public/Private:** Selecione **Public**.
 - Marque a opção **"Add a README file"**.
4. Clique em **"Create repository"**.

1.2. Adicionar Colaboradores:

1. Dentro do seu novo repositório, clique na aba **"Settings"**.
2. No menu lateral esquerdo, clique em **"Collaborators"**.
3. Clique no botão **"Add people"**.

4. Digite o nome de usuário do GitHub dos outros 3 membros do seu grupo, um de cada vez, e convide-os para o repositório.
5. Os outros membros do grupo precisam aceitar o convite que receberão por e-mail ou na seção de notificações do GitHub.

1.3. Criar o Arquivo de Lista de Alunos:

1. Volte para a página principal do seu repositório (aba **"Code"**).
2. Clique no botão **"Add file"** e selecione **"Create new file"**.
3. No campo de nome do arquivo, digite: `LISTA DE ALUNOS.txt`
4. No campo de conteúdo do arquivo, adicione um cabeçalho simples, como:

Lista de Presença - Atividade de Laboratório Git/GitHub

- 5.
- 6.
7. Role para baixo e clique em **"Commit new file"**.

Parte 2: Colaboração (Para TODOS os Alunos do Grupo)

Agora, todos os 4 membros do grupo deverão executar os passos a seguir em suas próprias máquinas.

2.1. Clonar o Repositório:

1. No GitHub, na página principal do repositório, clique no botão verde **"< > Code"**.
2. Copie a URL HTTPS (ex:
`https://github.com/usuario/lab-git-colaborativo-cmp2304.git`).
3. Abra o terminal (ou Git Bash no Windows) em sua máquina local, navegue até o diretório onde deseja salvar o projeto e execute o comando:
4. Bash

```
git clone <URL_DO_REPOSITÓRIO_COPIADA>
```

- 5.
- 6.

2.2. Puxar Atualizações (Boa Prática):

Antes de começar a fazer suas alterações, é fundamental garantir que sua versão local do projeto esteja sincronizada com a versão mais recente do repositório remoto.

Bash

```
git pull origin main
```

2.3. Adicionar seu Nome à Lista:

1. Abra o arquivo `LISTA DE ALUNOS.txt` em um editor de texto.
2. Adicione seu número de matrícula e seu nome completo em uma nova linha. O arquivo deverá ficar parecido com isto (cada aluno adiciona sua própria linha):

Lista de Presença - Atividade de Laboratório Git/GitHub

20250001 Fulano de Tal

20250002 Ciclano da Silva

- 3.
- 4.
5. Salve o arquivo.
6. Agora, use os seguintes comandos no terminal para enviar sua alteração:
7. Bash

Adiciona o arquivo modificado à área de Staging

`git add "LISTA DE ALUNOS.txt"`

Faz o commit da alteração com uma mensagem descritiva

`git commit -m "Adiciona <SEU_NOME> à lista de alunos"`

Envia o commit para o repositório remoto no GitHub

`git push origin main`

- 8.
9. **Atenção:** É possível que, ao tentar fazer o `push`, você encontre um erro de "conflito". Isso ocorre se outro colega já enviou uma alteração no mesmo arquivo. Se isso acontecer, execute `git pull origin main` novamente para baixar as alterações dele, resolva os conflitos localmente no arquivo de texto, e então execute os comandos `add`, `commit` e `push` novamente.

2.4. Criar e Publicar sua Branch Pessoal:

Cada aluno deve criar uma branch para trabalhar de forma isolada.

1. Use o comando abaixo para criar uma nova branch e já mudar para ela. Substitua `<SEU_NUMERO_DE_MATRICULA>` pelo seu número real.
2. Bash

`git checkout -b <SEU_NUMERO_DE_MATRICULA>`

- 3.
- 4.
5. Envie sua nova branch para o repositório no GitHub:
6. Bash

`git push -u origin <SEU_NUMERO_DE_MATRICULA>`

- 7.

8.

2.5. Adicionar seu Currículo (na sua branch):

1. Dentro da sua nova branch, crie um arquivo HTML simples chamado curriculo.html. O conteúdo pode ser básico, apenas para cumprir a tarefa.

Exemplo de conteúdo para curriculo.html:

2. HTML

```
<!DOCTYPE html>
<html>
<head>
  <title>Currículo de <SEU_NOME></title>
</head>
<body>
  <h1><SEU_NOME_COMPLETO></h1>
  <p>Estudante de Ciência da Computação.</p>
  <p>Matrícula: <SEU_NUMERO_DE_MATRICULA></p>
</body>
</html>
```

- 3.

- 4.

5. Adicione, faça o commit e envie este novo arquivo para a **sua branch** no GitHub:

6. Bash

```
git add curriculo.html
```

```
git commit -m "Adiciona arquivo de currículo"
```

```
git push
```

- 7.

- 8.

2.6. Adicionar Arquivo de Apresentação (na sua branch):

1. Ainda na sua branch, crie um arquivo de texto com o seu nome (ex: fulano_de_tal.txt).
2. Dentro deste arquivo, adicione a seguinte frase, substituindo <NOME DO ALUNO> pelo seu nome completo:

Sou aluno da disciplina CMP2304 e meu nome é <NOME DO ALUNO>

- 3.

- 4.

5. Adicione, faça o commit e envie este arquivo para a **sua branch**:

6. Bash

```
git add <SEU_NOME>.txt
git commit -m "Adiciona arquivo de apresentação pessoal"
git push
```

- 7.
- 8.

Parte 3: Finalização - Integrando seu Trabalho (Merge)

O passo final é integrar o trabalho que você fez em sua branch pessoal (<SEU_NUMERO_DE_MATRICULA>) de volta à branch principal (main). Isso é feito através de um **Pull Request**.

3.1. Criar um Pull Request:

1. Acesse a página do repositório no GitHub.
2. O GitHub provavelmente mostrará uma notificação amarela dizendo que sua branch teve um push recente, com um botão **"Compare & pull request"**. Clique nele.
3. Se não vir a notificação, vá para a aba **"Pull requests"** e clique em **"New pull request"**.
4. Configure o Pull Request:
 - **base:** main
 - **compare:** <SEU_NUMERO_DE_MATRICULA>
5. Adicione um título claro, como Finaliza atividades de <SEU_NOME>.
6. Clique em **"Create pull request"**.

3.2. Realizar o Merge:

1. O Líder do Projeto (ou qualquer membro do time) irá agora ver o Pull Request na aba correspondente.
2. Analise as alterações na aba **"Files changed"**.
3. Se tudo estiver correto e não houver conflitos, clique em **"Merge pull request"** e depois em **"Confirm merge"**.

Parabéns! Você completou a atividade de laboratório e praticou um fluxo de trabalho colaborativo completo com Git e GitHub.