**Sumário**

[**1.** **Links para materiais teóricos das aulas:** 2](#_Toc124797802)

[**2.** **Conceitos de testes** 3](#_Toc124797803)

[**3.** **GIT** 5](#_Toc124797804)

[**3.1.** **Comandos do GIT** 6](#_Toc124797805)

# **Links para materiais teóricos das aulas:**

* **Introdução ao desenvolvimento moderno de Software** - Somente conceitos da DEV, Infra, BD, Cloud e tals

Apresentação completa: <https://academiapme-my.sharepoint.com/:p:/g/personal/nubia_dio_me/EYHcjptuOoNPs4qzd2upfmwBaLoG_FfSdzZH3zJiBvABiw?e=XYsmFR>

* **Apresentação Fundamentos de Qualidade de Software – Somente teoria**

<https://docs.google.com/presentation/d/1lf5Z39hwxqIa-WGLAxkcuJTpEjO-TfDz/edit#slide=id.p15>

* **Metodologias Ágeis e Ciclo de Desenvolvimento de Software**

[3 - Ciclo de desenvolvimento de software e metodologias ágeis.pptx - Apresentações Google](https://docs.google.com/presentation/d/1Wugx7FflaNrMHo_Ec57rDW0oJdDdlSHB/edit#slide=id.p2)

* **CTFL**

<https://docs.google.com/presentation/d/1CGgJrm5yCwyM4BYPL3dzFSGfoEXw4G67/edit#slide=id.p2>

* **Dia a dia de um QA (projeto)**

[Lab 1 - O dia a dia de um QA\_ A prática de testes manuais funcionais .pptx (sharepoint.com)](https://academiapme-my.sharepoint.com/:p:/g/personal/renato_dio_me/EZn3n5RAgFFJgz1SePsXsxkBivxcAhWx90n9jzvRKFnX2Q?e=019g8H)

* **JAVA:**

<https://glysns.gitbook.io/java-basico/>

* **Criando repositório no GitHub**

<https://drive.google.com/file/d/1IZu0qohv1JOmxjEra1lknDiiStU68bl4/view>

Tabela

Descrição gerada automaticamente com confiança baixa

# **Conceitos de testes**

Diagrama

Descrição gerada automaticamenteDiagrama

Descrição gerada automaticamenteDiagrama

Descrição gerada automaticamente

# **GIT**

COMANDOS PARA GERAÇÃO DE CHAVE PUB E PRIV DO GIT

1. Gera a chave

$ ssh-keygen -t ed25519 -C yubiara@hotmail.com

1. Copia NO TERMINAL E
2. Uma imagem contendo Diagrama

   Descrição gerada automaticamenteCOLA NO GIT

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  | Diagrama  Descrição gerada automaticamente |  |  |
|  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
|  |  |  |  |  | |
|  | Texto  Descrição gerada automaticamente |  |  |
|  |  |  |  |  |
|  |  | Diagrama  Descrição gerada automaticamente com confiança médiaInterface gráfica do usuário, Aplicativo  Descrição gerada automaticamente | |  |
|  |  |  |  |

## **Comandos do GIT**

1. Git clone

Git clone é um comando para baixar o código-fonte existente de um repositório remoto (como, por exemplo, o Github). Em outras palavras, git clone, basicamente, faz uma cópia idêntica da versão mais recente de um projeto em um repositório e a salva em seu computador.

1. Git branch

Branches (algo como ramificações, em português) são altamente importantes no mundo do git. Usando as branches, vários desenvolvedores conseguem trabalhar em paralelo no mesmo projeto simultaneamente. Podemos usar o comando git branch para criar, listar e excluir as branches.

1. Git checkout

Esse também é um dos comandos do Git mais usados. Para trabalhar em uma branch, primeiro, é preciso "entrar" nela. Usamos git checkout, na maioria dos casos, para trocar de uma branch para outra. Também podemos usar o comando para fazer o checkout de arquivos e commits.

1. Git status

O comando git status nos dá todas as informações necessárias sobre a branch atual.

1. Git add

Ao criarmos, modificarmos ou excluirmos um arquivo, essas alterações acontecerão em nosso espaço de trabalho local e não serão incluídas no próximo commit (a menos que alteremos as configurações).

Precisamos usar o comando git add para incluir as alterações de um ou vários arquivos em nosso próximo commit.

1. Git commit

Talvez esse seja o comando mais usado do Git. Quando chegamos a determinado ponto em desenvolvimento, queremos salvar nossas alterações (talvez após uma tarefa ou resolução de problema específica).

1. Git push

Após fazer o commit de suas alterações, a próxima coisa a fazer é enviar suas alterações ao servidor remoto. Git push faz o upload dos seus commits no repositório remoto.

1. Git pull

O comando git pull é usado para obter as atualizações de um repositório remoto. Esse comando é uma combinação de git fetch e git merge, o que significa que, quando usamos git pull, ele recebe as atualizações do repositório remoto (git fetch) e aplica imediatamente as alterações mais recentes em seu espaço de trabalho local (git merge).

1. Git revert

Às vezes, precisamos desfazer as alterações que fizemos. Existem várias maneiras de se desfazer as alterações em nosso espaço de trabalho local ou remotamente (dependendo do que você necessita), mas devemos usar esses comandos com cuidado para evitar exclusões indesejadas.

1. Git merge

Quando você concluir o desenvolvimento em sua branch e quando tudo funcionar bem, a etapa final é fazer o merge (mesclar ou unir, em português) da branch com a branch pai (dev ou master/main, em geral). Isso é feito com o comando git merge.