



# GitHub



# O Uso do versionamento

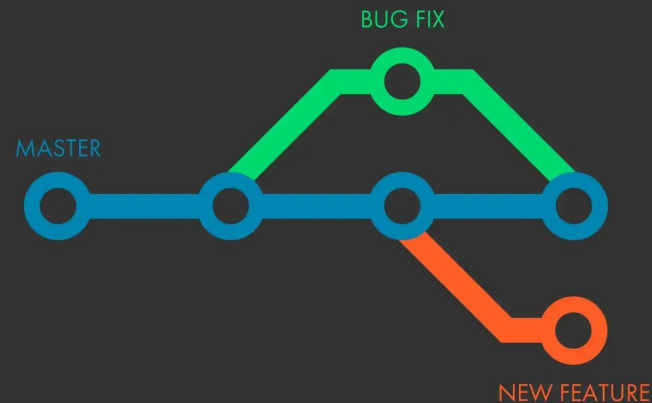
## Git Hub e Git Desktop

Prof. Fábio Leonel



1

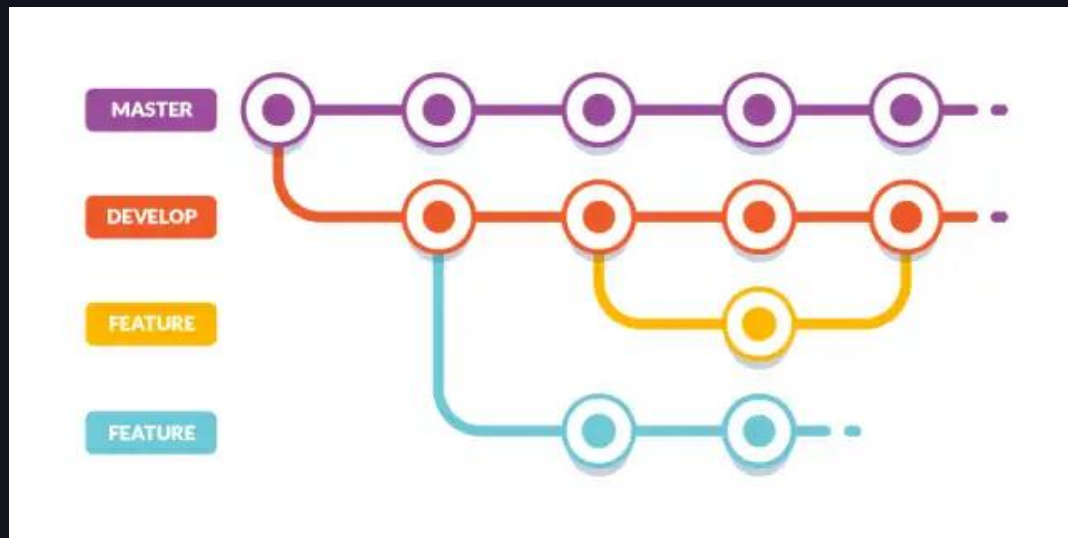
# Versionamento de software



# O que é?

- ▶ O versionamento (também conhecido como controle de versão) é um processo que acompanha e rastreia as alterações feitas no código ou arquivos de um projeto ao longo do tempo.
- ▶ Permite que as equipes de desenvolvimento e análise registrem, acompanhem e gerenciem essas alterações, desde pequenas modificações até novos recursos, correções de erros e backups.

# Os métodos Master e branches



**2**

# **Git, GitHub e Git Desktop**

# As diferenças

## ▶ Git:

- ▶ É uma ferramenta de linha de comando que permite rastrear e gerenciar alterações em um projeto.
- ▶ Permite que você salve versões de um projeto, faça "commits" (salva as alterações), branches (cria ramificações do projeto) e "merges" (junta as branches).
- ▶ Funciona localmente, em seu computador.

# As diferenças

## ► GitHub:

- É uma plataforma online, uma "rede social para desenvolvedores e analistas", que hospeda repositórios Git.
- Facilita a colaboração entre desenvolvedores e analistas, permitindo que eles compartilhem código ou arquivos, revisem mudanças, criem branches e façam "pull requests" (solicitação de integração de código ou arquivos).
- Oferece recursos adicionais como "issues" (problemas), "wiki" (documentação).
- Pode ser acessado através de um navegador web ou do GitHub Desktop.

# As diferenças

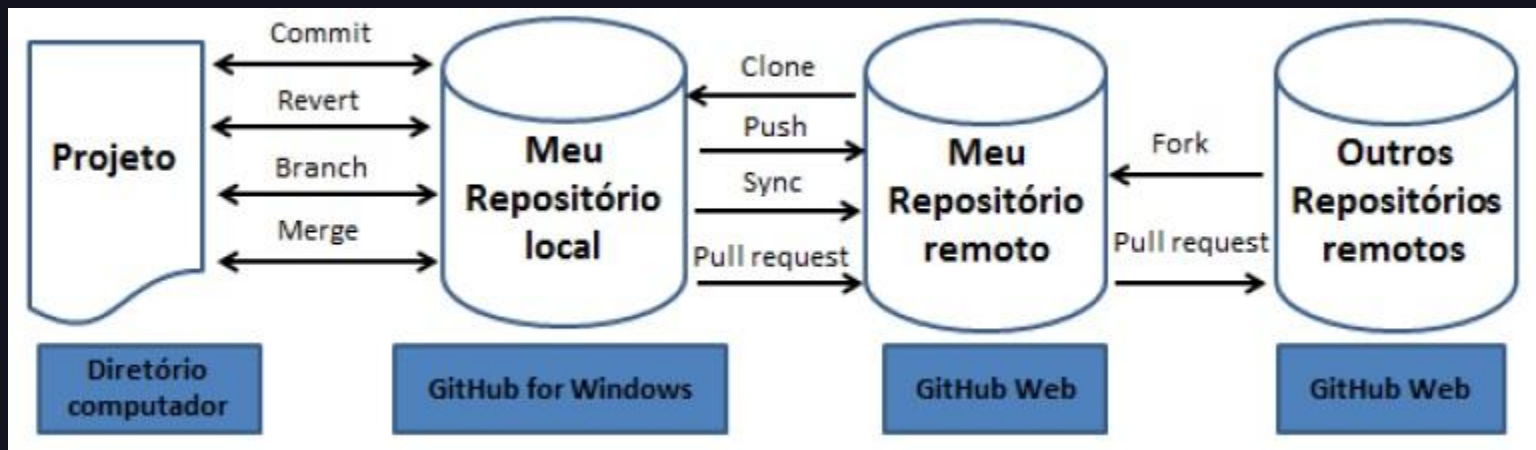
- ▶ Git Desktop:
  - ▷ É uma interface gráfica para o Git, que facilita a interação com os comandos de Git.
  - ▷ Permite visualizar o histórico de commits, criar branches, fazer "commits" e "pulls" de forma mais intuitiva, sem ter que usar a linha de comando.
  - ▷ É uma ferramenta que facilita o uso do Git, especialmente para quem está começando ou prefere uma interface gráfica para gerenciar seus repositórios.



**3**

# **Os conceitos importantes**

# Interações das funcionalidades do Git/GitHub



# Os conceitos importantes

- ▶ **Repository:** Local onde fica todos os arquivos do projeto, inclusive o histórico de versões.
- ▶ **Commit:** Coleção e controle das alterações realizadas. Sempre que for necessário é possível retroceder um commit.
- ▶ **Revert:** Reverte (retrocede) um commit efetuado.
- ▶ **Branch:** É uma ramificação do seu projeto. Cada branch representa uma versão do seu projeto. Podemos seguir uma linha de desenvolvimento a partir de cada branch.

# Os conceitos importantes

- ▶ **Push (Publish):** Envia os commits dos arquivos locais para um repositório remoto.
- ▶ **Clone:** Obtém uma cópia de um repositório remoto. Quando desejar colaborar com algum projeto de software livre utilize esta opção.
- ▶ **Pull:** Atualiza as alterações locais efetuadas com o repositório remoto.
- ▶ **Merge:** É a capacidade de incorporar alterações do Git, onde acontece uma junção dos branches.

