

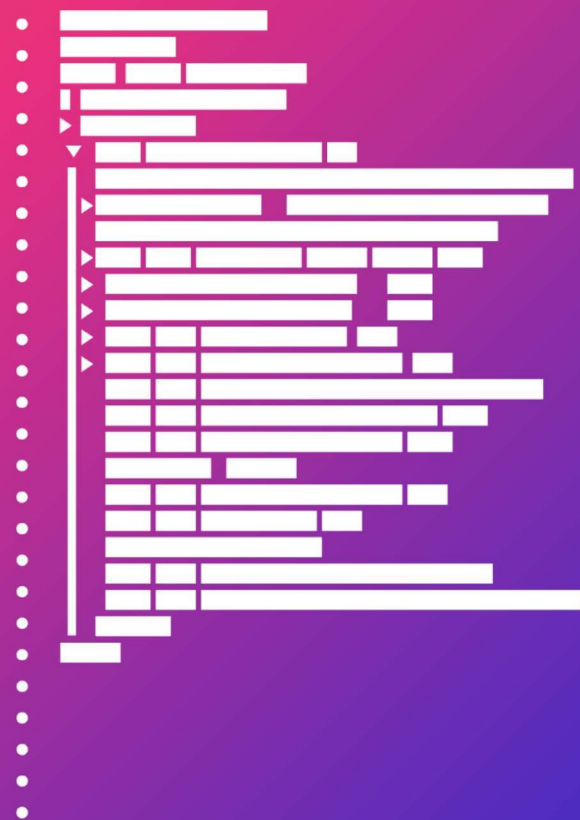
</ Versionamento de software >

DEVinHouse

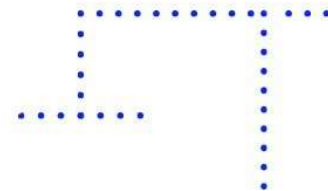
Parcerias para desenvolver a sua carreira



ACATE

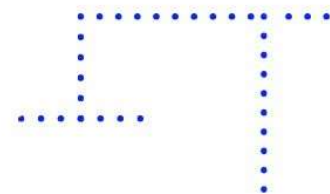


Agenda



- Versionamento
- Git básico
- Branch
- Github

.....> Versionamento



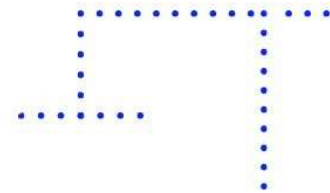
Quando nos referimos a **versionamento de código e software**, falamos de uma medida administrativa para governar a evolução de um software. Podemos identificar a origem de um bug, reutilizar códigos que só existiram em versões antigas do software e principalmente evoluir um software com a segurança de que nossas modificações não são definitivas e podem ser recuperadas a qualquer momento futuro.

.....> Versionamento

Muito comum antigamente e presente até hoje, uma forma simples de versionar um software é criar novas pastas com todo o código da versão original para novas alterações. esse gerenciamento pode ser eficiente em softwares pequenos e com poucos desenvolvedores

 softwareV1	15/03/2021 16:51	Pasta de arquivos
 softwareV2	31/03/2021 16:35	Pasta de arquivos

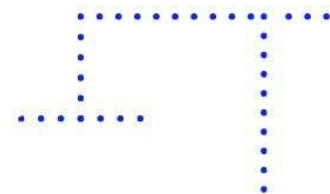
Versionamento semântico



Versionamento semântico reflete a necessidade do software ser representado numericamente por versões. Comumente vemos v1 ou v2 mas há um padrão segundo a especificação.

- Um número de versão normal DEVE ter o formato de X.Y.Z, onde X, Y, e Z são inteiros não negativos, e NÃO DEVE conter zeros à esquerda. X é a versão Maior, Y é a versão Menor, e Z é a versão de Correção. Cada elemento DEVE aumentar numericamente. Por exemplo: 1.9.0 -> 1.10.0 -> 1.11.0.

.....> Versionamento

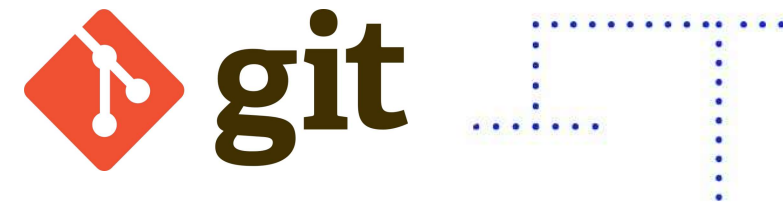


softwareV1	15/03/2021 16:51	Pasta de arquivos
softwareV2	31/03/2021 16:35	Pasta de arquivos

Apesar do versionamento com pastas ter o seu valor histórico. Hoje existem soluções bem mais modernas de como governar as **alterações de código**. Estes “**sistemas de versionamento de código**” auxiliam no desenvolvimento colaborativo.



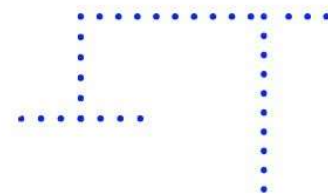
> **Git básico**



Criado por Linus Benedict Torvalds e lançado em 2005, se tornou a principal ferramenta de controle de versão.

Através dela podemos desenvolver projetos na qual diversas pessoas podem contribuir simultaneamente editando e criando novos arquivos e permitindo que os mesmos possam existir sem o risco de suas alterações serem sobrescritas.

Git básico



Fluxo de trabalho:

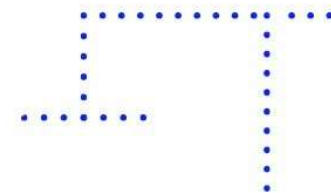
- Você modifica arquivos no seu diretório de trabalho.
- Você **prepara** os arquivos, adicionando eles à sua área de preparo.
- Você faz o **commit**, que leva os arquivos como estão na área de preparo e armazena de forma permanente no diretório do Git.

Git básico

Exemplo de utilização:

- Em uma pasta com os arquivos, iniciaremos um repositório git com “**git init**”.
- Em seguida se deve **preparar** os arquivos para adicioná-los de fato ao repositório. Fazemos isso com “**git add .**”
 - Note que o “.” representa o diretório atual e tudo que há nele
 - “git add index.html” ou “git add img/imagem.png” ou ainda “git add img/.” para tudo dentro de uma pasta
- Por fim criamos um commit integrando o código com “**git commit -m “mensagem do commit”**”

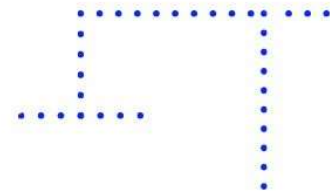
Git básico



Além do gerenciamento local do repositório, existe o gerenciamento **remoto**, ou simplesmente **repositório remoto**

Esse repositório **é o original** do qual os locais são uma cópia, e é para onde as alterações dos repositórios locais vão.

> Git básico



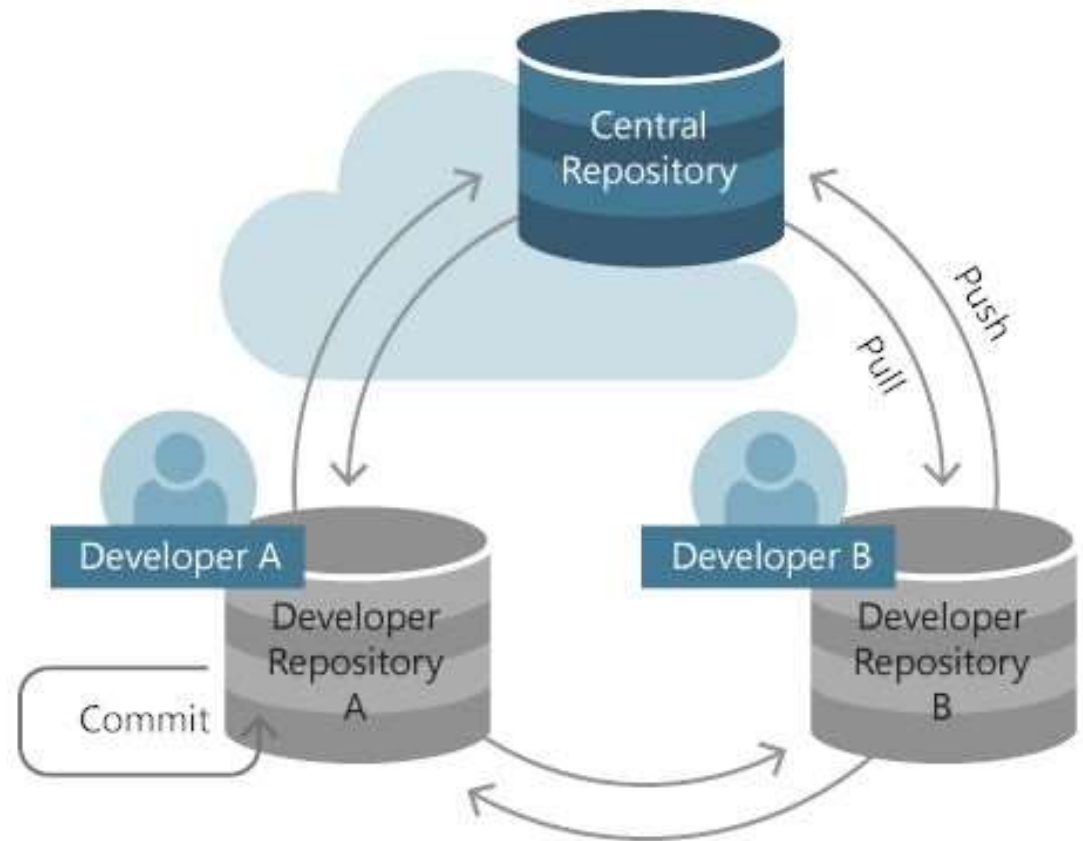
Os repositórios podem ser **públicos** ou **privados**. Em ambos os casos para alterarmos o repositório principal precisamos de uma permissão correspondente.

Essa permissão pode ser concedida caso o repositório não seja seu mas existe outro processo de alterar o código principal de forma colaborativa sem a devida permissão.

Git básico

Fluxo de trabalho remoto:

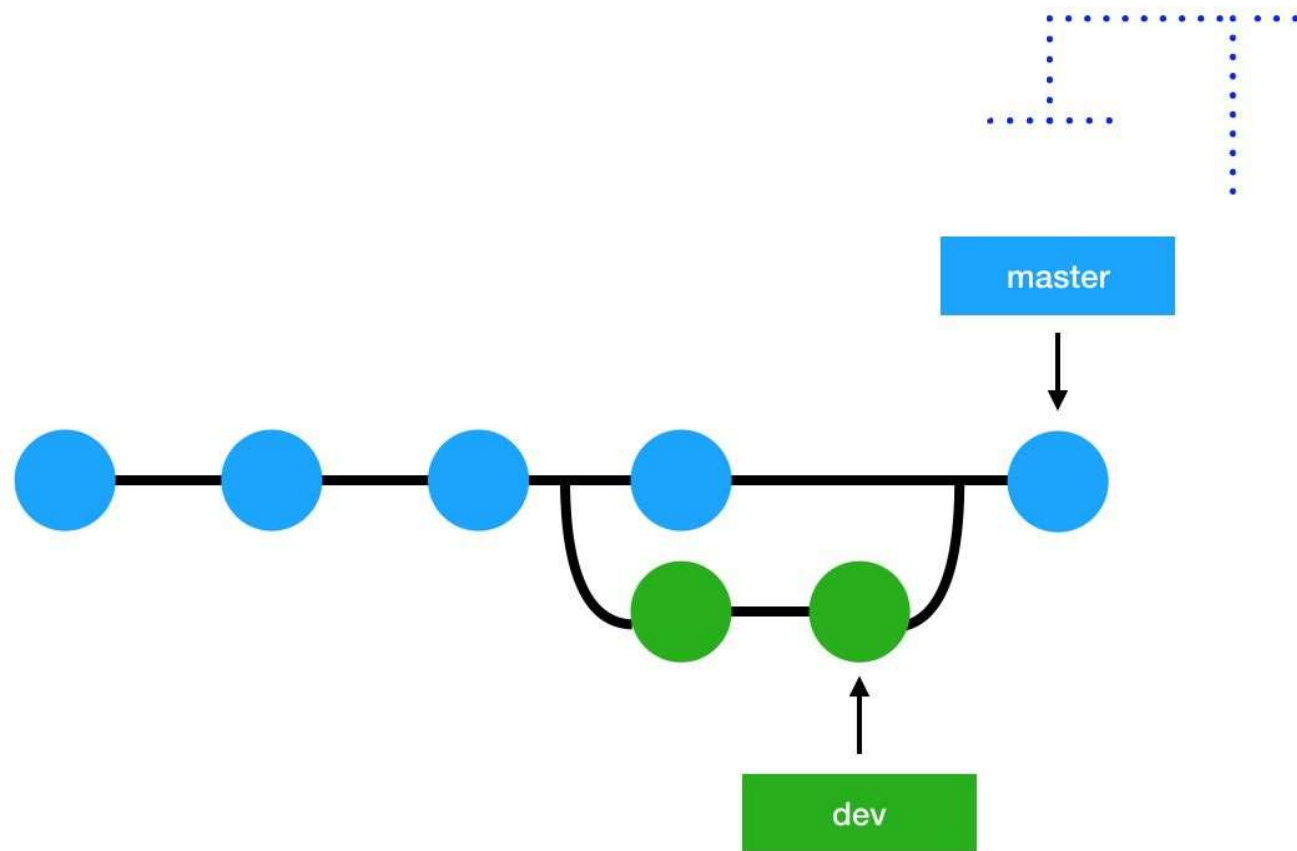
- Obtemos um repositório remoto com “**git clone url-do-repositório**”
- Enviamos uma alteração de código(commit) com “**git push**”
- Obtemos as alterações de um repositório posteriores a ser clonado com “**git pull**”



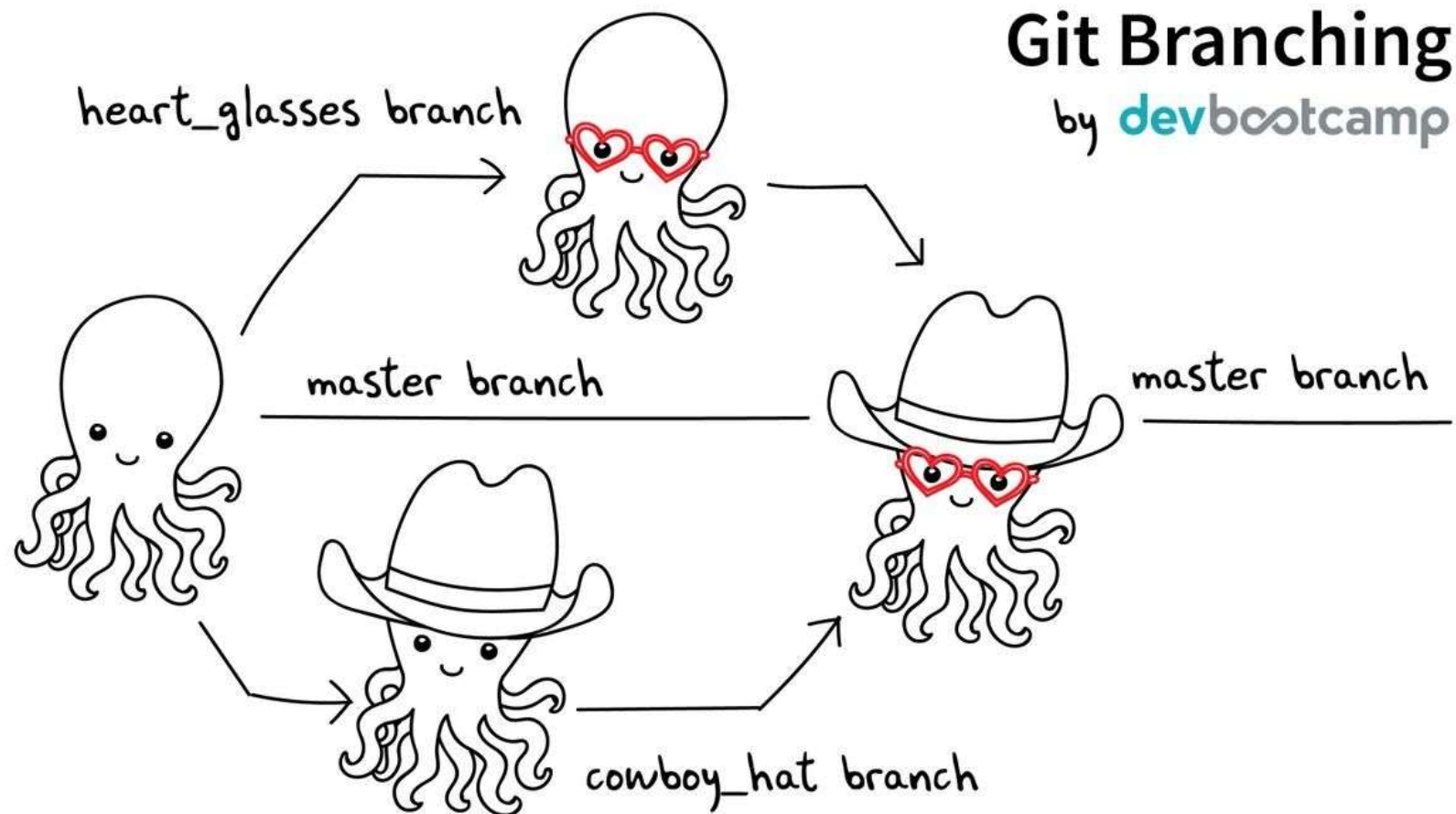
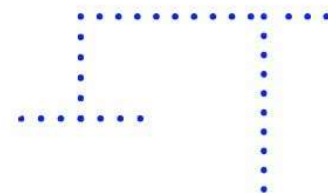
> Branch

Outro conceito básico no git é o de **branches** ou **ramificações**.

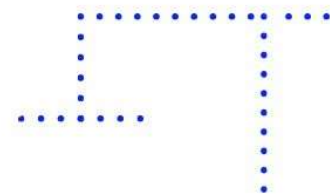
Quando criamos um novo repositório a **branch “main”** ou **“master”** é criada.



Branch



> Branch

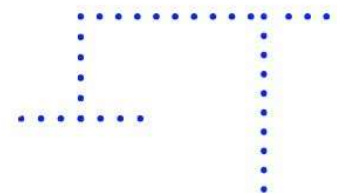


Na prática, branch começa a nos mostrar como **trabalhar de forma colaborativa**, onde eu não espero a entrega de outro desenvolvedor para executar e integrar as minhas alterações com o repositório remoto.

Para criar uma nova branch usamos “**git checkout -b nome-da-branch**”

- **git checkout -b alteraMenu**
- **git checkout -b alteraRodape**

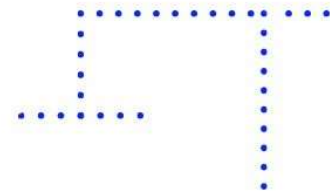
.....> **Github**



Github

- Produto da **microsoft** com diversas funcionalidades. A principal é servir como uma hospedagem de **repositórios remotos**.
- Apesar da forte associação do github com projetos **open-source**, podemos tratar projetos com diversas licenças, além de projetos privados
- Como interface, nos permite navegar facilmente pelas diversas versões de um software, permitindo identificar **quais** alterações foram feitas, **por quem** e **quando**.
- Há outros produtos como **github pages** ou **github copilot**

Github

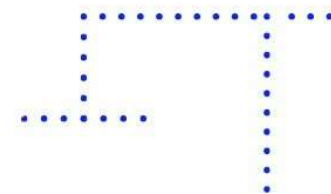


No github podemos encontrar trechos de código, módulos e sistemas completos, dos quais podemos utilizar e alterar para qualquer finalidade.

Façamos o clone do repositório

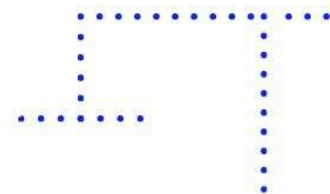
<https://github.com/thomaspark/flexboxfroggy> depois executamos o **npm install** para baixar suas dependências.

Exercício



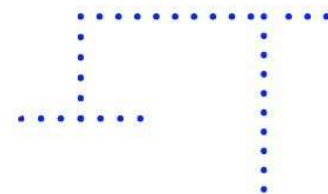
- Com uma conta no **github** crie um novo repositório vazio.
- **Clone** o projeto para sua máquina
- mova os arquivos criados durante as aulas para a nova pasta criada após o clone e execute **git add** .
- Execute o **git commit** em suas alterações
- Execute o **git push** e acesse o github para verificar seus arquivos

Para casa



- Descreva de forma sintética, informal e com suas palavras, para que serve os seguintes comandos
 - **git status**
 - **git --version**
 - **git log**
 - **git branch**
 - **git stash**
 - **git config**
- Mande em um arquivo .txt simples pelo **slack** para o professor

.....> Material Complementar



- <https://semver.org/lang/pt-BR/>
- <https://chambers.io/2018/04/17/git-vs-the-competition.html>

Obrigado



SENAI

ACATE