

# INTRO AO GIT / GITHUB

Workshop PyLadies Day

# /WHOIS

- Lívia Souza aka Liv
- Engenheira de Software no CESAR
- Formada em ADS na Unibratec e quase pós-graduada em DEVAPPS na CESAR School (falta o TCC 😓)
- Trabalhando na área de TI desde 2014 - me convide pra uma cerveja e te conto o que eu fazia antes de cair nessa cilada
- Amo praia, cerveja, heavy metal, atividades físicas, e claro, contribuir com a comunidade de TI no meu tempo livre 🤘



\*Mal chegamos em junho e já tô doida pra comer canjica 😊🌽 sim, sou de Touro.



# QUAL A DIFERENÇA?

Git vs GitHub



# SISTEMAS DE CONTROLE DE VERSÃO

São usados para monitorar mudanças em arquivos e coordenar o trabalho nestes arquivos entre várias pessoas.

Usado principalmente em desenvolvimento de software, mas pode ser usado para qualquer coleção de arquivos.

Como um sistema distribuído, tem o objetivo de suportar fluxos de trabalho não-lineares, garantindo velocidade e integridade de dados.

# USANDO GIT

Abrir um terminal, verificar se tem git instalado - Usaremos o **GIT BASH**  
Baixar em: <https://gitforwindows.org>

**git --version**

Configurando git

**git config --global user.name "meu nome"**

**git config --global user.email "meu-email@email.com"**

Verificando as configurações

**git config --list**

# BÁSICOS DO TERMINAL

Exibir localização atual

**pwd**

Listar arquivos e pastas

**ls**

Listar arquivos e pastas - incluindo detalhes e arquivos ocultos

**ls -la**

Criando uma pasta

**mkdir git-pyladies**

Entrar na pasta

**cd git-pyladies**

Dica:

- Usar a tecla tab para auto-completar

# INICIANDO O GIT

Iniciando o repositório git na pasta

**git init**

Criando um novo arquivo

**touch index.txt**

Adicionando texto no arquivo (nova linha)

**echo "Meu primeiro arquivo" >> index.txt**

Verificando o status dos arquivos (usaremos bastante esse comando)

**git status**

# STATUS DO GIT

Adicionando os arquivos para staging (preparar o commit)

**git add .** (adiciona todos os arquivos modificados)

Ou

**git add <nome-do-arquivo>** (adiciona o arquivo específico)

Verificar status dos arquivos novamente

**git status**

Submeter os arquivos adicionados

**git commit -m “Repositorio iniciado, primeiro arquivo adicionado”**

Verificar status dos arquivos novamente

**git status**

Verificar o histórico local

**git log**

# COMANDOS COMUNS

Descartar temporariamente alterações que não estão em staging

**git stash**

Adicionar de volta as últimas alterações descartadas

**git stash pop**

Remover do staging

**git reset HEAD -- <nome-do-arquivo>**

- Remove o arquivo especificado do staging

Ou

**git reset HEAD -- <nome-do-diretório>**

- Remove o diretório especificado do staging

Ou

**git reset --hard HEAD**

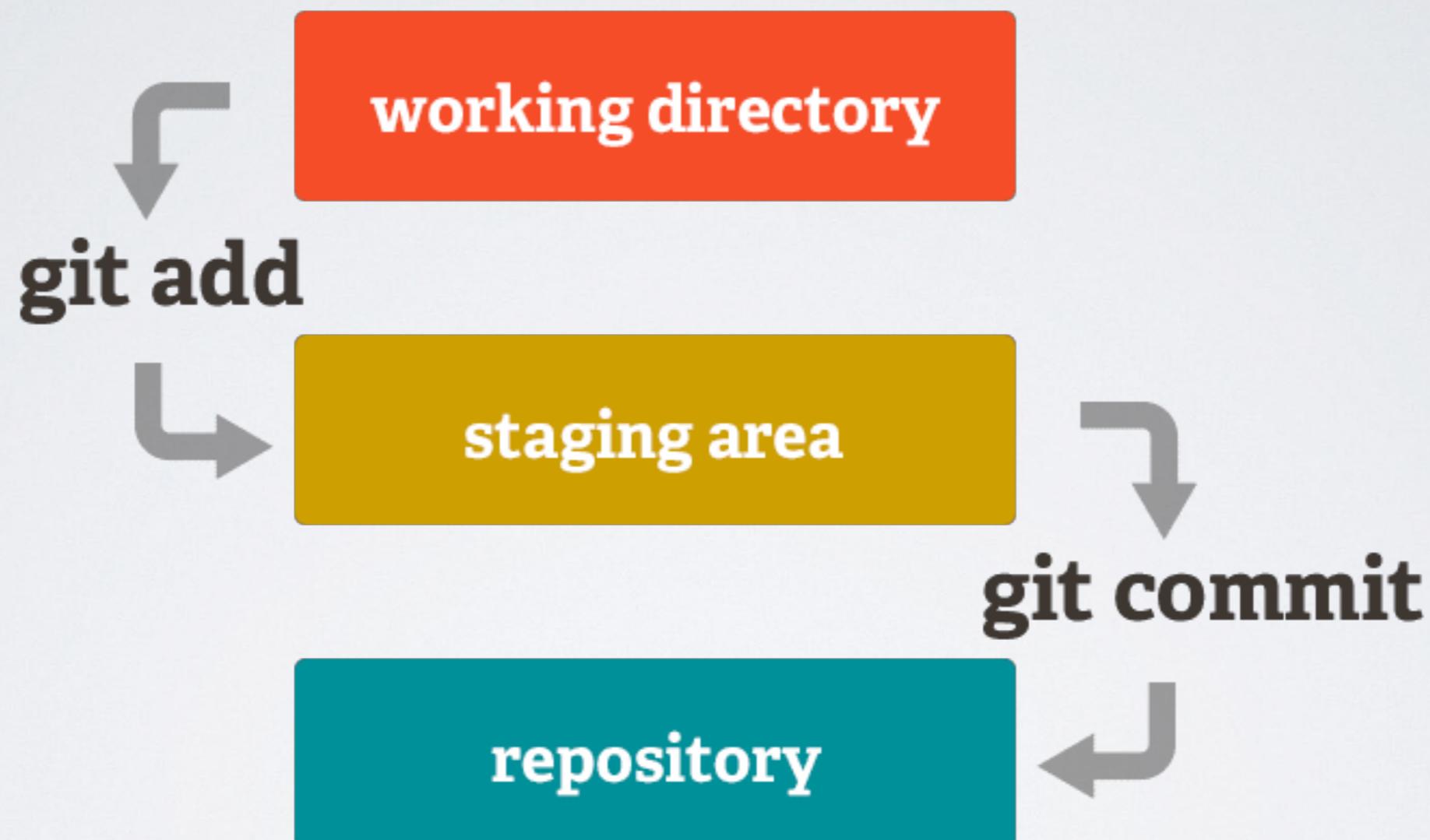
- Remove TUDO que estiver em staging - não apaga

Enviando para o repositório remoto

**git push**

Boa prática: sempre verificar o que acabou de fazer com o comando **git status**

# O QUE FIZEMOS ATÉ AGORA?



Atlassian



**Bitbucket**

**source  
forge**



**GitHub**

**GitLab**

HOSPEDAGEM REMOTA DE  
REPOSITÓRIOS GIT

# USANDO GITHUB

Crie sua conta em: **https://github.com/**

Crie um novo repositório, copie o link HTTPS gerado

ex: **https://github.com/TigerRobocop/git-pyladies.git**

De volta ao terminal, verifique o repositório remoto configurado

**git remote -v**

Adicione o link do repositório GitHub

**git remote add origin <link-repo-github>**

Verifique a configuração do repositório remoto novamente

**git remote -v**

# ENVIANDO PARA O GITHUB

Verificar o histórico local

**git log**

Subir para o repositório remoto (primeira vez)

**git push -u origin master**

Vezenes seguintes

**git push**

Volte ao site GitHub e atualize a página, para ver os arquivos disponibilizados!

# CONTRIBUINDO NO GITHUB

Crie um Fork do repositório:

**<https://github.com/TigerRobocop/git-pyladies>**

Faça um clone do repositório criado na seu local

**git clone <link-https-do-repo>**

Crie uma branch

**git branch -d <nome-da-nova-branch>**

# DESAFIOS

1. Adicione seu nome no arquivo **PARTICIPANTES.md**
2. Crie um arquivo de perfil no diretório **Profiles**
3. Crie um script Hello World em Python no diretório **Scripts**

# FONTES

- <https://medium.com/@vinhle95/are-you-a-junior-developer-lets-start-showcasing-your-work-on-github-from-today-da688a913c04>
- <https://dev.to/lberlin/a-github-guide-for-people-who-don-t-understand-github-n50>
- <https://medium.com/cs-note/git-and-github-for-beginners-i-tutorial-263caa01f9c3>
- <https://dev.to/sublimegeek/git-staging-area-explained-like-im-five-lanh>
- <https://github.com/AliceWonderland/hacktoberfest>

# OBRIGADA!

Vcs são TOP TOP TOP

[about.me/livr souza](http://about.me/livr souza)