

Git

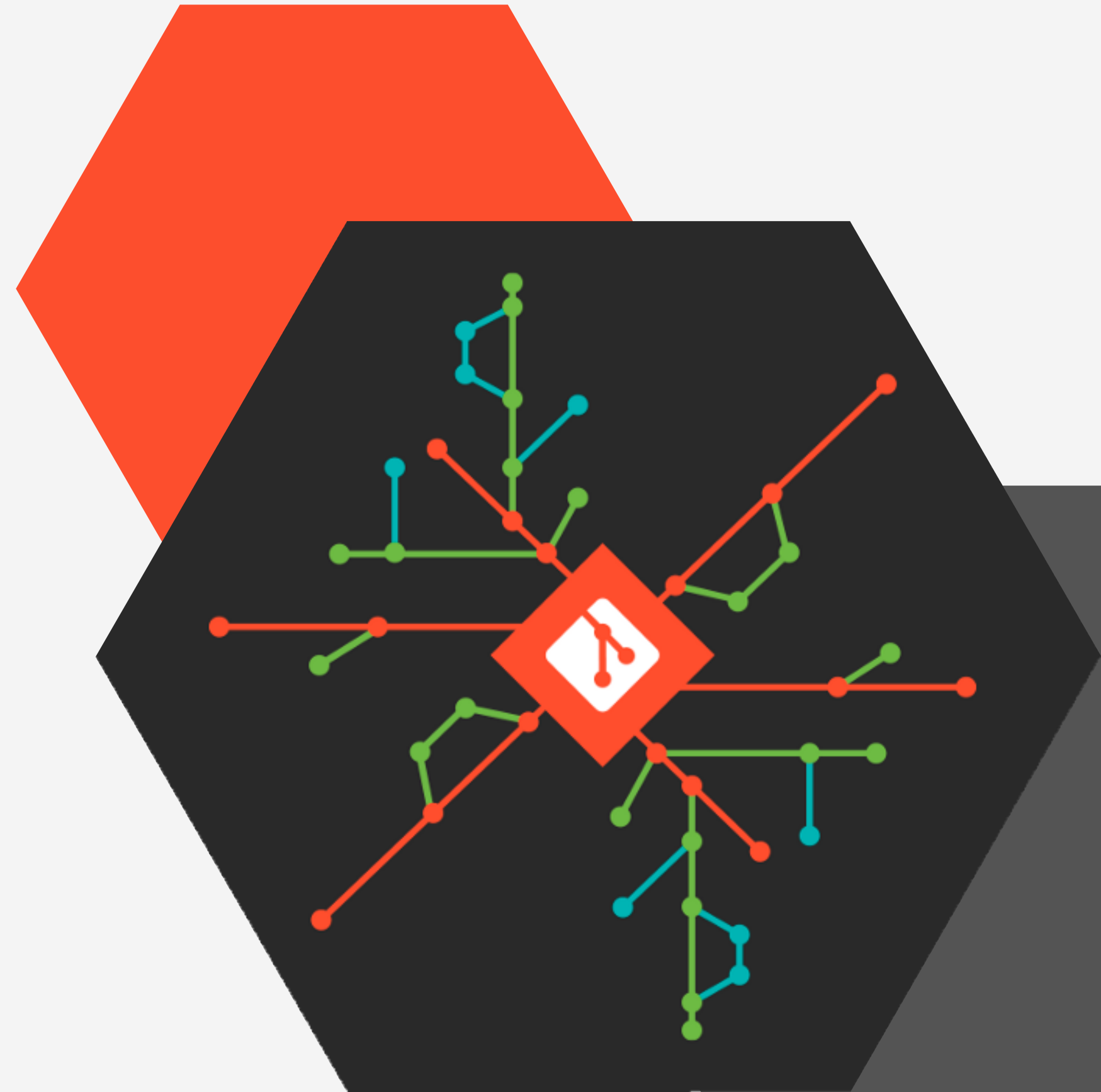
O que é, versionamento, características, e uso

Apresentadores: Felipe F. Róza, Gustavo B. Conceição,
Gustavo J. Baierski e Vinícius A. Soethe



O que é **Git**?

- Versionamento, Colaboração, Armazenamento



Por que **Git** é importante?



Divisão de projetos

Projetos individuais ou em equipe, salvamento na nuvem e versionamento.

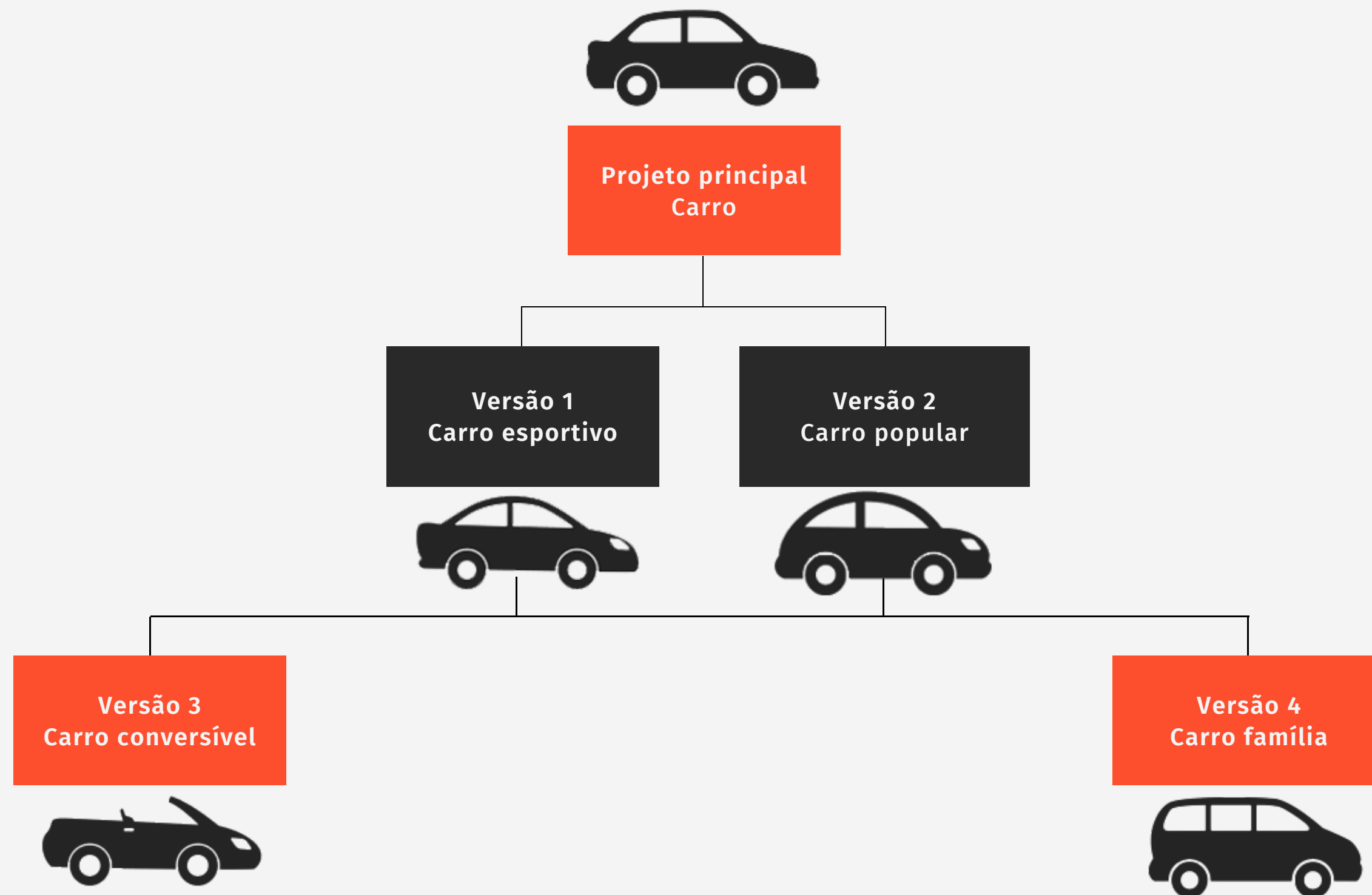
Mercado

Empresas, "dependência" e importância

Experiências próprias

Como foram nossas experiências com o Git?

O que é **versionamento**?



Como funciona o versionamento do Git?



modified



staged



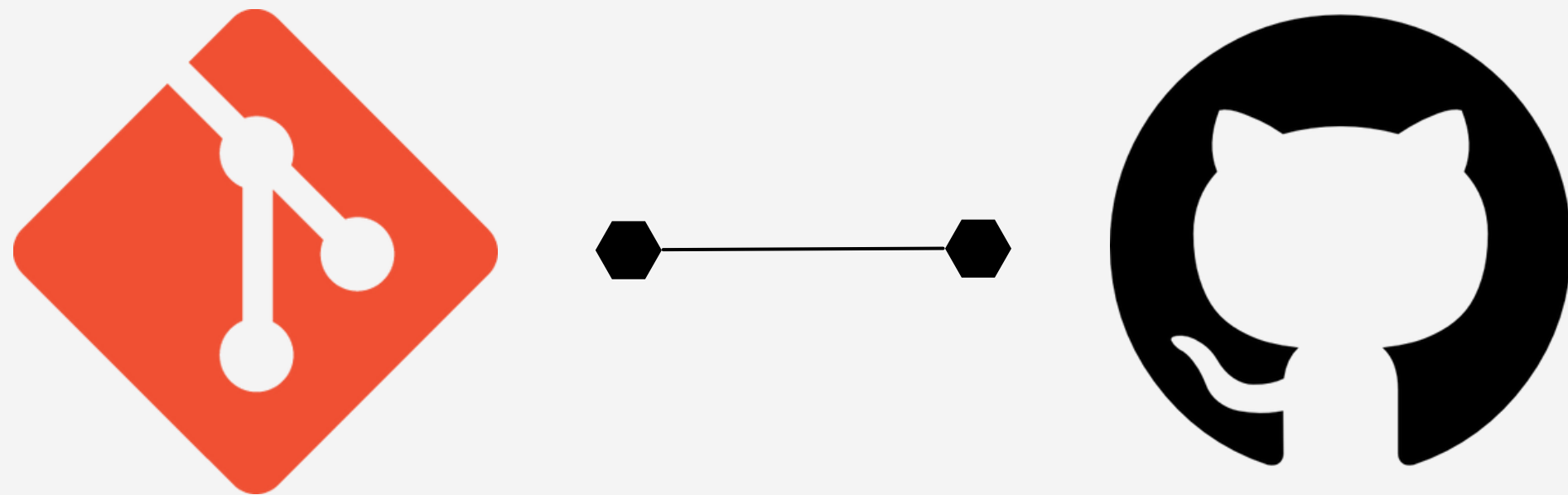
committed



cloud



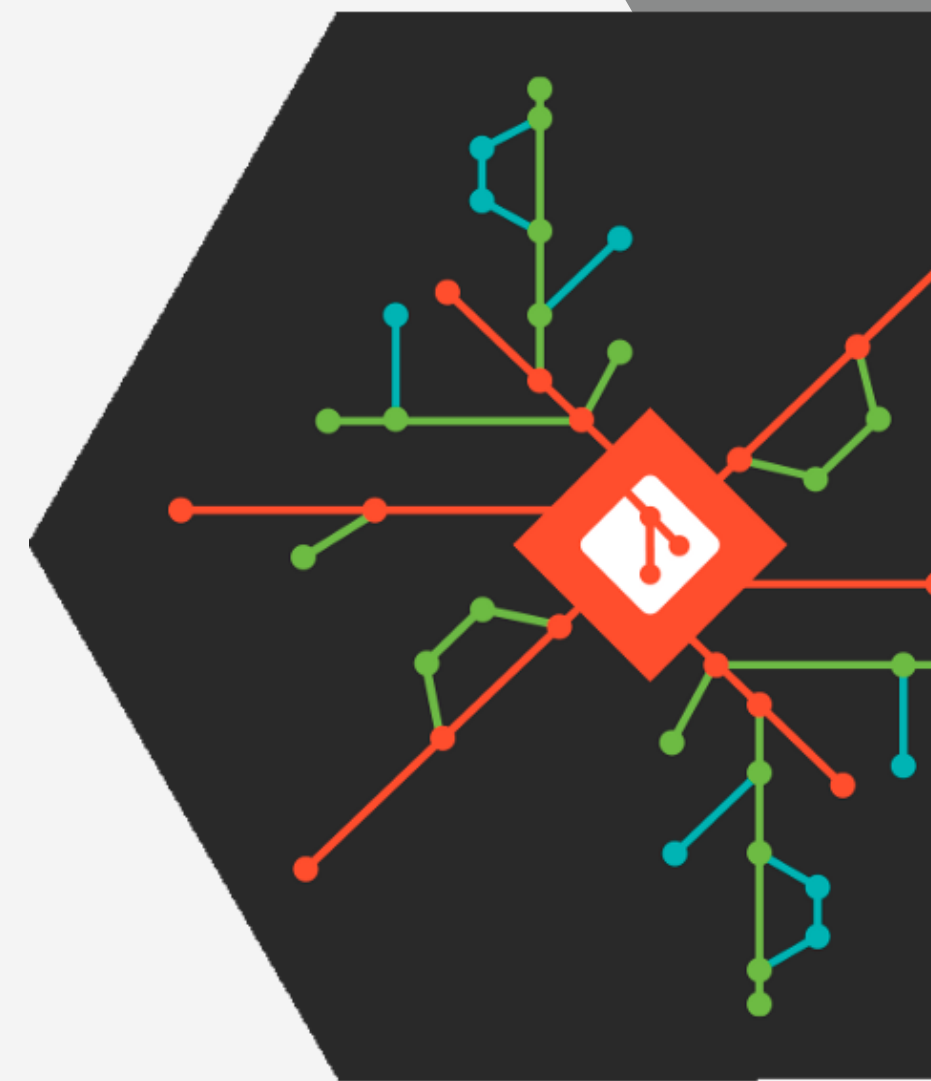
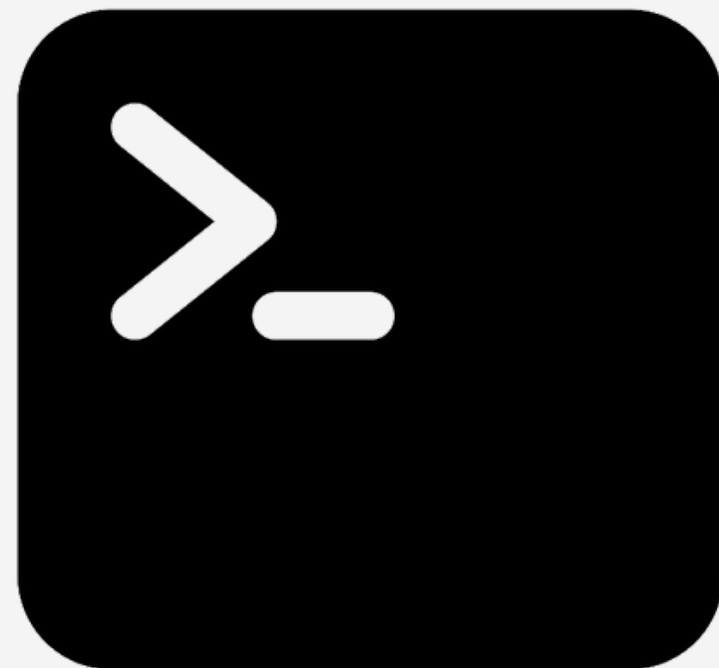
Diferença entre Git e Github



Plataformas de armazenamento:



Comandos do Git



Configuração inicial

```
git config -l
```

```
git config user.name NOME
```

```
git config user.email EMAIL@EXEMPLO.COM
```

```
git config --global
```



Iniciando um Repositório



git init

VS

fork

VS

git clone



O que é um **commit**?



Mudanças locais

`git status`

`git add . (-A, --all, *)`

`git diff [branch]`

`git commit -m "mensagem do commit"`



A importância de "commitar" bem



Commit: "Fluxo 1 finalizado"

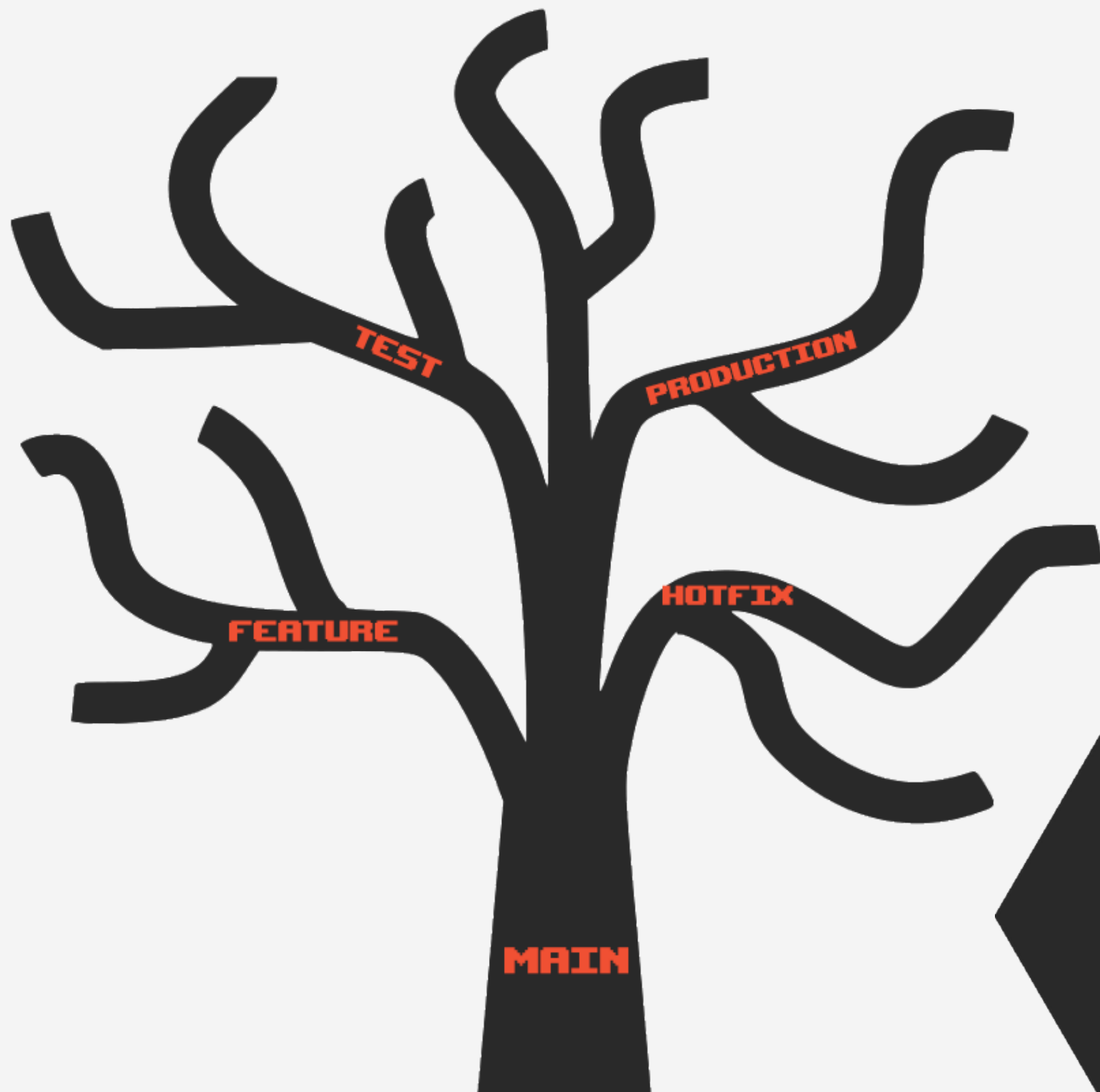


Commit 1: "Criando formulário de usuário"

Commit 2: "Criando comunicação com o backend"

Commit 3: "Corrigindo validação de ID usuário no backend"

O que é uma **branch**?



Trabalhando com **branches**

git branch

git checkout <branch>

git branch <nova branch>

git branch -d <branch>



Mesclando o desenvolvimento

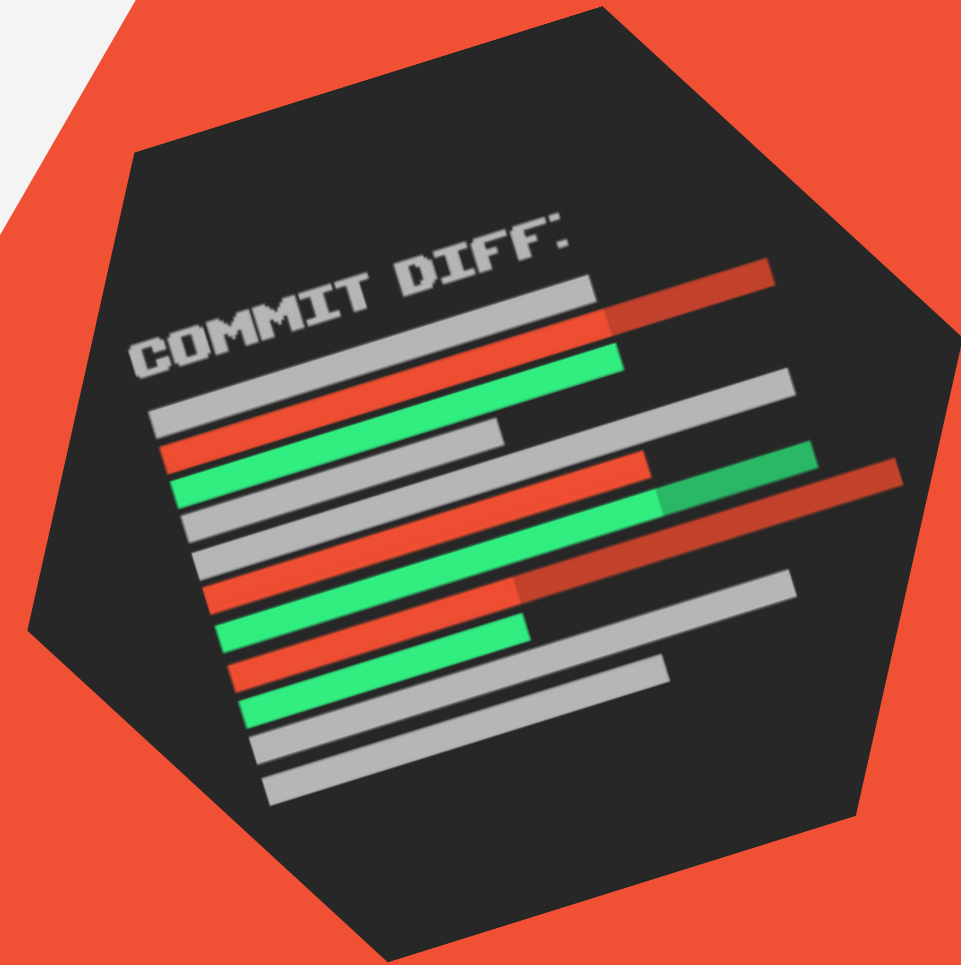
Merge & Rebase

`git merge <branch>`

`git rebase <branch>`

`git rebase --abort`

`git rebase --continue`



Atualizando o ambiente local e publicando mudanças

git fetch

git pull <remote> <branch>

git push <remote> <branch>



Revertendo e descartando alterações

git reset --hard

git checkout <arquivo>

git revert <commit>

git log



Link para o repositório:

Utilizar função **fork** do Github



github.com/ViniciusAcordiSoethe/CursoSepeGit

Quadro de tarefas:

Desenvolvedor Frontend 1:

Alterar cores do site (background, botão ou outro elemento no arquivo CSS).

Desenvolvedor Backend 1:

Adicionar um novo evento (Copiar, colar e alterar informações)

Desenvolvedor Backend 2:

Deletar o evento "Como falar em japonês"

Desenvolvedor Frontend 2:

Alterar formato dos elementos (Bordas, efeitos de hover ou tamanho)

Dica: evite mexer no mesmo lugar que seu colega de equipe para não causar conflitos no merge!

COMANDOS GIT



INICIALIZAÇÃO

GIT INIT

GIT CLONE URL

BRANCHES

GIT BRANCH

GIT CHECKOUT BRANCH

GIT BRANCH NOVA-BRANCH

GIT CHECKOUT -D BRANCH

ALTERAÇÃO

GIT STATUS

GIT ADD .

GIT COMMIT -M "MENSAGEM"

GIT COMMIT --AMEND

PUBLICAÇÃO

GIT FETCH

GIT PULL REMOTE BRANCH

GIT PUSH REMOTE BRANCH

MESCLAGEM

GIT MERGE BRANCH

GIT REBASE BRANCH

GIT REBASE --ABORT

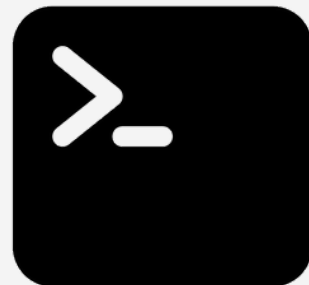
GIT REBASE --CONTINUE

DESFAZER

GIT RESET --HARD HEAD

GIT RESET --SOFT HEAD

GIT REVERT COMMIT



Contatos

Agradecimentos especiais ao professor
Marco Rojas por fazer o papel de mediador!



Felipe Furtado Róza

@fepisle

github.com/FelipeFurtadoRoza

linkedin.com/in/felipe-furtado-roza



Gustavo Bonifácio Conceição

@bonifacio.gusta

github.com/GustavoBonif

linkedin.com/in/gustavo-bonifacio



Gustavo José Baierski

@gbaierski

github.com/gbaierski

linkedin.com/in/gustavo-baierski



Vinícius Acordi Soethe

@thesoethe

github.com/ViniciusAcordiSoethe

linkedin.com/in/viniciusacordisoethe/

FIM! :))