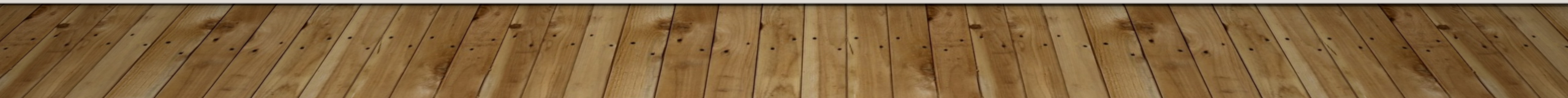


ESTRUTURA DE DADOS



AGENDA

- Introdução ao GIT

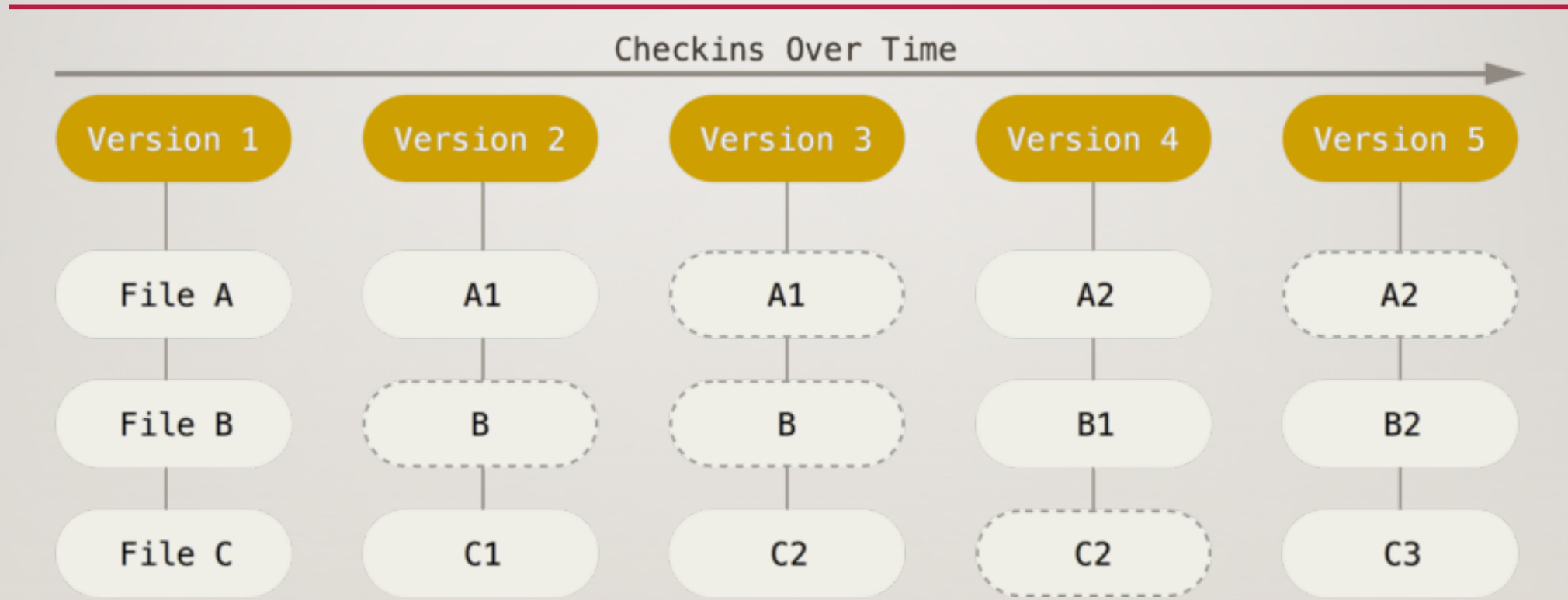
DESAFIO

- **Realizar com sucesso todas as atividades de sala e utilizar o github para armazená-las.**
 - **OBS:As atividades podem ser terminadas em casa, caso o tempo em sala não seja suficiente.**

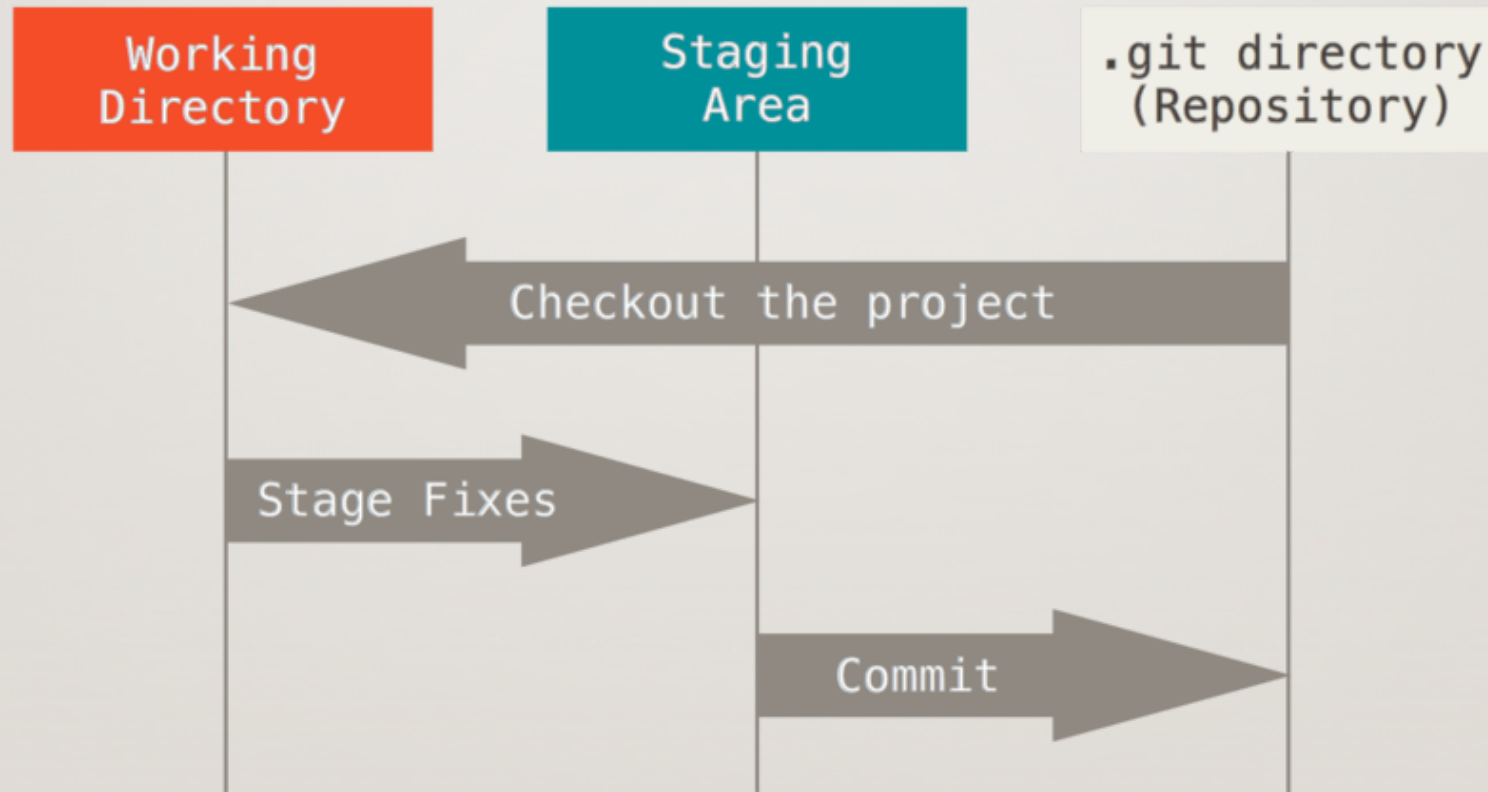


1 ponto adicional na prova com menor nota

GIT

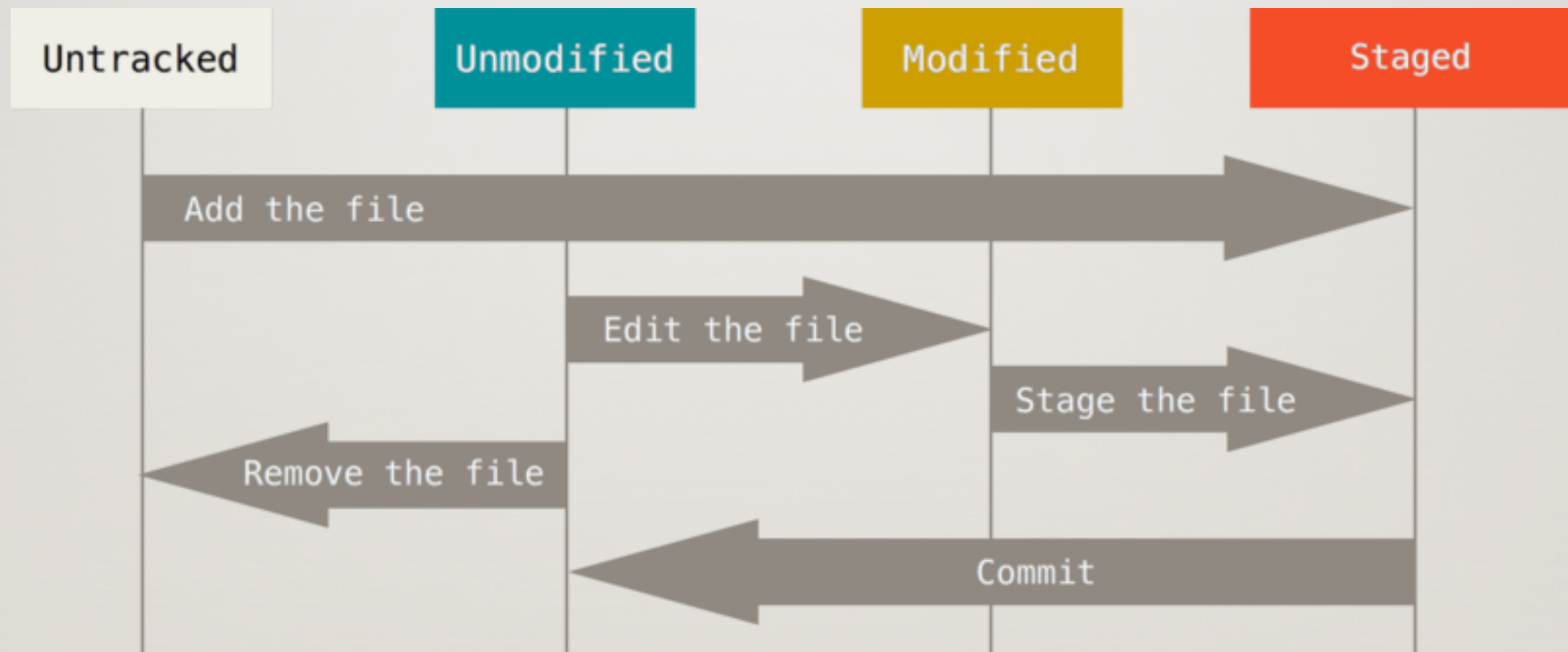


GIT



GIT

- Modificar no diretório de trabalho (modified)
- Indexar (especificar quem vai para stage área) (staged)
- "Comitar" (committed)



GIT – PÓS INSTALAÇÃO

- `git config --global user.name "John Doe"`
- `git config --global user.email johndoe@example.com`
- Se desejar mudar o editor: `git config --global core.editor emacs` ou `git config --global core.editor "'C:/Program Files (x86)/Notepad++/notepad++.exe' -multiInst -nosession"` (windows)
- Settings: `git config --list` ou `git config user.name`
- Git help. Ex: `git help config`

GIT – COMANDOS BÁSICOS

- Iniciando um repositório: `git init` (no diretório do repositório) e `git clone` (repo remoto)
- Adicionando ao index:
 - `git add .`
 - `git add *.c`
 - `git add README.md`
- Clonando: `git clone https://github.com/libgit2/libgit2 <outro dir>`
- Verificando situação dos arquivos: `git status`
- Ignorando arquivos: `.gitignore`
- `git diff <--staged>` / `git rm` / `git mv`
- Commit: `git commit`, `git push`, `git pull`
- Enviando e buscando: `git push` e `git pull`

ATIVIDADE

- Clonar seu repositório
- Buscar no google um template gitignore
- Criar um arquivo .gitignore se não existir
- Criar um programa "Hello World"
- Compilar, testar
- Adicionar ao index (stage)
- Commitar, submeter ao github e verificar no site

ATIVIDADE

- Clonar o projeto em um novo diretório
- Modificar a mensagem do programa
- Verificar o status do repositório
- Add, commit, push
- Verificar as mudanças no site
- Voltar ao diretório inicial
- Git pull