Guidelines

**Projeto** - <https://github.com/victorroatt/engenharia-de-software>

Clone

Para que a equipe consiga começar a trabalhar no projeto é necessário que eles efetuem o clone do projeto em suas próprias máquinas, para isso devem executar o seguinte comando no local onde o repositório deve ser clonado.

git clone <https://github.com/victorroatt/engenharia-de-software.git>

Branch

Para evitar problemas na versão main, é necessário criar um branch para que torne mais fácil a modificação dos arquivos. Deve ser executado o seguinte comando na raiz do repositório.

git branch “nome do branch”

git branch criar-carrinho

Checkout

Para navegar para o novo branch criado é necessário utilizar o comando checkout, dessa forma utilizada o comando abaixo.

git checkout criar-carrinho

Push

Após terem sido feitas alterações no novo branch, no final do dia sempre devemos fazer um push das alterações feitas. Primeiramente é necessário adcionar todas as alterações para tornarem “staged”, com o seguinte comando

git add .

Agora que elas estão staged nós podemos criar uma captura do branch local através do comando commit. É necessário que sempre que o comitte seja efetuado que uma mensagem seja escrita para identificação da equipe. Usamos o seguinte comando:

git commit -m “adicionado novo.txt”

Por ultimo podemos finalmente efetuar o push do branch local para o branch remoto que toda equipe está trabalhando, com o seguinte comando.

git push origin criar-carrinho

Merge

Digamos que o branch criado para criar o carrinho tenha sido concluido e está funcionando corretamente no projeto atual. A próxima etapa então é fazer o merge do branch criar-carrinho no main branch. Primeiramente nós fazemos um checkout no main branch.

git checkout main

Após isso executamos o comando de realizar o merge do branch criar-carrinho.

git merge criar-carrinho

E então fazemos um pull para sincronizar todas informações com o main, com o comando,

git push

Pull

É importante executar o pull todo dia no começo do trabalho para pegar os últimos arquivos do github atualizados para trabalhar nos arquivos mais novos. Pode ser que conflitos sejam gerados com as duas diferentes versões de arquivos, esses conflitos precisam ser corrigidos manualmente. Assim rodamos o seguinte comando para pegar todos os arquivos.

git pull

Diff

O comando diff é utilizado para comparar fontes de dados do git, que podem ser commits, ramificações, arquivos e outros. Ou seja, ele compara mudanças locais, por exemplo, com a versão do branch em que você está, e devolve no terminal todas as diferenças existentes entre ambos.

git diff