



Apresentação Final de Engenharia de Software

Universidade Federal do Tocantins _ Câmpus Palmas (06/12/2023)

Disciplina: Engenharia de Software 2023/02

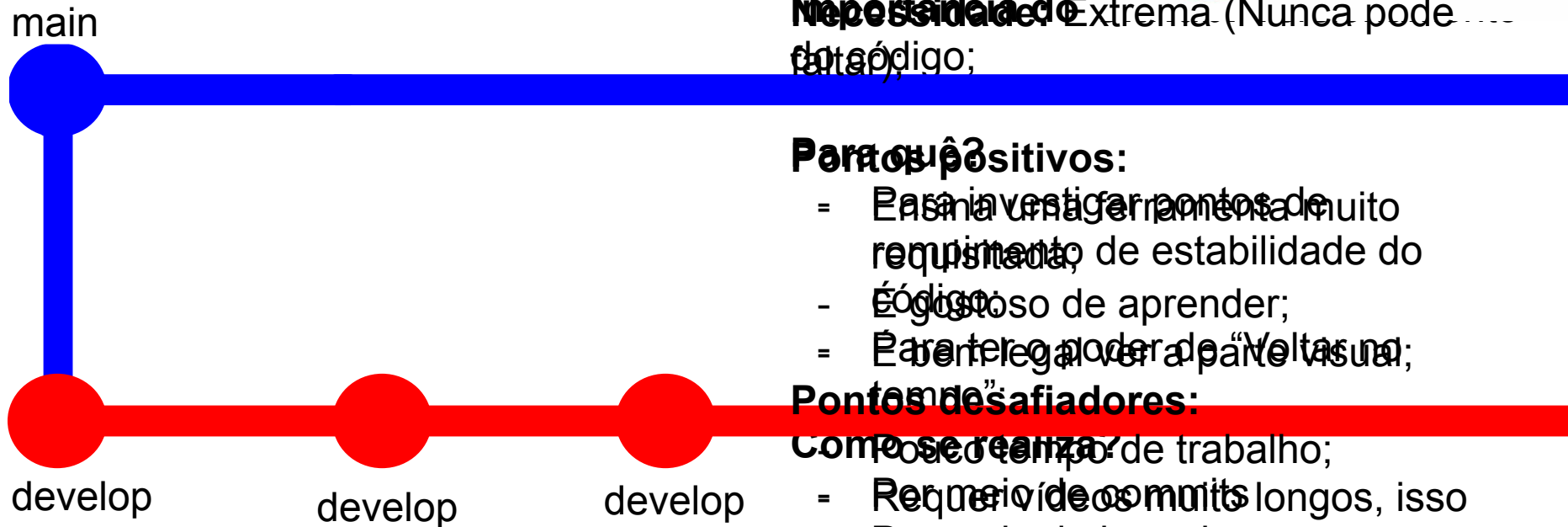
Orientador: Edeilson Milhomem

Discentes:

- Luís Felipe Krause de Castro
- Luiz Filipe de Souza Alves
- Douglas Ribeiro Santos de Sá
- Patryck Henryck Moreira Silva
- Gabryel Soares Marques
- Rafael de Oliveira Ribeiro
- Cássio Coutinho Lima



1º Parte - Aprendendo Git e Github



Impressão de Extrema (Nunca pode ...
do código;

Pontos positivos:

- = Para investigar pontos de muito requisição de estabilidade do código;
- É fácil de aprender;
- = Para ter o poder de "Voltar no tempo".

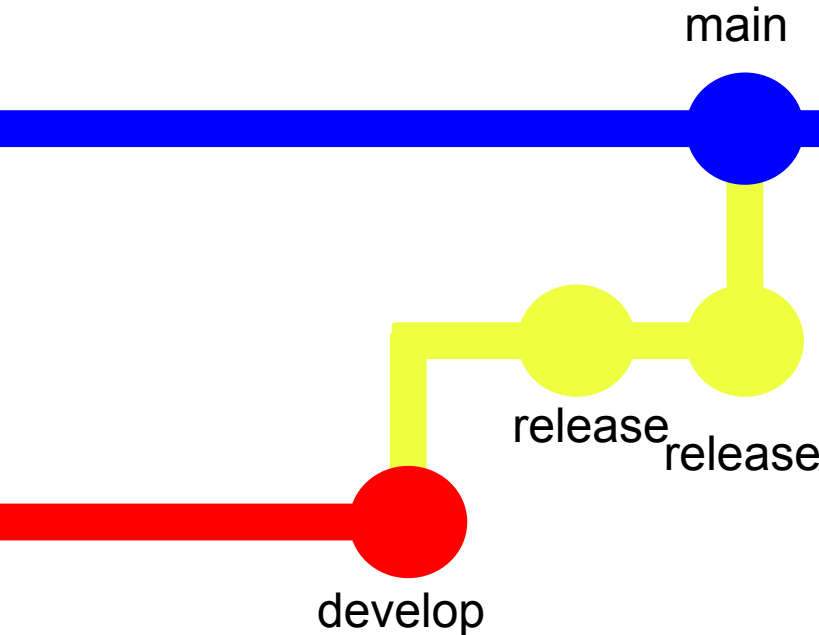
Pontos desafiadores:

Como se realiza?
Pode ser realizado de trabalho;

- = Requer vídeos curtos longos, isso
- Para a maneira mais rápida;

\$ git add ...
"git add" é o comando para commitar o código. "git commit" é o comando para criar um novo commit. "git push" é o comando para enviar o código para o repositório remoto. "git pull" é o comando para puxar o código do repositório remoto. "git clone" é o comando para clonar um repositório. "git checkout" é o comando para alternar entre branches. "git merge" é o comando para fundir um branch no outro. "git revert" é o comando para reverter uma alteração. "git reset" é o comando para reverter o estado do repositório. "git rm" é o comando para remover arquivos. "git mv" é o comando para mover arquivos. "git log" é o comando para visualizar o histórico de commits. "git diff" é o comando para visualizar as diferenças entre dois commits. "git status" é o comando para verificar o estado do repositório. "git clean" é o comando para remover arquivos não rastreados. "git fetch" é o comando para baixar o código do repositório remoto sem fundir. "git reset --hard" é o comando para reverter o estado do repositório para o commit especificado. "git checkout --orphan" é o comando para criar um novo branch a partir de um commit específico. "git branch --delete" é o comando para deletar um branch. "git branch --list" é o comando para listar todos os branches. "git branch --show-current" é o comando para mostrar o branch atual. "git branch --create" é o comando para criar um novo branch. "git branch --move" é o comando para mover um branch. "git branch --rebase" é o comando para rebasear um branch. "git branch --track" é o comando para criar um branch e rastreá-lo para um branch remoto. "git branch --untrack" é o comando para desassociar um branch de um branch remoto. "git branch --set-upstream" é o comando para definir o branch remoto para um branch. "git branch --unset-upstream" é o comando para remover o branch remoto de um branch. "git branch --format" é o comando para definir o formato do nome do branch. "git branch --color" é o comando para habilitar a cor no nome do branch. "git branch --no-color" é o comando para desabilitar a cor no nome do branch. "git branch --verbose" é o comando para mostrar informações adicionais sobre o branch. "git branch --help" é o comando para mostrar a ajuda do comando. "git branch" é o comando para gerenciar branches. "git" é o sistema de controle de versão. "git add" é o comando para adicionar arquivos ao índice. "git commit" é o comando para salvar as alterações no repositório local. "git push" é o comando para enviar as alterações para o repositório remoto. "git pull" é o comando para baixar as alterações do repositório remoto e fundir no branch atual. "git clone" é o comando para criar uma cópia local de um repositório remoto. "git checkout" é o comando para alternar entre branches. "git merge" é o comando para fundir um branch no outro. "git revert" é o comando para reverter uma alteração. "git reset" é o comando para reverter o estado do repositório. "git rm" é o comando para remover arquivos. "git mv" é o comando para mover arquivos. "git log" é o comando para visualizar o histórico de commits. "git diff" é o comando para visualizar as diferenças entre dois commits. "git status" é o comando para verificar o estado do repositório. "git clean" é o comando para remover arquivos não rastreados. "git fetch" é o comando para baixar o código do repositório remoto sem fundir. "git reset --hard" é o comando para reverter o estado do repositório para o commit especificado. "git checkout --orphan" é o comando para criar um novo branch a partir de um commit específico. "git branch --delete" é o comando para deletar um branch. "git branch --list" é o comando para listar todos os branches. "git branch --show-current" é o comando para mostrar o branch atual. "git branch --create" é o comando para criar um novo branch. "git branch --move" é o comando para mover um branch. "git branch --rebase" é o comando para rebasear um branch. "git branch --track" é o comando para criar um branch e rastreá-lo para um branch remoto. "git branch --untrack" é o comando para desassociar um branch de um branch remoto. "git branch --set-upstream" é o comando para definir o branch remoto para um branch. "git branch --unset-upstream" é o comando para remover o branch remoto de um branch. "git branch --format" é o comando para definir o formato do nome do branch. "git branch --color" é o comando para habilitar a cor no nome do branch. "git branch --no-color" é o comando para desabilitar a cor no nome do branch. "git branch --verbose" é o comando para mostrar informações adicionais sobre o branch. "git branch --help" é o comando para mostrar a ajuda do comando. "git branch" é o comando para gerenciar branches. "git" é o sistema de controle de versão.

2º Parte - Gitflow, Release



Release do Produto

Pontos positivos:

- Separar o produto em pontos positivos

Pontos negativos:

- Cada mudança de versão

Pontos positivos:

- Pode ser trabalhoso

Pontos negativos:

- Às vezes o trabalho é menor que a

Pontos positivos:

- Pode gerar uma numeração muito grande

Pontos negativos:

- Às vezes o trabalho é menor que a

Pontos positivos:

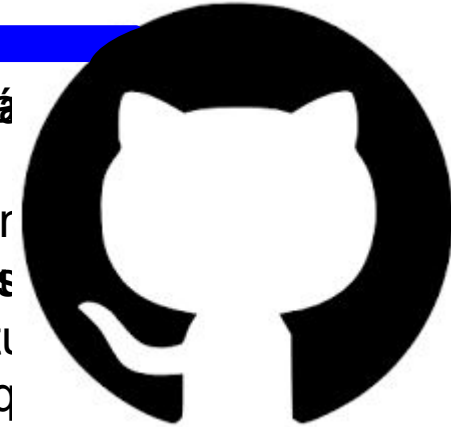
- Pode gerar uma numeração muito grande

Pontos negativos:

- Às vezes o trabalho é menor que a

Pontos positivos:

- Pode gerar uma numeração muito grande



\$ git push origin --tags "Gitflow é uma maneira de trabalhar em equipe"



3º Parte - Trabalho em equipe



1. Lições:

1. Validação de requisitos:

- Colaboração efetiva;
- Adaptação a mudanças;
- Entendimento profundo;
- Melhora na comunicação;
- Prototipagem rápida;

2. Desafios:

- Documentação clara;
- Iteração contínua;
- Pressão por prazos;

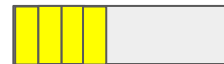
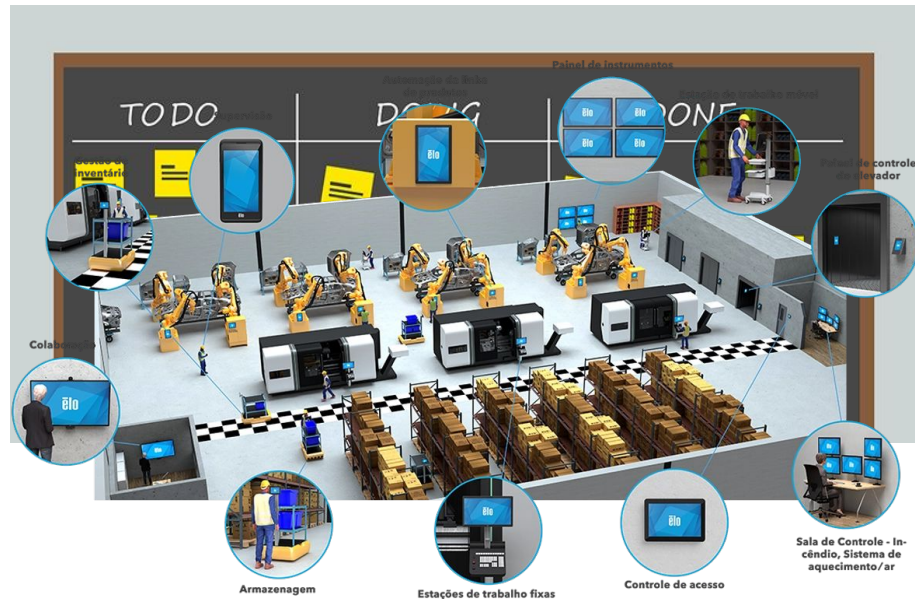
2. Validação das tecnologias:

- Gerenciamento de conflitos;
- Manter documentação;
- Análise de requisitos técnicos;
- Complexidade no início;
- Pesquisa e avaliação;
- Possíveis conflitos de merge;
- Consultas a especialistas;

3. Validação incremental:

3. Experiência:

- GitHub como plataforma de colaboração;
- GitFlow para gerenciamento de branches;
- Colaboração descentralizada;

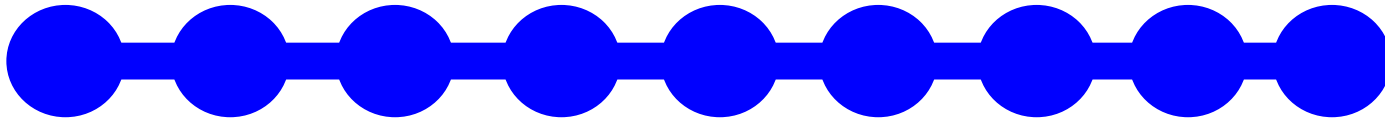




4º Parte - Visão Geral

Os padrões em desenvolvimento de Software, como o Model-View-Controller (MVC), são fundamentais para o sucesso de um projeto. Eles fornecem uma estrutura organizada para o desenvolvimento de sistemas.

- Organização e Manutenção
- Reaplicação do Código
- Testes Incrementais
- Padronização da Equipe

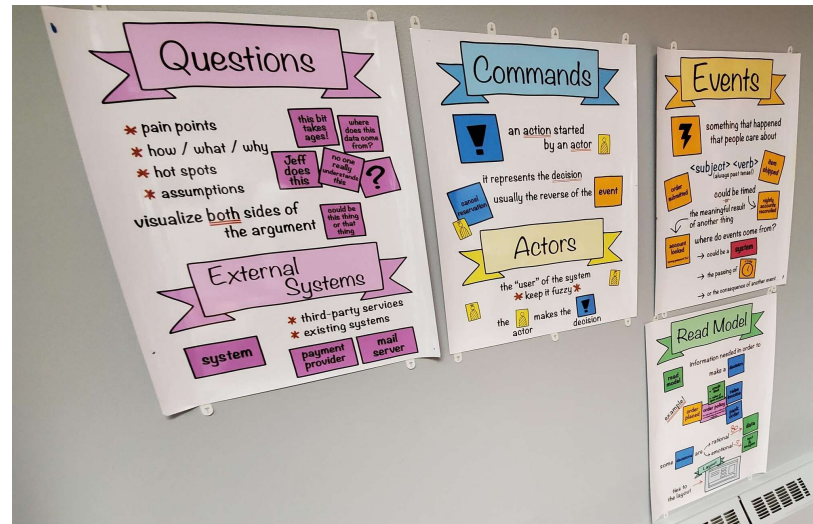


6º Parte - Produto



Retornos proibitórios

- No primeiro momento os grupos foram divididos em equipes de 4 a 6 pessoas para discutir e definir o problema a ser estudado e a necessidade de um sistema de informação para resolver o problema.
- Stakeholders segundo momento depois de definir o problema e a necessidade de um sistema de informação, os grupos foram orientados a identificar os stakeholders envolvidos no processo de desenvolvimento do sistema.
- O terceiro momento foi a realização de um brainstorming para identificar as funcionalidades do sistema, considerando as necessidades dos stakeholders.



5º Parte - Disciplina



Testes Unitários

Qual a importância de Testes Unitários em projetos de Software?

☐ Qual a importância de Testes Unitários em projetos de Software?

Identificação de Erros Precocemente

Facilitação da Manutenção
código (funções, métodos, ou classes)
estejam funcionando corretamente,


Melhoria na Qualidade do Código
precocemente para o máximo de casos
de testes possíveis

Redução de Custos a Longo Prazo



7º Parte - Produto



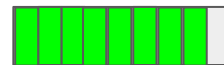


CPF:

Senha:

[Entrar](#)

[Esqueci a Senha](#)



Final



É isso, acabou?

- Acabou!
- Então tchau brigado!

