Módulo 5

BackEnd Java

Rodrigo Pires

1 Conhecendo o GIT

Conhecendo o GIT

Conhecendo o GIT

Antes de falar de GIT:

Você está trabalhando em um projeto com mais 3 desenvolvedores ao mesmo tempo e precisa repartir as novas funcionalidades do sistema com eles. Como você faz a junção do código ao final do desenvolvimento? (Lembrando que o mesmo arquivo pode ser editado por mais de um dev).

Texto1

Conhecendo o GIT

Antes de falar de GIT:

Seu chefe pediu para você deletar uma funcionalidade do sistema que não é utilizada. Após 3 meses ele decidiu que quer essa funcionalidade de volta. Como você faz para recuperar essa funcionalidade e implementá-la no projeto mesmo após muitas outras mudanças?

Texto 2

Conhecendo o GIT

Antes de falar de GIT:

Você é muito otimista e por isso não possui nenhum sistema de backup automático da sua aplicação em ambiente de desenvolvimento. Um belo dia seu computador queima e você não havia copiado suas últimas features para um pendrive. Como você faz para recuperar esse código?

Conhecendo o GIT

O que é GIT ?

O Git é um sistema open-source de controle de versão utilizado pela grande maioria dos desenvolvedores atualmente. Com ele podemos criar todo histórico de alterações no código do nosso projeto e facilmente voltar para qualquer ponto para saber como o código estava naquela data.

Além disso, o Git nos ajuda muito a controlar o fluxo de novas funcionalidades entre vários desenvolvedores no mesmo projeto com ferramentas para análise e resolução de conflitos quando o mesmo arquivo é editado por mais de uma pessoa em funcionalidades diferentes.

Texto 4

Conhecendo o GIT

Composição do GIT ?

- Repositórios
- Pontos na história - Commits
- Ramificações - Branchs

Conhecendo o GIT

Repositórios

Os repositórios são os ambientes criados para armazenar seus códigos. Você pode possuir um ou mais repositórios, públicos ou privados, locais ou remotos, e eles podem armazenar não somente os próprios códigos a serem modificados, mas também imagens, áudios, arquivos e outros elementos relacionados ao seu projeto.

Texto 6

Conhecendo o GIT

Pontos na história - Commits

Tudo no Git é movido através dos pontos na história do projeto que são chamados de commits, esses pontos são formados por conjuntos de alterações em um ou mais arquivos e somados a um descritivo que resume as alterações nesse ponto.

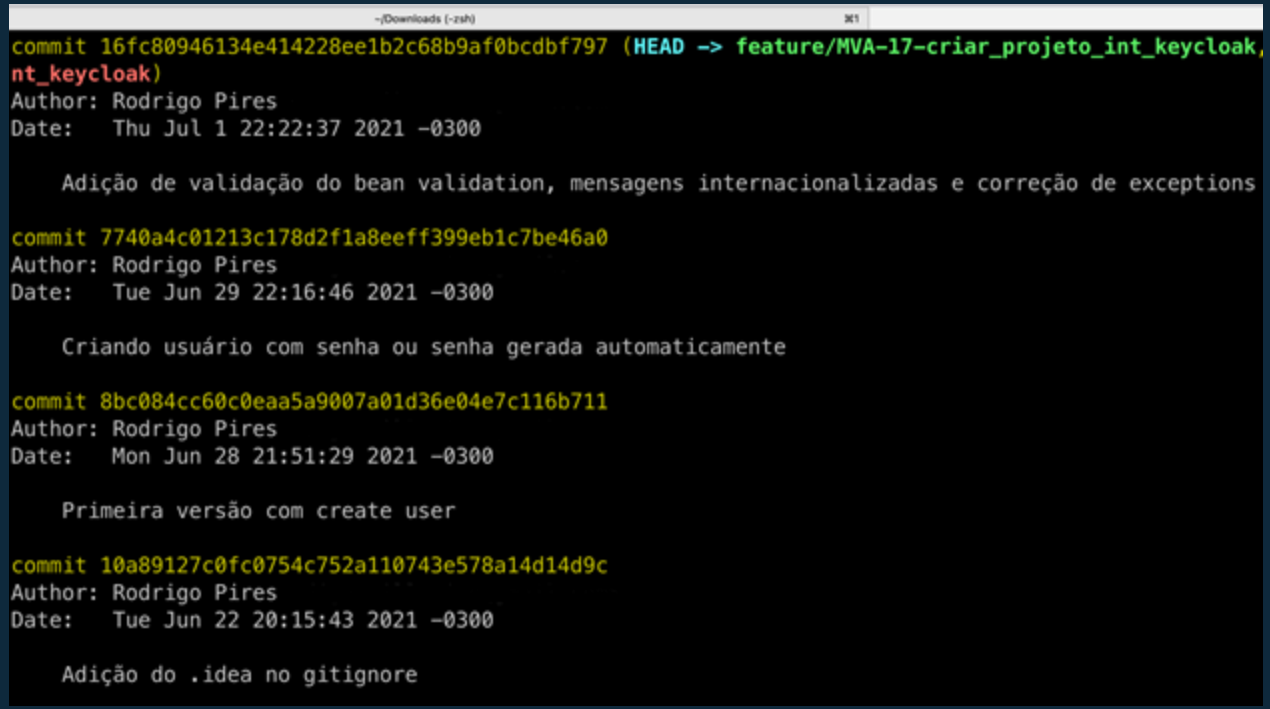
Conhecendo o GIT

Pontos na história - Commits

1. Configuração da estrutura do projeto
2. Estrutura da página de login
3. Estilos CSS da página de login
4. Estrutura da página de cadastro
5. Resolvido problema no login
6. Estilos CSS da página de cadastro

Conhecendo o GIT

Pontos na história - Commits



Conhecendo o GIT

Ramificações - Branchs

Imagine que você esteja trabalhando no meio de uma grande funcionalidade, que pode levar até 2 meses para terminá-la. Em uma bela manhã de sol seu chefe resolve pedir urgentemente uma alteração na versão em produção da aplicação, ou seja, você não pode utilizar o código em que está trabalhando pois o mesmo possui features inacabadas. Como resolver?

Texto 9

Conhecendo o GIT

Ramificações - Branchs

As ramificações ou branchs no Git são formas de termos uma mesma versão do código sofrendo alterações e recebendo commits de diferentes fontes e inclusive por diferentes desenvolvedores.

Dessa forma, nós podemos manter um ramo para nossa funcionalidade que irá levar mais tempo e trabalhar em outro branch com a versão em produção para realizar alterações mais urgentes. E fica tranquilo, no fim de tudo o Git ainda vai nos ajudar a unir os códigos desses dois ramos de forma muito simpática.

Conhecendo o GIT

- Ramificações - Branchs

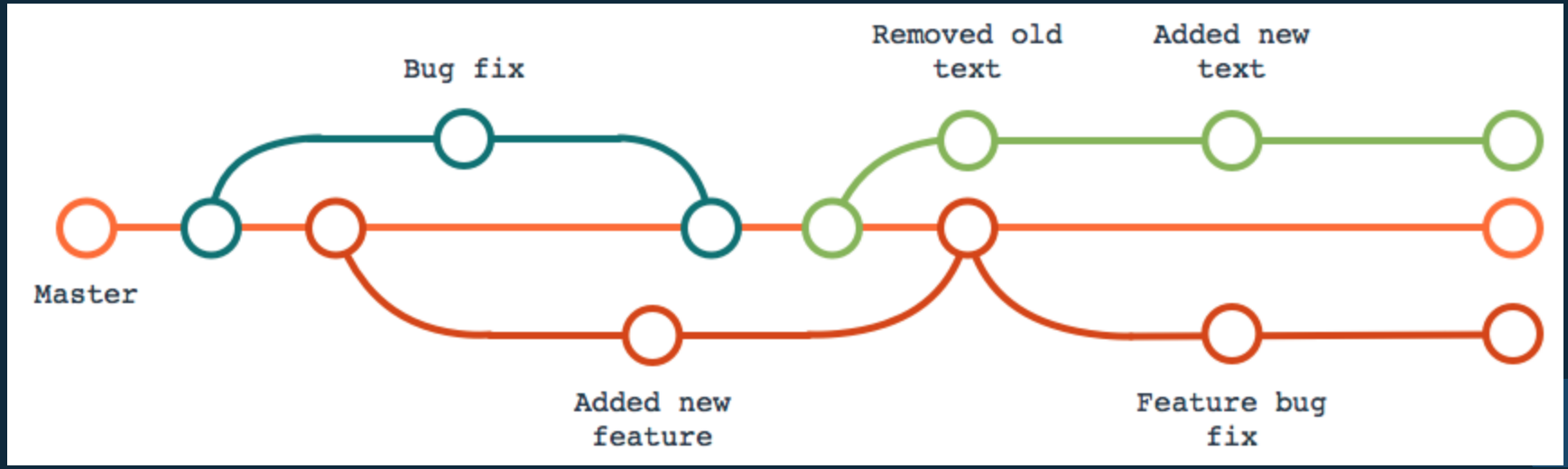


Imagem1

Conhecendo o GIT

Comandos básicos do GIT:

- **init:** este comando dá origem a um repositório novo, local ou remoto, ou reinicializar um repositório já existente;
- **clone:** este comando clona o código de um repositório para sua manipulação em outro ambiente;
- **commit:** este comando move os arquivos da state area para um repositório local;
- **add:** este comando adiciona um arquivo alterado a uma staging area, ou seja, o prepara para ser vinculado a um commit;

Texto 10

Conhecendo o GIT

Comandos básicos do GIT:

- **push**: este comando envia arquivos de um repositório local para um repositório remoto. No GitHub, por exemplo;
- **pull**: ao contrário do push, este comando traz um arquivo do repositório remoto para o repositório local.
- **checkout:** este comando é utilizado para trocar de branchs ou voltar um arquivo alterado para o seu estado anterior.
- **merge**: este comando serve para unir arquivos alterados ao arquivo original de um projeto. Em outras palavras, é ele quem une os branchs as commits.

Conhecendo o GIT

Comandos básicos do GIT:

- **status**: este comando mostra o status atual do repositório.
- **log**: este comando permite a visualização do histórico de commits de um arquivo ou usuário, ou o acesso de uma versão específica.

Conhecendo o GIT

Comandos avançados GIT:

- ➔ rebase: Igual ao comando Merge, mas pegando todos os commits confirmados em outra branch e colocando em baixo de todos os commits da branch atual.
- ➔ config --list: Mostra algumas configurações do repositório
- ➔ git config -–global user.name: "Meu Nome" » Define o nome de usuário para o Git
- ➔ git config -–global user.email: "email@dominio.com" » Define o e-mail de usuário para o Git (tem de ser o cadastrado no GitHub)

Conhecendo o GIT

Comandos avançados GIT:

- **squash**: Adiciona vários commits em apenas um só
- **cherry pick:** Copia um commit de uma branch para a sua branch
- **reset:** Comando que volta o commit que está na área de staging e volta para a working area.
- **reset --hard HEAD~**: Volta o arquivo para o estado original, quando o mesmo já feito commit.

Conhecendo o GitHub

GitHub - https://github.com/

Legal, até agora falamos sobre algumas funcionalidades do Git mas tem um grande problema aí: como os outros desenvolvedores do time terão acesso a todo esse código e poderão também adicionar seus branchs e commits?

O Github é um serviço online de hospedagem de repositórios Git (como são chamados os projetos que utilizam Git). Com ele podemos manter todos nossos commits e ramos sincronizados entre os membros do time.

Além de servir como hospedagem, o Github possui muitas integrações com serviços que auxiliam no deploy da aplicação através de integração contínua.

Conhecendo o GitHub

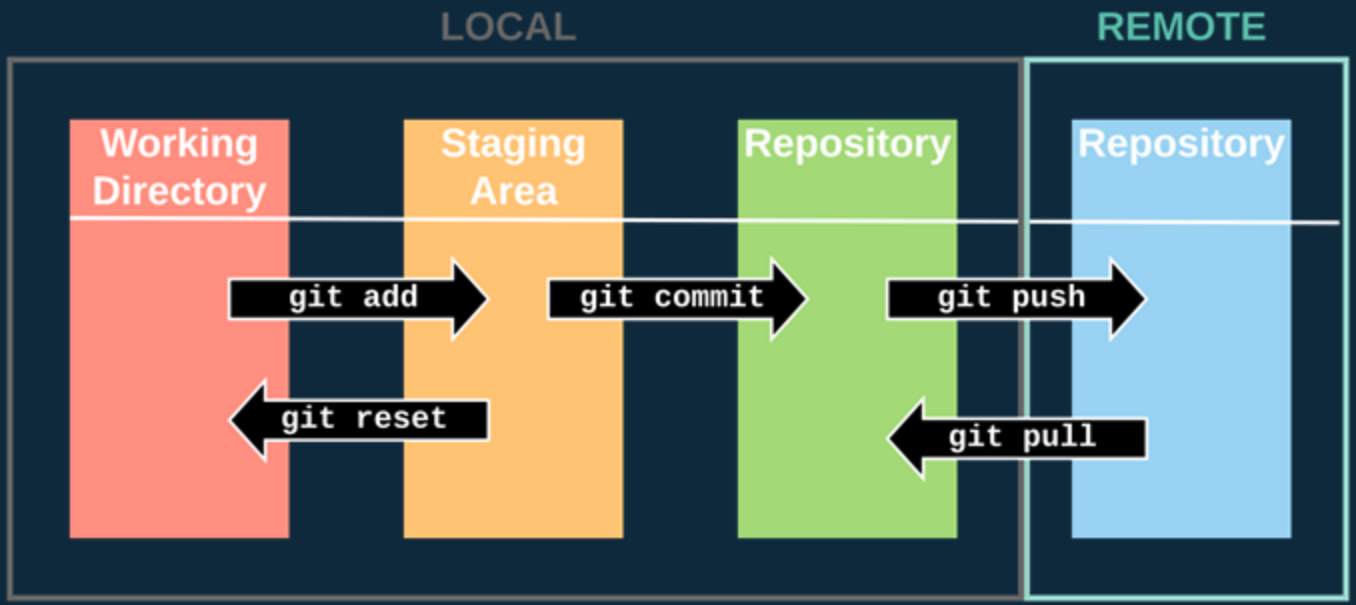


Imagem 2

Referências

- **Textos 1,2,3,4,5,6,7,8,9,10 e Imagem 1:**
  - https://blog.rocketseat.com.br/iniciando-com-git-github/