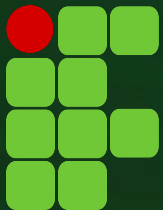


Minicurso de Programação Híbrida com PWA + Vue.js

Mário Carvalho

mariodearaujocarvalho@gmail.com

19 de outubro de 2018



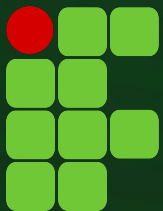
Versionamento de código avançado

Módulo 01 - Aula 02

Mário Carvalho

mariodearaujocarvalho@gmail.com

5 de outubro de 2018



Roteiro

- Terminologias
- Configurando Nossa Identidade
- Verificando Nossas Configurações
- Explorando a Documentação de Ajuda
- Obtendo Um Repositório Git
- Inicializando Um Repositório Em Um Diretório Existente
- Clonando Um Repositório
- Gravando Alterações no Repositório
- Verificando o Status de Seus Arquivos

Roteiro

- **Adicionando Novos Arquivos**
- **Fazendo Commit de Suas Mudanças**
- **Visualizando os Logs**
- **Trabalhando Com Repositórios Remotos**
- **Pushing para Seus Remotos**
- **Atualizando Seus Repositórios**
- **Livros Disponibilizados Gratuitos**
- **Projeto Prático: Pondo em prática os comandos mencionados.**

Objetivo



Apresentar novas funções e configurações da tecnologia Git. Trabalhar com Markdown para futuras documentações de código.



Repositório do minicurso

Git do minicurso

- Todas as apresentações, códigos, referências e demais arquivos do minicurso vão ser colocados para download no Github em um repositório do minicurso:
- <https://github.com/MarioDeAraujoCarvalho/mini-programming-course-hybrid-with-pwa-with-vuejs-ufms-2018-2>

Termos frequentes de um repositório

Git

Terminologias

- **master**: branch principal do repositório.
- **clone**: cópia de um repositório git existente, para o seu ambiente local.
- **commit**: envio de arquivos para o repositório local.
- **fetch ou pull**: pega as últimas atualizações de um repositório remoto. A diferença entre fetch e pull é que pull combina as duas coisas: pega as atualizações de código do repositório remoto e efetua sua mesclagem com o repositório local.

•

Termos frequentes de um repositório

Git

Terminologias

- **push**: nós utilizamos para enviar os nossos arquivos para um repositório remoto.
- **remote**: são os locais remotos do seu repositório, normalmente em algum servidor central, como o GitHub.
- **branch**: é uma branch no Git nada mais é do que um rótulo particular em determinado node.

Configurando Nossa Identidade

Terminal/CMD

- Abra o seu terminal de comandos



Configurando Nossa Identidade

git config

- A primeira coisa que devemos fazer quando instalar o Git é definir o nosso nome de usuário e endereço de e-mail. Isso é importante porque todos os commits no Git utilizam essas informações, e está imutavelmente anexado nos commits que realizamos:
- `bash`
- `$ git config --global user.name "Mário Carvalho"`
- `$ git config --global user.email mario@email.com`

Verificando nossas configurações

git config --list

- Para verificarmos se nossas credenciais estão corretas, vamos usar o seguinte comando.
- bash
- *\$ git config --list*
- result
- *user.name=Mário Carvalho*
- *user.email=schacon@gmail.com*

Explorando a documentação de ajuda

git help

- Caso você precise de ajuda usando o Git, existem três formas de se obter ajuda das páginas de manual (manpage) para quaisquer comandos do Git:
- bash
- *\$ git help <verb>*
- Por exemplo:
- *\$ git help config*
- result
- *Documentação detalhada do comando...*

Obtendo um Repositório Git

git init e clone

- Você pode obter um projeto Git utilizando duas formas principais. A primeira faz uso de um projeto ou diretório existente e o importa para o Git. A segunda **clona** um repositório Git existente a partir de outro servidor.

Inicializando um Repositório

git init

- Inicializando um Repositório em um Diretório Existente.
- bash
- *\$ mkdir my-directory-repository*
- *\$ cd my-directory-repository*
- *\$ git init*

Clonando um Repositório

git clone

- Você clona um repositório com `git clone [url]`. é importante você saber o link correto. Por exemplo:
- `bash`
- *`$ git clone git://github.com/johndoe/hello-git.git`*

Verificando o Status de Seus Arquivos

git status

- A principal ferramenta utilizada para determinar quais arquivos estão em quais estados é o comando `git status`. Se você executar este comando diretamente após uma clonagem, você deverá ver algo similar a isso:
- `bash`
- *`$ git status`*

Adicionando Novos Arquivos

git status

- Para passar a monitorar um novo arquivo, use o comando `git add`. Para monitorar o arquivo README, você pode rodar isso:
- `bash`
- *`$ git add .`*

Fazendo Commit de Suas Mudanças

git push

- Agora que a sua área de seleção está do jeito que você quer, você pode fazer o commit de suas mudanças. Lembre-se que tudo aquilo que ainda não foi selecionado — qualquer arquivo que você criou ou modificou que você não tenha rodado o comando `git add` desde que editou .
- `bash`
- *`$ git commit -a -m "Texto do meu commit"`*
- *`$ git push`*

Gravando Alterações no Repositório

git commit

- Você tem um repositório Git com seus arquivos para esse projeto. Você precisa fazer algumas mudanças e fazer o commit das partes destas mudanças em seu repositório cada vez que o projeto atinge um estado no qual você queira gravar.
- bash
- *\$ git add .*
- *\$ git commit -a -m "Minha mensagem"*
- *\$ git push origin master*

Visualizando os logs

git log

- Depois que você tiver criado vários commits, ou se clonou um repositório com um histórico de commits existente, você provavelmente vai querer ver o que aconteceu.
- bash
- *\$ git log*

Trabalhando com Remotos

git clone

- Para ser capaz de colaborar com qualquer projeto no Git, você precisa saber como gerenciar seus repositórios remotos.
- bash
- *\$ git clone git://github.com/johndoe/hello-git.git*
- *\$ cd hello*

Pushing Para Seus Remotos

git push

- Quando o seu projeto estiver pronto para ser compartilhado, você tem que enviá-lo para a fonte você pode rodar o comando abaixo para enviar o seu trabalho para o servidor:
- bash
- *\$ git add .*
- *\$ git commit -a -m "Minha mensagem"*
- *\$ git push origin master*

Atualizando seus repositórios

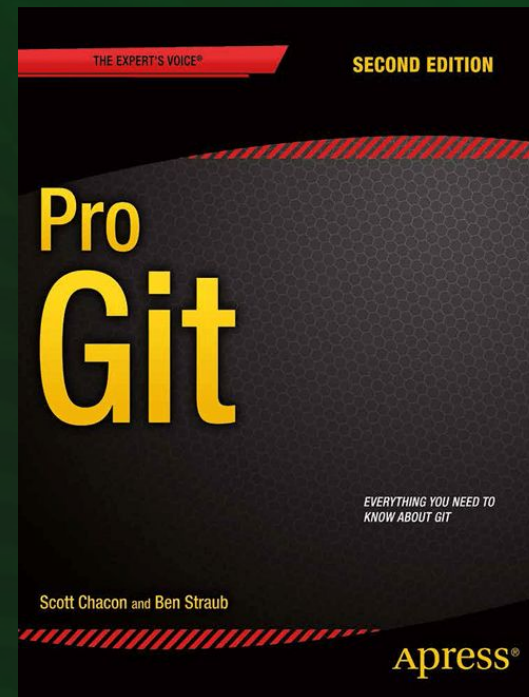
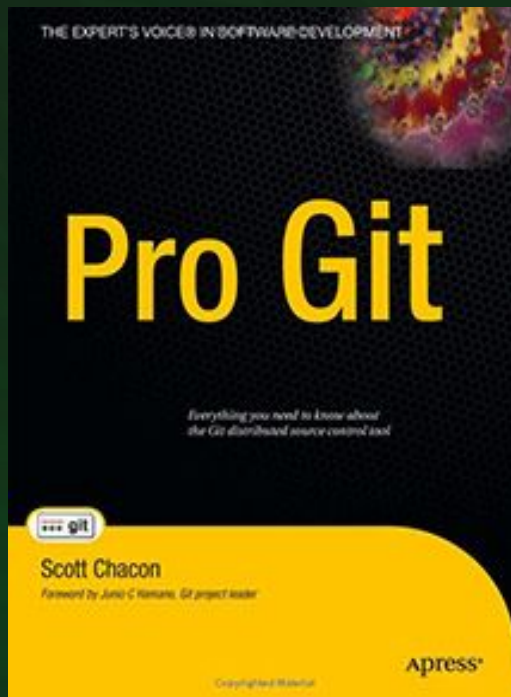
git pull

- Quando o seu projeto for mantido por mais de uma pessoa, é recomendado fazer pull toda vez que for fazer alterações no projeto. Navegue até seu repositório, abra o terminal dentro dela e digite:
- bash
- *\$ git pull*

Livros disponibilizados gratuitos

Links dos livros

- <https://git-scm.com/book/pt-br/v1>
- <https://git-scm.com/book/pt-br/v2>



Oque é Markdown?

Markdown - <https://markdownguide.org/>



Oque é Markdown?

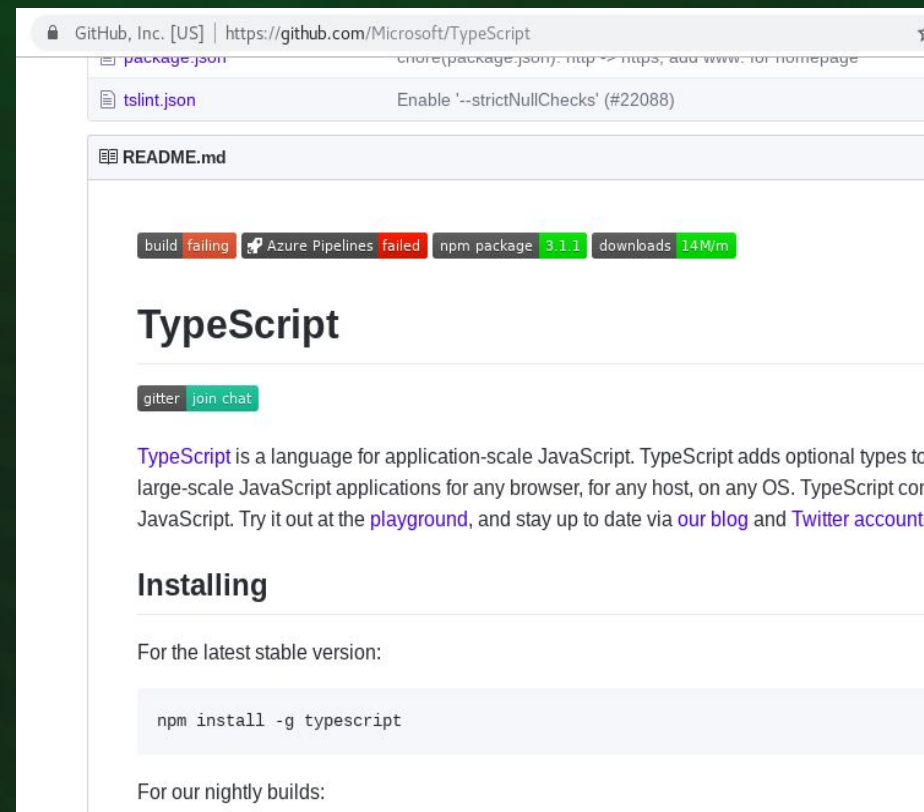
Uma forma inteligente de documentar

- **Markdown é uma linguagem simples de marcação originalmente criada por John Gruber e Aaron Swartz. Markdown converte seu texto em HTML válido.**

Virou um padrão de documentação de projetos

Result

```
'''hash
```



Projeto Prático

Criando, entendendo e versionando projetos.

- **Trabalhando com repositórios Git**
- **Entendendo e trabalhando com o Git.**
- **Trabalhando com Markdown**
- **Livros gratuitos de Git**

Tenha isso tatuado em seu coração

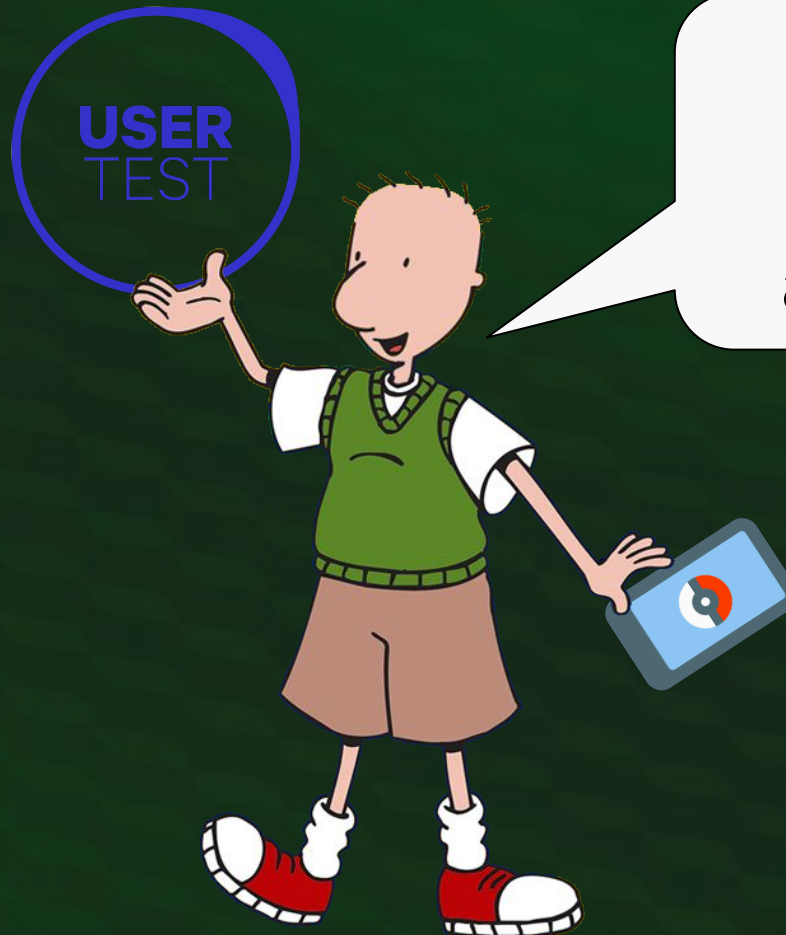
O mundo não é um
grande arco-íris...
... Uma prova na
sexta de manhã.

*Adaptado de Rudson Lima,
Live-O.*



Vamos à prática?

Explorando os conteúdos e desenvolvendo os projetos na prática.



**Alguém
ainda
acordado?**

Obrigado! Dúvidas?

Mário Carvalho
mariodearaujocarvalho@gmail.com
+55 (67) 99610-0530

