CONTROLE DE VERSÃO

Introdução a Ciências da Computação

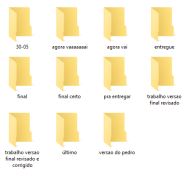
Atualizado em: 10 de junho de 2024

Iago Carvalho

Departamento de Ciência da Computação1

INTRODUÇÃO

Quando você está desenvolvendo um código, um algoritmo ou algum texto, normalmente são geradas diversas versões de um mesmo documento



2

ORGANIZAÇÃO

Como você organiza estas diversas versões de documentos?

Cópia por e-mail

Versão mais atualizada na nuvem

Substituição do arquivo mais novo

*. . .*

Será que isso é eficiente?

3

ALTERAÇÕES

Nas alternativas anteriores, é praticamente impossível ter um controle das alterações que cada arquivo sofreu com o tempo

O que foi alterado

Quando foi alterado

Porquê foi realizada tal alteração

Difícil organizar um trabalho feito só por uma pessoa. Imagine um trabalho feito por um grupo de 5 ou 6 pessoas

Diversas cópias de arquivo

Cada arquivo com uma alteração diferente

Como "juntar"tudo?

4

CONTROLE DE VERSÃO

Um Sistema de Controle de Versão (VCS, do inglês *Version Control System*) é uma ferramenta útil para rastrear facilmente alterações em seus documentos

Torna possível compartilhar arquivos

Facilita a colaboração entre diversas pessoas

Armazena o histórico de alterações dos arquivos

Hoje em dia são utilizados VCS distribuídos em nuvem

Sistemas baseados no protocolo Git

*◦* Github

*◦* Gitlab

5

CONTROLE DE VERSÃO E REPOSITÓRIOS

Em um VCS distribuído, seus projetos são repositórios 1

Todo repositório fica salvo em nuvem

Cada repositório contém todos os arquivos de seu projeto

Cada repositório também possui o histórico de seu projeto

As diferentes alterações realizadas

*◦* Por quem, quando e por quê

As várias versões de seu projeto

Desenvolvedores podem acessar e resgatar a última versão disponível e fazer uma cópia local

Pode-se trabalhar nesta cópia local

A cada alteração, é possível enviar novamente ao servidor e atualizar a versão do repositório

1Documentação sobre repositórios

6

FLUXO DE TRABALHO - CLONING

Quando um repositório é criado, ele é armazenado remotamente na nuvem

O Cloning (clonagem) é o ato de criar uma cópia local de um repositório para seu próprio dispositivo

Repositório na nuvem e local devem ser *sincronizados*

Facilita a adição, remoção e alteração de arquivos

Possibilita a utilização de qualquer interface gráfica de

desenvolvimento

7

FLUXO DE TRABALHO - COMMIT E PUSH

*Committing* e *pushing* são maneiras de adicionar as alterações feitas em sua máquina local ao repositório remoto

Desta forma, é possível compartilhar suas alterações com outras pessoas

Um commit é como se fosse um *checkpoint* que confirma alguma alteração realizada

É comum adicionarmos uma mensagem explicando este

*commit*

Registrados em sua máquina local

O push é a ação de enviar todos os *commits* para o servidor Finalmente, compartilhar seus arquivos

8

FLUXO DE TRABALHO - PULL

Um pull é o contrário de um *push*

Aqui nós estamos atualizando nossa máquina local com os arquivos mais atuais do servidor em nuvem

Normalmente, você faz um *clone* para "entrar"no repositório

Depois, sempre que for atualizar os arquivos em sua máquina local, você deve realizar *pulls*

9

GITHUB

O Github é o maior VCS distribuído da atualidade

Comprado pela Microsoft em 2018 por 7,5 bilhões de dólares Grátis para utilizar

*◦* Versão PRO com adicionais

*◦* Conta de estudante

10

INTRODUÇÃO A VCS

Nosso objetivo hoje é fazer com que todos tenham uma conta em algum VCS distribuído

A sugestão é que todos criem uma conta no Github

*◦* https://github.com/

Entretanto, qualquer outro VCS pode ser utilizado

*◦* Recomenda-se alternativas também baseadas em Git

*◦* Gitlab (https://gitlab.com)

Não recomenda-se a utilização de outros protocolos

SVN

Mercurial

Outros

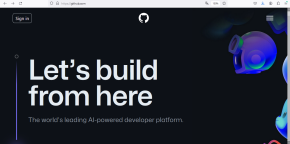
*◦* Veja outros aqui no Stack Overflow

11

CRIANDO UMA CONTA NO GITHUB

Nesta aula vamos focar no Github da Microsoft

Recomenda-se criar a conta com seu e-mail da UNIFAL

12

PERFIL NO GITHUB

Após criarem a conta, vocês terão seu perfil

13

GITHUB PAGES

O GitHub Pages é uma maneira simples e gratuita de criar e hospedar seu site na web

O nome de seu site será <usuario>.github.io

Para isto, você deve criar um novo repositório com o nome <usuario>.github.io

Este repositório deverá ser público

14

GITHUB PAGES

Este repositório poderar ser configurado como seu site pessoal Agora basta você criar o arquivo de entrada para seu site.

Por padrão, o GitHub Pages procurará um arquivo *index.html*, *index.md* ou *README.md* como o arquivo de entrada do seu site

15

ATIVIDADE DO DIA

Criar o repositório de seu site e incluir uma página *index.html*

Na próxima aula nós veremos conceitos um pouco mais avançados sobre o protocolo Git

16