

Desenvolvimento para WEB

Prof. José Rodolfo Beluzo

Sumário

- GIT
 - GITHUB

GIT - O que é?

- Sistema de controle de versões
- Trabalho em grupo no mesmo projeto
- Garantia de integridade dos arquivos criados
- Cada diretório de trabalho do Git é um repositório com um histórico completo e habilidade total de acompanhamento das revisões
- desenvolvido por Linus Torvalds

GIT - Instalação

- Para utilizar um sistema GIT, você deve instalar este nas máquinas as quais irá utilizar o sistema
- Iremos utilizar um sistema “Portável”:
 - <https://git-scm.com/download/win>
 - Realizem a instalação no pendrive

GITHUB - O que é?

- Hub (traduzido do Inglês, "pivô") ou concentrador é o processo pelo qual se transmite ou difunde determinada informação, tendo, como principal característica, que a mesma informação está sendo enviada para muitos receptores ao mesmo tempo (broadcast). Este termo é utilizado em rádio, telecomunicações e em informática.

Dean, Tamara (2010). Network+ Guide to Networks. [S.l.]: Delmar. pp. 256–257

- Serviço WEB que oferece as funcionalidades GIT
- <http://www.github.com>
 - Fazer seu cadastro...

GIT - Principais comandos

Inicialmente, criar um repositório no GITHUB

- Acessar o Git BASH
- Escolher a pasta que irá vincular seu repositório local ao GITHUB
 - (para navegar pelas pastas, digite: `cd nome_pasta`)

CONFIGURAÇÕES INICIAIS

- `$ git config --global user.name "YOUR NAME"`
`$ git config --global user.email "YOUR EMAIL ADDRESS"`

GIT - Principais comandos

Clonando repositório do GITHUB

- git clone <https://github.com/><username>/<nome_repositorio>.git

*Substituir <username> por seu nome de usuário no github;

*Substituir <nome_repositorio> pelo nome de seu repositório no github.

GIT - Principais comandos

Preparando a adição de um arquivo no repositório local

- `git add <arquivos...>`

*Substituir <arquivos...> por uma lista de arquivos a ser enviado separados pelo caracter “espaço”

GIT - Principais comandos

Adicionando os arquivos no repositório local e preparando os arquivos do repositório local para sincronização ao GITHUB

- `git commit -m "<Mensagem de sincronização aos desenvolvedores envolvidos no projeto>"`

*Substituir <Mensagem de sincronização aos desenvolvedores envolvidos no projeto> por uma mensagem útil aos desenvolvedores do projeto. Por exemplo: Atualização de folhas de estilo CSS para modificação de design para design responsivo.

GIT - Principais comandos

Sincronizando mudanças no repositório local com repositório no GITHUB

- `git push`

Acesse seu repositório no github.com e verifique as mudanças

Exercício -

Carregar o site “Portifólio” em um repositório denominado “Portfolio” em seu repositório no GITHUB

GIT - Principais comandos

Seja a seguinte situação:

Você possui um computador em um escritório e um computador em sua casa.

O projeto está sendo gerenciado no GITHUB (um terceiro computador “Servidor”).

Toda alteração deve ser armazenada no Servidor.

- **Pergunta: Qual seria um problema nesta situação?**

Escreva sua resposta em um arquivo TXT e envie no moodle.

GIT - Principais comandos

Resgatando alterações realizadas em um repositório local para outro repositório local

- `git pull`

O comando `git pull` realiza alterações no sistema

GIT - Principais comandos

Verificar versões do repositório

- `git log -pretty=oneline`

Lista as versões existentes no repositório - da última para a primeira

GIT - Principais comandos

Voltar versão do sistema

- `git checkout <codigo_da_versao>`

Para obter o código da versão utilize o comando de log do slide anterior

GIT - Principais comandos

TAGS de versões

- `git tag -a <nome_da_tag> <codigo_da_versao> -m "Mensagem referente a esta versão"`

GIT - Principais comandos

Exibindo informação sobre TAGS de versões

- `git show <nome_versao>`

GITHUB - Colaboradores

- **Acessar o repositório**
- **Configurações (settings)**
- **Colaboradores (Collaborators)**
- **Informe o nome de usuário do colaborador no GITHUB**

Exercício

- **Em grupo de 4 ou 5 pessoas (três grupos de 5 pessoas; um grupo de 4 pessoas), identifique um proprietário de um repositório e este criará o repositório “exercício GIT”.**
- **Adicione os outros integrantes no repositório e o professor (ifsp-jrbeluzo)**
- **O Líder deve distribuir as tarefas de programação de acordo com os arquivos a seguir:**
 - **Index.php / estilos.css(Participante 1)**
 - **calculadora.php / gera_calculo_calculadora.php (Participante 2)**
 - **imc.php / gera_calculo_imc.php (Participante 3)**
 - **form_cadastro_pessoa.php / cadastro_pessoa.php (Participante 4)**
 - **Funcoes.php / cabecalho.php (Participante 5 para grupo de 5; participante 1 para grupo de 4)**

Exercício

- **Index.php / estilos.css / cabecalho.php (Participante 1)**
- 1) **Index.php**: Deve conter uma chamada para o cabeçalho com o título da atividade + os links para as funcionalidades: cadastro de usuário, calculo de imc e calculadora. Também deve conter 3 tabelas (caso haja dados cadastrados nas mesmas):
 - a) Tabela de Usuários
 - b) Tabela de cálculos da calculadora
 - c) Tabela de Calculos de IMC
- 2) **estilo.css**: Deve conter uma folha de estilo para ser aplicada a todas as páginas, incluída no arquivo cabecalho.php

Exercício

- **calculadora.php / gera_calculo_calculadora.php (Participante 2)**
-
- 1) **calculadora.php**: Formulário que coleta dois números e a operação desejada (+, -, *, /, %) e quem está realizando a operação (deve vir dos cadastros de usuários realizados na sessão).

 - 2) **gera_calculo_calculadora.php**: recebe os dados de calculadora.php, exibe o resultado, informa quem solicitou o calculo (anônimo, caso não tenha sido selecionado nenhum usuário) e grava o resultado para o respectivo usuário na sessão para ser exibido na index.php

Exercício

- **imc.php / gera_calculo_imc.php (Participante 3)**

- 1) **imc.php**: Formulário que coleta dois números (Altura e peso de uma pessoa) e quem está realizando a operação (deve vir dos cadastros de usuários realizados na sessão).
- 2) **gera_calculo_imc.php**: recebe os dados de imc.php, exibe o resultado, informa quem solicitou o calculo (anônimo, caso não tenha sido selecionado nenhum usuário) e grava o resultado para o respectivo usuário na sessão para ser exibido na index.php

Exercício

- **form_cadastro_pessoa.php / cadastro_pessoa.php (Participante 4)**
-
- 1) **form_cadastro_pessoa.php**: Formulário que coleta nome, email, sexo, data de nascimento, cpf, cidade e estado.

 - 2) **cadastro_pessoa.php**: recebe os dados de form_cadastro_pessoa e grava na sessão para ser exibido na index.php e nas seleções de calculadora.php e imc.php

Exercício

- **funcoes.php (Participante 1 ou 5)**
-
- 1) **funcoes.php**: Funcoes para o calculo de IMC e calculadora a ser utilizada nos arquivos de calculo.

 - 2) **cabecalho.php**: Deve conter as especificações definidas no 1o item assim como a chamada do arquivo estilo.css e a inclusão do arquivo de funções.