

Organização do Lab 1

Software

LAMP Stack:

- Linux Mint Rosa 17.3 Mate 32-bit (Kernel 3.19)
- Apache 2.4.7
- MySQL 5.5
- PHP 5.5

Software instalado:

- | | |
|----------------|--------------------------------------|
| • Vim | Editor de código |
| • Git | Controle de versões |
| • Sublime-Text | Editor de código |
| • Netbeans 8.1 | IDE (plugins Java SE, PHP, HTML5) |
| • Workbench | Gerenciador de bancos de dados MySQL |
| • phpMyAdmin | Gerenciador de bancos de dados MySQL |
| • Meld | Comparador de código fonte |
| • PyRenamer | Renomear arquivos em lote |
| • Gimp | Edição de imagens mapa de bits |
| • Inkscape | Edição de gráficos vetoriais |
| • Firefox | Web browser |

Script configuração:

<https://github.com/aledruetta/distro-install/blob/master/mint17-install.sh>

Estações de trabalho

Usuário administrador: etec

Password: *****

Usuário normal: aluno

Password: aluno

IP: dinâmica (DHCP)

Diretório de trabalho: /home/aluno/html

// link simbólico → /var/www/html

URL: <http://localhost>

phpMyadmin: <http://localhost/phpmyadmin>

Banco de dados

Usuário administrador: root

Password: *****

Usuário: aluno

Password: aluno

Database: aluno_...

Servidor

Usuário administrador: etec

Password: *****

Usuário normal: helth

Password: helth

IP: 192.168.0.170

Diretório de trabalho: /home/etec/html

// link simbólico → /var/www/html

Permissões: 775 (directórios) e 664 (arquivos)

Grupo: www-data

URL: <http://server>

phpMyadmin: <http://server/phpmyadmin>

Banco de dados

Usuário administrador: root

Password: *****

Usuário administrador com permissões em “helth”: helth_admin

Password: *****

Usuário normal: helth

Password: helth

Database: helth_...

Controle de versões

Fluxo de trabalho do Scrum Master

Apenas a **primeira vez**:

- Instalar e configurar o Git (no notebook pessoal)
- Clonar o repositório do projeto

```
$ sudo apt install git
$ git config --global user.name "Nome Sobrenome"
$ git config --global user.email "usuario@servidor.com"
$ git clone https://github.com/CaicarasHackerClub/hug-health.git
$ git checkout devel
```

Sempre que começar a trabalhar, dentro da pasta do projeto:

- Atualizar o código com o que tem de novo no repositório:

```
$ git checkout devel
$ git pull origin devel
```

- Editar o código (bonitinho)
- Testar o código (muito)
- Verificar se alguma coisa mudou no repositório enquanto isso:

```
$ git pull origin devel
$ git log
```

- Testar de novo
- Se está tudo certinho fazer o **commit**:

```
$ git status
$ git add arquivo_modificado1 arquivo_modificado2 arquivo_novo etc
$ git commit -m "Título descritivo do commit até 65 caracteres"
$ git push -u origin devel (primeira vez só)
$ git push (daí pra frente o git já vai saber)
```

Fluxo de trabalho dos membros do time

Sempre que começar a trabalhar, dentro da pasta do projeto:

- Atualizar o código com o que tem de novo no repositório:

```
$ git checkout devel  
$ git pull origin devel
```

- Editar o código (bonitinho)
- Testar o código (muito)
- Verificar se alguma coisa mudou no repositório enquanto isso:

```
$ git pull origin devel  
$ git log
```

- Testar de novo
- Se está tudo certinho, levar pro **Scrum Master** integrar no repositório