|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ATA DE REUNIÃO - PESQUISA APLICADA A NOVAS METODOLOGIAS PARA DESENVOLMENTO DE SISTEMAS** | | | | | |
| **DATA**: | 11/04/2014 | **LOCAL**: | Escritório - WLL Sistemas | **RESPONSÁVEL**: | Ronaldo |
| **ASSUNTO**: | Desenvolvimento no Git | | | | |
| **PARTICIPANTES** | Alison – Elzo – Ronaldo – Vinicius – William | | | | |

Reunião sobre o processo de programação e compartilhamento do sistema:

Foi decidido que para o projeto, seria utilizado um novo método de Desenvolvimento Colaborativo de Programação em Equipe chamado de Git.

Foi criado acesso para todos integrantes do Projeto com usuário e senha, este por sua vez será utilizado para controlar o Fluxo e andamento do Projeto.

O programa Git, será utilizado para compartilhar os ativos de software, onde os desenvolvedores da organização podem aproveitar o código pré-construído, onde um integrante pode continuar o código do outro, podendo até mesmo encontrar erros e corrigi-los, com isso, se alguém melhora o código, toda equipe ganha, economiza tempo e dinheiro.

A partir de vários estudos sobre o uso do Git, o integrante e Líder da equipe William, desenvolveu um manual de utilização para os demais integrantes, o mesmo pode ser conferido nas páginas abaixo:

**Configurando e utilizando GIT no Windows e Linux**

Na internet existe um vasto material para se aprimorarmos, podemos começar com o link abaixo: <http://gitscm.com/book/pt-br/>.

Podemos trabalhar com linha de comando, mas existem programas para fazer a interface gráfica tanto em Windows quanto em Linux, sempre que digitar uma linha de comando pressionar a tecla ENTER no final.

1 – Instalação do git :

**Linux (Ubuntu): # apt-get install git-core**

**Windows:** baixar o instalador nesse link: http://git-scm.com/downloads, após a instalação no

windows acesse o atalho Git Bash para abrir o prompt de comando.

2 – Na configuração inicial, não é necessário navegar para nenhum diretório, todos os comandos

se iniciam com a palavra **Git** .

**Linux e Windows: git config --global user.name “SeuNome”**

**Linux e Windows: git config --global user.email “SeuEmail”**

**Linux e Windows: git config --global color.ui auto**

No Linux pode-se indicar um editor padrão para edição de configurações do Git:

**Linux: git config --global core.editor vi, vim, nano e etc.**

Para verificar as configurações:

**Linux e Windows: git config –list**

3 – É necessário gerar uma chave de identificação para cada máquina, filiada ao usuário. Essa

chave pública será gravada no site GtiHub junto com o nome do usuário (Cada um de vocês), não é necessário usar a palavra Git antes.

**Linux e Windows: ssh-keygen -t rsa**

Será solicitado que digitem um nome para os arquivos que serão gerados, e uma senha que terá de ser confirmada, uma mensagem é exibida informando em qual diretório está sendo gravado o arquivo contendo a chave pública.

Esse arquivo fica como oculto, então é necessário que configure o S.O. para exibir arquivos ocultos, geralmente fica com e extensão .PUB, com o auxilio de um editor de textos copie o conteúdo desse arquivo.

4 – Habilitar usuários para commit no servidor GitHub.

Como sabem o nosso projeto agora está hospedado no servidor GitHub então é necessário que todos os integrantes do grupo estejam cadastrados como colaboradores para terem permissão de commit no projeto. Vocês devem acessar esse endereço https://github.com/ e acessem o painel administrativo com meu email: ?????@ig.com.br e minha senha: XXXXX.

Na parte de inferior da página cliquem no link ***projeto-saude*** e na próxima página na lateral direita, cliquem no link ***Settings***, na próxima página na lateral esquerda cliquem no link ***Deploy keys*** e no botão ***Add deploy key***.

Na janela que abrir, vocês devem digitar no campo título o nome de vocês e no campo key devem colar o conteúdo do arquivo com a chave pública e pressionar o botão ***Add Key***, pronto vocês já estão habilitados a efetuar commit no projeto.

5 – Clonar o diretório do projeto hospedado para máquina física de vocês.

Aconselho a excluir a pasta *~~ProjetoPedro~~* que vocês já possuem no diretório do apache e criar uma nova **vazia**, uma vez que a versão que está no servidor é a mais recente. Utilizando o prompt Git Bash naveguem até o diretório que vocês criaram, estou assumindo que seja *~~ProjetoPedro~~* mesmo.

Dentro do diretório digitem o os seguintes comandos:

Criar um repositório local:

**Linux e Windows: git init**

Agora vamos adicionar o endereço do repositório remoto a git local:

**Linux e Windows: git remote add origin** [**https://github.com/wllfl/projeto-saude**](https://github.com/wllfl/projeto-saude)

Para copiar os arquivos remotos para a pasta local basta usar o comando abaixo:.

**Linux e Windows: git pull origin máster**

Observação: poderíamos clonar o repositório remoto com o comando “**git clonegit://github.com/wllfl/projeto-saude**”, mas isso iria criar outra pasta dentro do diretório corrente.

6 – Commitar alterações no código fonte.

É importante mencionar que antes de qualquer alteração nos fontes é aconselhável executar novamente o comando pull que foi mencionado acima. Esse comando atualiza os fontes novamente, vai que um de vocês já tenho comitado algo no servidor.

Basicamente temos que adicionar os arquivos que queremos controlar antes de comitar, nesse caso, vamos adicionar todos os arquivos que estão dentro do diretório corrente:

**Linux e Windows: git add.**

Para comitar e confirmar sua alteração no repositório local basta usar o comando:

**Linux e Windows: git commit -m “Mensagem contendo a justicativa da alteração”**

O commit é feito localmente agora temos que subir nossas alterações para o servidor, será solicitado username (?????@ig.com.br) e senha(XXXXX).

**Linux e Windows: git push origin máster**

**Sempre executando o comando pull antes de iniciar as alterações, se o git notar que existem divergências entre os arquivos do servidor e os locais desde o último push para o atual ele vai emitir uma mensagem de erro e cancelar o push.**

7 – Comandos Extras:

Para remover um repositório remoto origin:

**Linux e Windows: git remote rm origin**

Para llistar os repositórios remotos existentes:

**Linux e Windows: git remote –v**

**Esses comandos foram testados um por um e funcionou no *Windows Seven*, *Ubuntu* e *Fedora*, se surgirem problemas digitem www.google.com na url do navegador e mãos a obra na pesquisa.**

**OBSERVAÇÕES: Conselho, só comitem depois de testar muito bem o código e várias vezes, subam as alterações para o servidor somente de tempos em tempos, não toda hora, até porque vocês tem que colocar a mensagem para justificar o commit, então, seja objetivo.**

**Exemplo: não é porque adicionou um <br> ou aumentou uma margem que vocês precisam incluir, deixem juntar mais alterações para enviar para o servidor.**