|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nome** | **Implementado** | **Testado** | **Funciona** | **Observações** |
| Init | sim | sim | Parcialmente | Se o diretório não existe, ele será criado.  Se o diretório existe e esta vazio, ele será iniciado.  **Problema: Se o diretório e já possui repositório, ele será sobreescrito.**  **Solução:**  Verificar se já existe um repositório não vazio.  **Responsável: Danielli** |
| Add | sim | sim | sim | Se o arquivo existe, ele será adicionado ao repositório.  **Problema: não avisa o usuário, caso o arquivo não exista. mostra o status do ultimo arquivo adicionado.**  **Solução:** Verificar se arquivo existe, caso não exista alertar o usuário.  **Responsavel: Pedro.** |
| Remove |  | sim | sim | Se o arquivo não esta commitado para o repositório, ele será excluido.  Se o arquivo esta commitado para o repositório, ele será removido e excluído.  **Existem outros tipos de remoção, como remoção do histórico, remoção do índice, etc.**  **Problema: não avisa o usuário, caso o arquivo não exista.**  **Solução:** Verificar se arquivo existe, caso não exista alertar o usuário.  **Responsavel: Pedro.** |
| Commit | sim | sim | sim | **Problema: mensagem para o usuário é apenas o ID do arquivo, adicionar mais dados.**  **Solução:** Mostrar nome do arquivo.  **Responsável: Danielli** |
| Clone | sim | em Teste | em Teste | Se o diretório de destino do comando clone **não existe**, ele será criado.  Após isso os arquivos serão clonados.  Se o diretório de destino do comando clone **existe**, e esta vazia e não for um repositório, ele é iniciado, e os arquivos são clonados.  Implementar em Ssh.  **Problema: Se o diretório de destino clone existe, não esta vazia, e não for um respositório, os arquivos não serão clonados.**  **Se o diretório de destino clone existe, e é um repositório seja ele vazio ou não, os arquivos não serão clonados.**  **Obs: o próprio codigo JGit, não permite este tipo de caso.**  **motivo:**  **if (directory.exists() && directory.listFiles().length != 0)**  **throw new JGitInternalException(MessageFormat.format(**  **JGitText.get().cloneNonEmptyDirectory, directory.getName()));**  **command.setDirectory(directory);**  **return command.call().getRepository();**  arquivo fonte**: CloneCommand.java**  **Solução:** Pensar se vale mudar a estrutura do comando JGit ou aceitar essa condição.  **Responsável: Danielli e Pedro** |
| ShowStatus | sim | Parcialmente | A Averiguar | Alterar o Formato da Saida  Funcionando para casos pequenos.  **Responsável: Danielli, Pedro**  Averiguar em repositórios grandes, com muitos arquivos. |
| Log | sim | sim | sim | - Alterar a saida para o usuário.  - Acrescentar: Author, Date e Message do commit.  **Responsavel: Pedro.** |
| CreateBranch | sim | sim | sim | Mudar a saida para o usuário.  **Responsavel: Pedro.** |
| ShowBranch | sim | sim | sim | Mudar a saida para o usuário.  **Responsavel: Pedro.** |
| DeleteBranch | sim | sim | sim | Mudar a saida para o usuário.  **Responsável: Danielli** |
| diffBranch | sim | parcialmente |  | O diff é feito entre os arquivos do branch que ja foram comitados.  **Responsavel: Danielli, Pedro.** |
| CheckoutBranch | não |  |  |  |
| Desfazer Commit | não |  |  | Material de consulta: <http://git-scm.com/book/pt-br/Git-Essencial-Desfazendo-Coisas> |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**Prioridades:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nome | Implementado | Testado | Funciona | Observação |
| Merge | não |  |  |  |
| Push | não |  |  | Implementar em Https e Ssh |
| Pull | não |  |  | Implementar em Https e Ssh |

|  |  |
| --- | --- |
| Atividade | ação |
| Janelas para busca de diretorios | Implementar uma interface para que seja possivel encontrar o caminho de um diretório de maneira mais rapida, sem que seja necessário digitar o caminho. |
| WebService | Estudar e implentar exemplos de funcionalidades WebServices. |
| Correção de Erros do Projeto | Corrigir os bugs e os erros do código já implementado. |