

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
DE SÃO PAULO**

Campus São João da Boa Vista

Trabalho Final de Curso

4º ano – Curso Técnico em Informática

Prof. Breno Lisi Romano

**Organização do projeto Reclame São João através da ferramenta de
trabalho SVN**

Aluno: Rafael Dattoli de Pontes

Prontuário: 142078x

São João da Boa Vista – SP

2017

Resumo

Esta pesquisa, que tem como base o projeto produzido pelos alunos do quarto ano de informática do Instituto Federal, chamado Reclame São João, foca em mostrar como a ferramenta de trabalho SVN é importante para o desenvolvimento de um trabalho, podendo ser um projeto de grande porte ou não.

O objetivo principal desta pesquisa, é mostrar como o SVN influencia e é importante para o desenvolvimento de um projeto, no qual é mostrado toda a história do aplicativo e como ele é essencial para o auxílio direto com o trabalho dos usuários, já que ele exerce muito bem seu papel, que é o controle dos dados de um projeto e consequentemente a organização das tarefas.

O método utilizado para esta pesquisa, foi através da explicação sobre o que é a ferramenta, caso o interessado pelo assunto ainda não saiba, junto com o esclarecimento do seu funcionamento e a apresentação da participação da ferramenta direto com o projeto, e no fim, a apresentação da opinião de alguns alunos do quarto ano sobre como foi suas respectivas experiências com os primeiros contatos com o uso da ferramenta.

Os resultados obtidos com a pesquisa são considerados bastante satisfatórios, levando em conta os dados apresentados e utilizados para o desenvolvimento, e também a questão dos dados pessoais retirados diretamente do projeto, o que gerou uma originalidade e simplicidade para a explicação geral apresentados com o decorrer da pesquisa.

Palavras chaves: Organização, controle de versão, ferramenta SVN.

Sumário

1	Introdução	5
2	Desenvolvimento	6
2.1	Fundamentação teórica	6
2.1.1	O que é um controle de versões?	6
2.1.2	Quais as vantagens do uso de um controle de versões?	7
2.1.3	O que mudou com o controle de versões?	8
2.1.4	Antes do SVN	8
2.1.5	A história do SVN	10
2.1.6	Os autores responsáveis pelo desenvolvimento do SVN	10
2.1.7	Disponibilidade do SVN	11
2.1.8	As funcionalidades do SVN	11
2.1.9	Os principais comandos do SVN utilizados no desenvolvimento do projeto	13
2.1.10	Como as pastas do projeto ficaram divididas no repositório SVN	14
2.1.11	Os responsáveis do projeto pelo repositório SVN	17
2.1.12	Outras ferramentas utilizadas no gerenciamento do projeto	19
2.1.13	Pesquisa sobre o SVN	20
3	Conclusões e Recomendações	24
4	Referências Bibliográficas	26

Lista de figuras

Figura 1: vários clientes conectados em um mesmo servidor.....	7
Figura 2: Tabela apresentando as diferenças entre CVS x Subversion.....	9
Figura 3: Print da documentação do projeto, no qual é apresentado os comandos do SVN..	12
Figura 4: Print da documentação do projeto, mostrando a divisão de pastas do desenvolvimento	15
Figura 5: Print da documentação do projeto, mostrando a divisão de pastas da documentação.....	15
Figura 6: Print da documentação do projeto, mostrando a divisão de pastas dos relatórios. .	16
Figura 7: Tabela apresentando a divisão das responsabilidades de gerências.	18
Figura 8: Imagem representando o diretório geral do kanbam em relação ao projeto Reclame São João.	19
Figura 9: Imagem representando como seria a criação de uma nova tarefa do redmine.	20

Lista de gráficos

Gráfico 1.....	21
Gráfico 2.....	22
Gráfico 3.....	23

1 Introdução

Esta pesquisa, produzida pelos alunos do quarto ano de informática do Instituto Federal de São Paulo, tem como objetivo focar e analisar vários aspectos do projeto “Reclame São João”, que tem o prazo para ser entregue de 2 semestres.

Ela será composta por três capítulos junto a subcapítulos, ou seja, “1.1 – 1.2- 1.3 – 1.0.1 – etc.” O primeiro capítulo é responsável pela introdução, o segundo contém o a apresentação do desenvolvimento e o terceiro apresenta a conclusão do trabalho e os resultados que a pesquisa gerou.

O principal objetivo que este projeto tem, é ajudar e orientar as pessoas que moram e passam pelo interior de São Paulo, em São João da Boa Vista. Com ele, as pessoas são orientadas e podem fazer suas avaliações e reclamações, ou deixar suas opiniões sobre quase todos os comércios da cidade. Sabendo assim, quais os melhores lugares da região para frequentar e o que os estabelecimentos podem fazer para melhorar seu desempenho e estarem sempre agradando seus clientes.

O Reclame São João é composto por 5 módulos, responsáveis por partes diferentes do desenvolvimento do projeto, com isso, vários temas para relatórios podem ser extraídos e aproveitados. Então através deste documento, temos uma explicação sobre como cada ramificação específica do projeto foi importante e como ela foi planejada e produzida pelos alunos.

Diante disso, a responsabilidade deste estudo científico com relação ao projeto desenvolvido pelo quarto ano, é em relação à ferramenta de trabalho que serve para organizar e armazenar todas as informações do projeto, chamada *Subversion* (SVN) um software que sem dúvidas facilita muito o trabalho de todos envolvidos no projeto.

É importante falar sobre o SVN, pois é uma ferramenta essencial para uma boa evolução do projeto, sem ele, teria riscos de perdas de informações e atrasos. Com o uso dele, é possível ter um gerenciamento dos arquivos e diretórios de tudo que acontece no projeto, com o controle das alterações e atualizações realizadas ao longo do tempo.

Um ambiente que não adota essa ferramenta pode vir a ocasionar alguns problemas. Em algum projeto parecido com o Reclame São João, seja do mesmo rumo ou não, e que não adota essa ferramenta, pode ter consequências, como: perda de dados, falta de organização, subscrição de documentos, falta de atualizações, entre outros. Esses são alguns aspectos que podem ser

considerados relevantes para falar sobre tal, e que mostra a importância de explicar e se aprofundar sobre esse mecanismo de trabalho.

Para explicar a grande vantagem, ou benefício do uso do SVN, é só pensar no foco principal da ferramenta, que é administrar as mudanças de informações que ocorrem em todo momento, em um projeto com vários programadores acontecem diversas modificações pequenas em todo instante, ou seja, existem várias pessoas mexendo em um mesmo trabalho e gerando de hora em hora diversas alterações, mesmo sendo grandes ou pequenas mudanças, e com o uso do SVN esse problema se torna algo bem mais simples e prático de ser resolvido.

2 Desenvolvimento

2.1 Fundamentação teórica

Para obter um melhor conhecimento desta pesquisa, é inevitável deixar de lado alguns conceitos mais profundos sobre a ferramenta abordada, e para isso deve ser feita uma fundamentação teórica com vários aspectos e detalhes relacionados ao assunto, desde o seu surgimento, no qual retrata sua evolução através de mudanças e alterações feitas com o passar do tempo, até o seu funcionamento atual, que é essencial para o desenvolvimento deste trabalho.

2.1.1 - O que é um controle de versões?

Um controle de versão geralmente é muito utilizado em instituições tecnológicas, e é um programa bem útil para o desenvolvimento de um projeto, tanto em casos mais simples como em casos mais complexos, pois um controle de versão é responsável por criar um repositório em um servidor qualquer, que vai ser utilizado por um cliente que esteja em contato direto com a máquina do servidor. Esse repositório vai armazenar as informações de um projeto, e isso é possível ser feito em uma quantidade ilimitada de dispositivos, ou seja, os documentos e informações que possuem nesse repositório não se perdem. Através de um controle de versões é possível adicionar, remover e atualizar informações, e caso algum dado seja alterado por engano é possível recuperá-lo, mas as funcionalidades que um

determinado controle de versões possui, será explicada em um outro momento, o que não vem ao caso. Por esses motivos um controle de versão é tão essencial como forma de auxílio em trabalhos, empresas e até mesmo para uma organização na vida pessoal das pessoas. [1]

Exemplificação de um controle de versões, no qual vários dispositivos se conectam em um mesmo servidor para controlar determinadas informações:

Figura 1: vários clientes conectados em um mesmo servidor.



Fonte disponível em: <<http://www.devmedia.com.br/sistemas-de-controle-de-versao/24574>> Acesso em: 24 de outubro de 2017.

2.1.2 - Quais as vantagens do uso de um controle de versões?

Existem diversas ferramentas que exercem a função do controle de versões, como o GIT, Mercurial, e o SVN, que se diferenciam principalmente pela adequação, dependendo do tipo de trabalho que será realizado, como também pelo desempenho, eficácia e simplicidade de uso e até mesmo pela sua popularidade, mas a ferramenta que foi utilizada para o auxílio do desenvolvimento do projeto Reclame São João, foi a ferramenta SVN, que foi escolhido pelo seu bom desempenho em um trabalho no qual se envolvem diversos desenvolvedores trabalhando em uma mesma objetividade, e apresenta algumas características vantajosas como:

- Com o SVN é possível selecionar a cópia do trabalho de um arquivo para manter a coerência e evitar conflitos entre os usuários;
- É possível criar e aplicar patches, mantendo consistência no projeto;
- Trabalhar com gráficos de revisão para verificar as alterações feitas no projeto;
- Ver quem fez mudanças durante o projeto;
- Deixar mensagens no projeto para explicar para o próximo usuário o que foi feito;
- Bloquear arquivos para que ninguém mais possa alterá-los enquanto você está trabalhando neles;
- Criar ramos para permitir a manutenção de várias versões do projeto uma única vez e depois juntar tudo. [3]

2.1.3 - O que mudou com o controle de versões?

Antes de existir programas para manter o controle de versões de um projeto, as coisas eram mais difíceis, pois caso alguém quisesse salvar alguma nova alteração dentro do projeto sem que a versão antiga fosse alterada, o usuário deveria criar dois arquivos e deixar ambos salvos, e caso acontecesse isso em um projeto maior, como por exemplo o Reclame São João, ocasionaria em um grande desperdício, pois traria um peso extra desnecessário de arquivos, o que seria algo que atrasaria todo o trabalho, podendo deixá-lo mais lento por conta do grande aumento de peso e também por conta da bagunça que o projeto ficaria. Atualmente, com a inovação dos controles de versões, os envolvidos em projetos podem salvar seus trabalhos sem medo de acabar perdendo o que já foi feito, permitindo ao programador, retornar a versão em caso de perdas ou alterações feitas de forma desnecessária, e usar ela para fazer melhorias e atualizações, sem gerar desperdício de espaço e sempre permitindo um projeto limpo e organizado, sem excesso de documentos sem importância e que sobrecarregam o repositório do projeto.

2.1.4 - Antes do SVN

Em 1984, os desenvolvedores de alguns projetos começaram a perceber a dificuldade de trabalhar em um grupo com várias pessoas que tinham o mesmo objetivo de desenvolvimento, e muitas vezes não podiam programar no mesmo horário por conta de problemas de comunicação e subscrição de arquivos, e foi quando foi lançado o primeiro código do Sistema de Versões

Concorrentes (CVS) publicado em 1986, que foi lançado oficialmente em 1989, com a maioria das funcionalidades da ferramenta que mais tarde se transformou em *Subversion*, em 2004. Um dos principais fatores que colaborou com essa evolução do aplicativo, foi em questão das limitações que a ferramenta CVS possuía, como a negação de mover os diretórios ou fazer uma alteração de seus nomes, ou seja, cada arquivo do subdiretório deveria ser alterado individualmente, o que ocasionava atrasos no desenvolvimento, e também a questão do “checkout” que não permitia que dois usuários alterassem o mesmo arquivo ao mesmo tempo, o que poderia gerar mais problemas do que eficiências em um trabalho. [2]

Tabela com as principais transformações da ferramenta CVS para o SVN

Figura 2: Tabela apresentando as diferenças entre CVS x Subversion.

Característica	CVS	Subversion
Autenticação via Active Directory ou SSH	SIM	NÃO
Rename	NÃO	SIM
Versionamento de Pastas e Metadados	NÃO	SIM
Commits Atômicos	NÃO	SIM
Tags e Branches	Identificação	Cópia
Mantém o histórico de cópias e arquivos renomeados	NÃO	SIM
Armazenamento	Sistema de Arquivos	Banco de Dados
Tratamento de arquivos binários	NÃO	SIM

Fonte disponível em: <<https://pt.slideshare.net/misaelssantos/gcs-aula-07-sistemas-de-controle-de-verses>> Acesso em: 16 de outubro de 2017.

2.1.5 - A história do SVN

Os responsáveis pelo desenvolvimento do SVN perceberam que o aplicativo que era usado como controle de versões antigamente estava precisando de mudanças, e foi quando Tim Kemp iniciou o desenvolvimento para aprimorar esse antigo aplicativo chamado CVS e deixou suas linhas de código trabalhadas no novo projeto em fonte online, ou seja, qualquer pessoa poderia ter acesso ao que foi feito por ele, o que fez com que Stefan Kung se juntasse ao trabalho para a continuação do desenvolvimento como programador, reescrevendo todo o código antigo e adicionando alguns novos comandos, dando origem ao mais novo controle de versões chamado Subversion, que inovou de uma maneira bastante estável, que fez com que novos usuários fossem atraídos ao, junto com Lubbe Onken, que ofereceu ajuda aos envolvidos e se uniu ao projeto do SVN, trazendo seus variados ícones aprimorados e uma logomarca para o aplicativo.

Depois de todo esse trabalho e inovações, acabou que em 23 de fevereiro de 2004 se tornou a data em que grande parte dos usuários do CVS deixaram de usar a ferramenta para migrar no novo sistema de controle de versões que estava muito mais atualizado, chamado *Subvserion* na versão 1.0, no qual possuía todas as características do CVS só que com uma aparência aprimorada e agora com a possibilidade de mover e renomear os documentos de dentro do repositório, que foi um dos principais motivos para fazer com que o CVS fosse descartado. [5]

2.1.6 - Os autores responsáveis pelo desenvolvimento do SVN

Para ser possível a criação do Subversion existem alguns nomes que devem ser citados por terem grande importância para a existência do aplicativo hoje, começando pelo britânico Tim Kemp, que foi responsável pela inicialização do projeto e que também viu as necessidades de um novo controle de versões mais prático e com mais funções e tomou coragem para dar início ao projeto, surgindo em seguida a participação do suíço Stefan Kung, que foi responsável com o projeto na área de programação e fez a liderança do desenvolvimento, logo depois surge Lubbe Onken com seus ícones inovadores, e sempre cuidando da programação visual do aplicativo, Simong Large também foi um grande nome colaborador para o desenvolvimento do projeto por ser responsável com a área da documentação do projeto, sempre facilitando e organizando os processos de desempenho do programa, e por último o austríaco Stefan Fuhrmann, que trabalhou com o cache de logs e com os gráficos de revisões do projeto, ou seja, para se tornar possível o desenvolvimento do controle de versões SVN necessitou-se de grandes nomes e intelectuais trabalhando juntos para ter um bom desempenho, também é importante lembrar os livros e manuais de instruções criados para ter um melhor entendimento e facilidade de uso do aplicativo para os novos usuários.

Esses detalhes que podem parecer simples é o que torna o Subversion um controle de versões prático e diferenciado, e que foi considerado o ideal para ser usado no desenvolvimento do projeto Reclame São João, que abordou alunos que não tinham conhecimento algum sobre um controle de versões, mas que acabou se tornando algo simples por conta da eficiência da ferramenta escolhida.

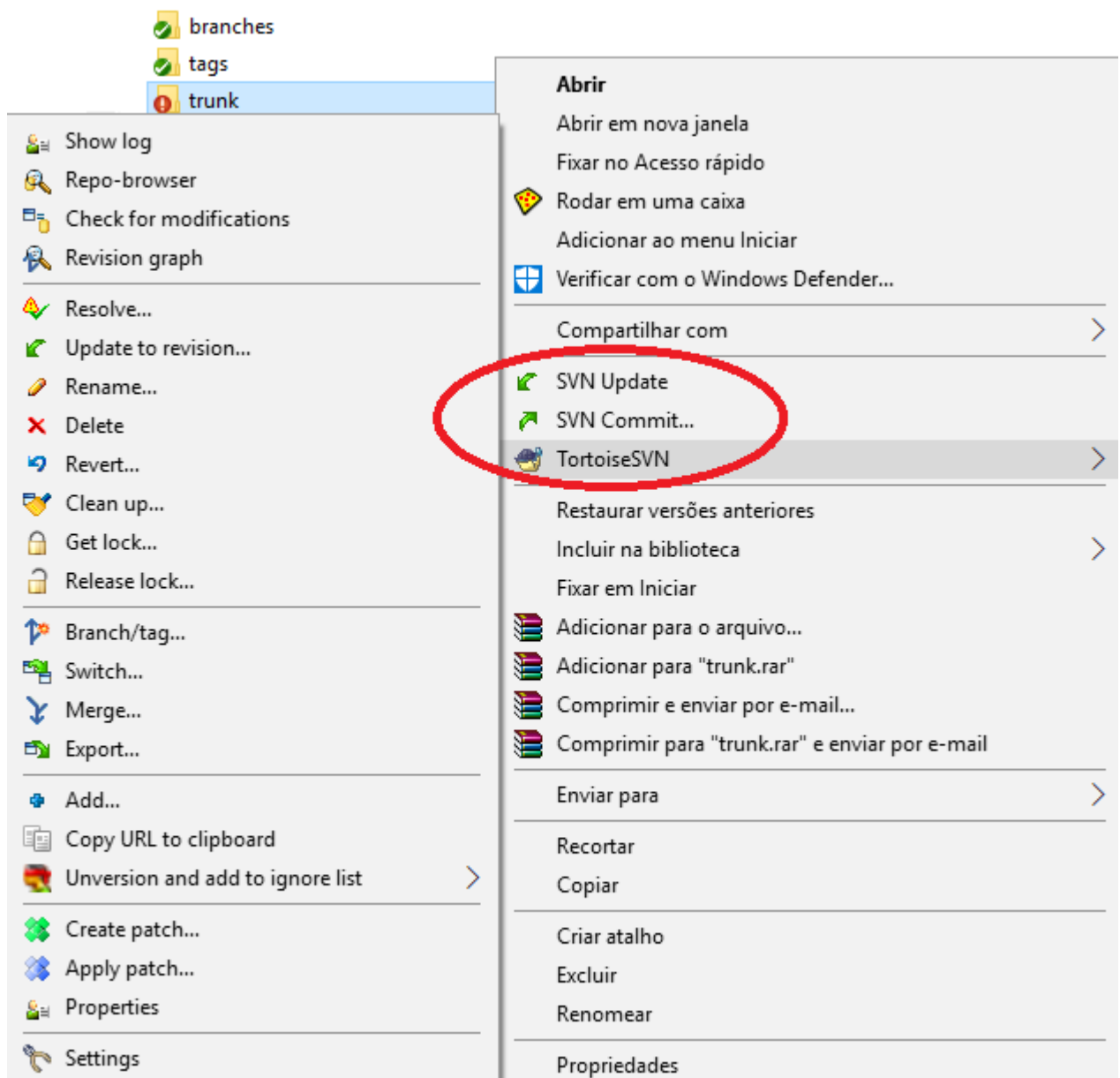
2.1.7 - Disponibilidade do SVN

O *Subversion* é um aplicativo grátis disponível para os sistemas operacionais Windows XP/Vista/7/8/8.1/10 e Linux, em 32 e 64 bits, sua última atualização foi em 16/08/2017, quando foi atualizado para a versão 1.9.7.27907 e necessita de apenas 14.80 *Megabytes* (MB) de espaço em disco para download, também é disponível em 41 idiomas diferentes, no qual o português do Brasil se inclui, e também é considerado um dos melhores aplicativos na sua função, pela sua eficiência e praticidade que traz ao usuário. [2]

2.1.8 - As funcionalidades do SVN

As funcionalidades que o Subversion pode oferecer são de grande número e de grande eficiência. Existem projetos em que a grande maioria de seus comandos são utilizados constantemente durante o desenvolvimento de um trabalho, mas para o Reclame São João foi necessário apenas o uso das funcionalidades mais comuns do programa, que é um assunto que será melhor detalhado no próximo tópico, pois agora será explicado o que cada comando do SVN faz quando executado, a imagem a baixo é uma imagem retirada da documentação do projeto dos alunos do quarto ano, e que apresenta a visão gráfica do controle de versões utilizado por eles e suas principais funcionalidades, que serão explicadas logo após a visualização da imagem:

Figura 3: Print da documentação do projeto, no qual é apresentado os comandos do SVN.



Fonte disponível em: [Repositório do projeto Reclame São João](#).

- O primeiro comando, e também o mais essencial para a existência do repositório, é o comando chamado *checkout*, que é usado justamente para fazer o primeiro download de todo um projeto para o repositório SVN, por esse motivo não é possível enxergar essa opção de comando na imagem a cima, já que esse é um comando que só é utilizado uma vez para habilitar o repositório para um novo usuário, que no caso já foi utilizado.

- O próximo comando é o commit, que é o comando que um usuário deve utilizar quando tiver feito alguma alteração no projeto e faz com que essa alteração fique salva e depois possa ser atualizada por qualquer outro usuário através do update.
- Como dito anteriormente, o comando update é o responsável por fazer as atualizações que foram feitas por qualquer usuário que usou e alterou algo de dentro do repositório, e então é possível ver essas atualizações feitas na forma de download dos novos arquivos atualizados pelo update.
- Caso algum usuário queira ver quais atualizações foram feitas por outros usuários dentro do repositório, ele pode usar o comando show log, que mostra quais foram as últimas modificações feitas no projeto, junto com informações de datas e horários.
- O comando add, é usado quando um usuário adiciona um novo arquivo dentro do repositório.
- O comando rename, é usado quando um usuário quiser renomear algum arquivo de dentro do repositório.
- O comando delete, é usado quando um usuário quer excluir algum arquivo de dentro do repositório.
- O comando export, serve para transportar o download de arquivos de um local para outro.
- O comando import, é usado para designar um novo repositório em uma estrutura de diretórios.
- O comando merge, é um comando bastante importante, supondo que aconteça um caso de dois usuários fazendo alterações em um mesmo arquivo, seria necessário a utilização desse comando para fazer a fusão das modificações feitas por ambos, fazendo com que não tenha perdas de dados ou subscrição de arquivos nas diretórios, um conflito que poderia atrasar o projeto.
- O comando revision, serve para ter uma numeração de cada modificação feita dentro do repositório, para ter um melhor controle das ações. [4]

2.1.9 - Os principais comandos do SVN utilizados no desenvolvimento do projeto

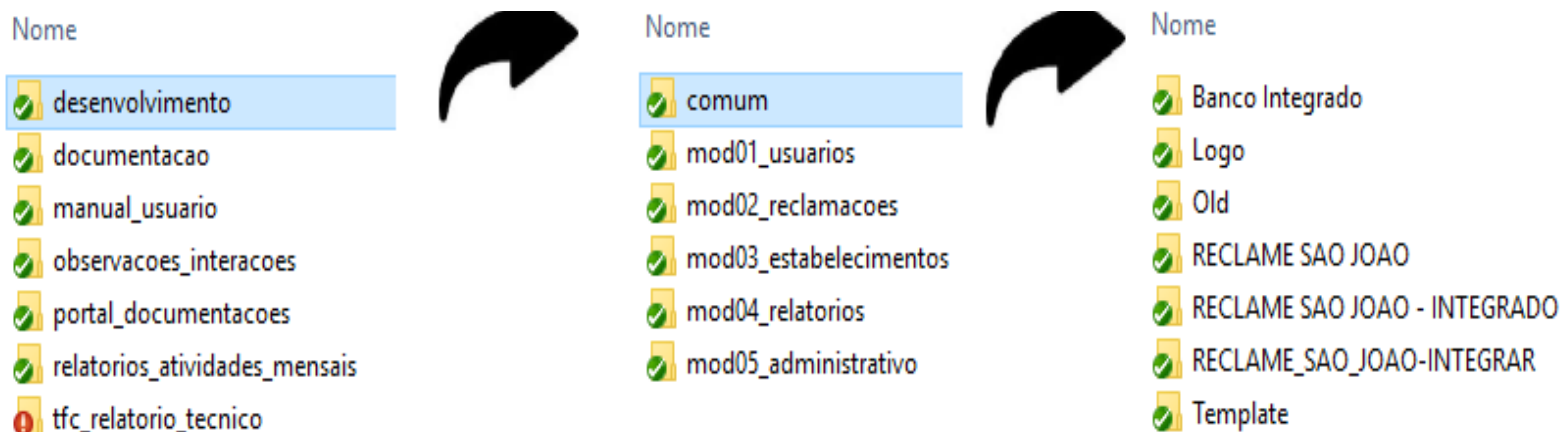
A ferramenta Subversion conta com vários tipos de funções de comandos e utilidades, porém, o projeto Reclame São João não teve a necessidade da utilização de comandos mais detalhados e específicos da ferramenta, e sim apenas dos comandos mais básicos que eram usados praticamente todo dia como o caso do *commit* e *update*, que são os comandos para alterar e

atualizar as informações feitas nos documentos de dentro do repositório. Durante o desenvolvimento do projeto não tiveram conflitos entre os próprios alunos usuários do SVN, e isso pelo fato de como as pastas foram divididas dentro do repositório, já que cada módulo possui suas respectivas pastas, podendo ter um controle de organização melhor sobre o que era modificado, e então somente no final do projeto foi feita a integração das linhas de código, o que fez com que não tivessem problemas com isso, e conseqüentemente não fosse preciso o uso de outros comandos mais específicos do subversion, como em casos de alguns projetos que acabam tendo que recorrer a versões anteriores por causa de perdas de dados.

2.1.10 - Como as pastas do projeto ficaram divididas no repositório SVN

Como citado no tópico anterior, para um melhor desempenho durante o desenvolvimento do projeto, as pastas dentro do repositório foram divididas por módulos, então cada grupo tinha suas pastas divididas para cada assunto, ou seja, para não ocorrer o risco de conflitos entre documentações, cada módulo possuía uma área separada para cada assunto. Um exemplo que pode ser utilizado é a existência de uma pasta geral para a documentação, no qual cada módulo possuía sua própria pasta dentro dessa pasta principal para trabalhar sem se preocupar com a documentação de outro módulo, e esse modelo serve igualmente para os outros assuntos como o do desenvolvimento, relatórios e trabalhos de conclusão. A única pasta que se diferencia das outras em relação a esse padrão, é a pasta do desenvolvimento, pois na etapa final ocorreu a integração de todos os códigos programados, mas esse processo de junção passou por uma série de etapas no qual todos os desenvolvedores tiveram participação e conscientização do ocorrido, fazendo com que não tivesse perigo de desentendimento após a união dos documentos. A seguir, existe a representação de exemplos com imagens retiradas do repositório SVN, mostrando como foi feita essa divisão de pastas de uma maneira visual e mais precisa, representando todo esse processo explicado de pastas dentro de pastas e sua organização.

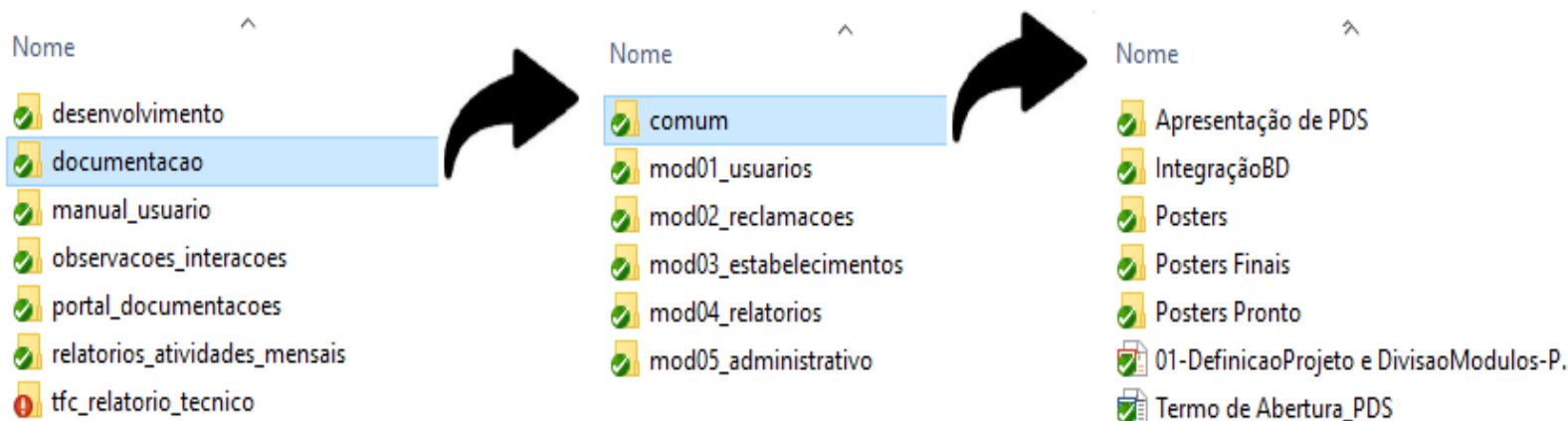
Figura 4: Print da documentação do projeto, mostrando a divisão de pastas do desenvolvimento



Fonte disponível em: **Repositório do projeto Reclame São João.**

Nessa imagem, é mostrado como foi feito a divisão das pastas dos desenvolvedores, no qual existia uma pasta principal “desenvolvimento” e dentro dela cada módulo possuía sua documentação de trabalho, e no final das contas tudo foi unido e jogado na pasta “comum”, que é onde se encontra a união de tudo que foi trabalho por todos os módulos durante o ano todo.

Figura 5: Print da documentação do projeto, mostrando a divisão de pastas da documentação.

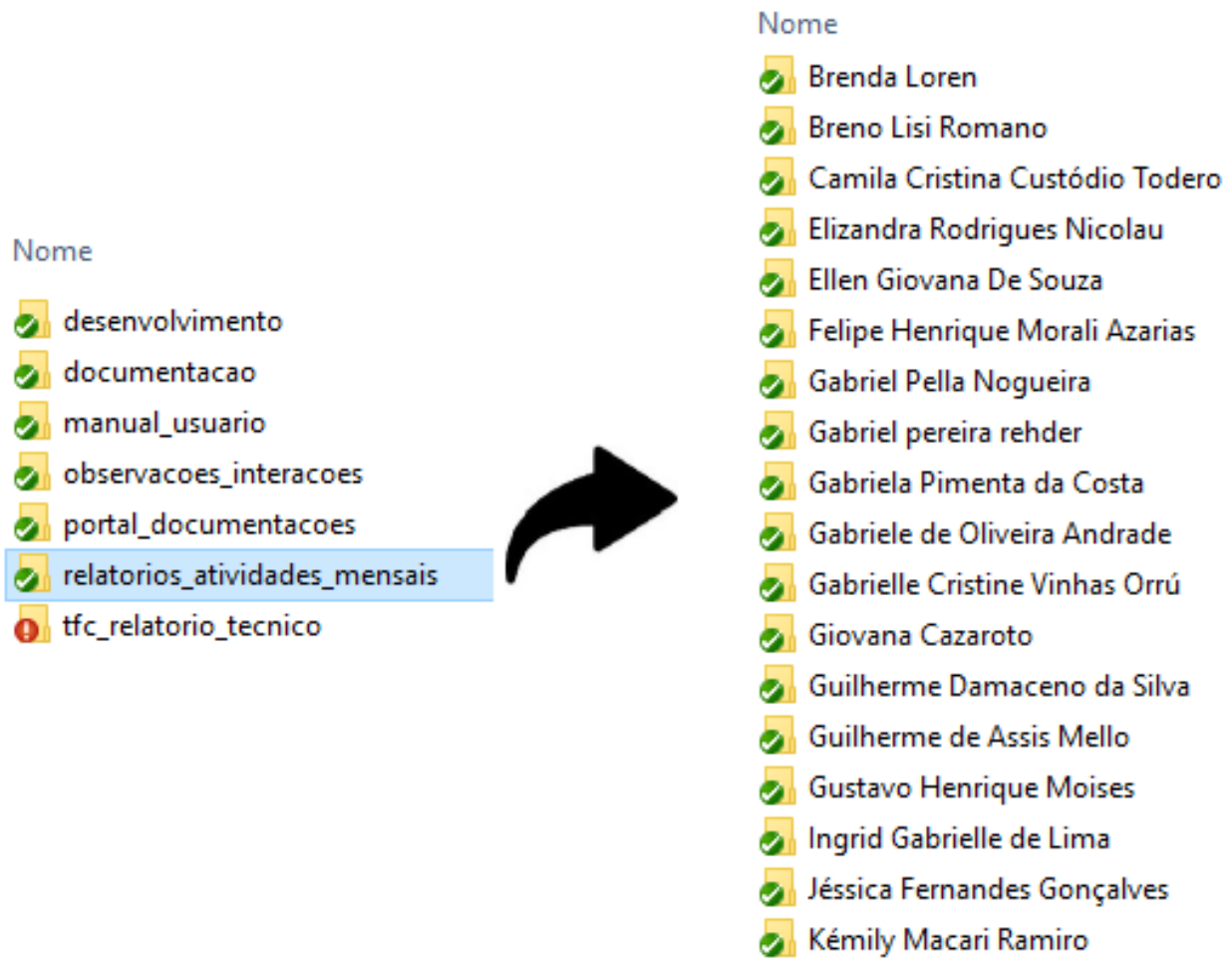


Fonte disponível em: **Repositório do projeto Reclame São João.**

Já nessa imagem, que mostra a divisão das pastas da documentação, também existe uma pasta individual para cada módulo trabalhar, e também possui uma pasta “comum”, mas que diferente da pasta de desenvolvimento, não teve a união de todas as informações em uma coisa só,

tendo dentro, portanto, apenas documentos no qual todos os módulos poderiam usar, como apresentação de pôster e apresentações feitas no final de cada bimestre, para apenas ficar em registro.

Figura 6: Print da documentação do projeto, mostrando a divisão de pastas dos relatórios.



Fonte disponível em: **Repositório do projeto Reclame São João.**

E por último, nas pastas de relatórios e atividades individuais de cada aluno, foi feita uma divisão bem simples, no qual cada aluno possuía uma pasta com seu próprio nome e com suas informações e documentações dentro dela.

2.1.11 - Os responsáveis do projeto pelo repositório SVN

O projeto Reclame São João, envolve outras ferramentas de trabalho além do SVN para questões de organização, utilidade e controle, que são ferramentas como o Kanbam e Redmine e que devem estar sempre atualizadas para que se possa ter um bom gerenciamento do desenvolvimento do trabalho, e para ser feita essas atualizações precisou ser feita uma divisão entre os membros de cada módulo, de modo que existisse uma coerência entre as responsabilidades de gerência. Cada módulo teve liberdade para escolher qual ferramenta gostaria de ficar responsável, entrando em um consenso de que os desenvolvedores deveriam ficar com o SVN por estarem sempre fazendo alterações e modificação diretamente no repositório e os analistas e DBA's, que se entendem melhor com a área de documentos deveriam se responsabilizar pela área de anotações, lembretes e gráficos de procedimento trabalhado, que são as principais funções das novas ferramentas apresentadas, também é importante lembrar que por módulo deveria existir pelo menos um aluno responsável pelas relações interpessoais do grupo, que seriam os alunos responsáveis pelo meio de comunicação entre os integrantes de cada módulo, para sempre manterem uma mesma linha de raciocínio durante o desenvolvimento do projeto, e todas essas informações foram salvas em uma tabela, para que não existissem problemas, que será apresentada logo abaixo.

Figura 7: Tabela apresentando a divisão das responsabilidades de gerências.

- **Controle e Acompanhamento de Atividades (Redmine)**
 - Mod 01: Jessica e Kemilly
 - Mod 02: Leticia e Gabriela Pimenta
 - Mod 03: Gabriele Andrade e Guilherme Damaceno
 - Mod 04: Guilherme de Assis Mello e Larissa Ribeiro
 - Mod 05: Amanda Souza e Ingrid Lima
- **Controle e Acompanhamento de Atividades (Kanbam)**
 - Mod 01: Matheus e Rodne
 - Mod 02: Giovana e Gabriel
 - Mod 03: Brenda e Felipe
 - Mod 04: Lucas Eduardo e Mayara Pires
 - Mod 05: Amanda Souza e Winnie Cristina
- **Relações Interpessoais entre os Grupos**
 - Mod 01: Ailton
 - Mod 02: Ellen e Beatriz Peixoto
 - Mod 03: Rafael Camilo e Gabrielle Orru
 - Mod 04: Luisa Villela e Marcus Vinicius
 - Mod 05: Luiz Fernando e Elizandra
- **Controle de Versões dos Artefatos (Subversion)**
 - Mod 01: Camila e Beatriz Barros
 - Mod 02: Rafael Pontes e Alexandre
 - Mod 03: Lilian e Felipe
 - Mod 04: Guilherme de Assis Mello e Marcus Vinicius
 - Mod 05: Gustavo e Gabriel Redher

Fonte disponível em: <<https://sites.google.com/site/blromano/disciplinas/pds2014>> Acesso em:

16 de outubro de 2017.

2.1.12 - Outras ferramentas utilizadas no gerenciamento do projeto

Como dito anteriormente, as ferramentas kanbam e redmine também foram utilizadas durante o desenvolvimento do projeto Reclame São João, e por este motivo é inevitável não citar nada sobre elas neste trabalho. A ferramenta kanbam é um auxílio que serve para manter a organização das tarefas feitas durante o projeto, a maneira como ela funciona é como se fosse um painel com todas as informações sobre o que está planejado para ser feito, se alguma tarefa está em andamento ou se já foi finalizada, e também é útil para ter um melhor controle do tempo das tarefas, podendo ter uma melhor forma de se organizar e verificar em qual etapa do projeto cada grupo da sala está trabalhando. A imagem a baixo representa os dados salvos do projeto Reclame São João através da ferramenta kanbam, no qual é possível ver que existiam na época 63 tarefas para se desenvolver, 11 em desenvolvimento, 4 em fase de testes e 108 finalizadas.

Figura 8: Imagem representando o diretório geral do kanbam em relação ao projeto Reclame São João.

Meus projetos (1)

Id	Projeto	Colunas
#7	<input type="checkbox"/> <input type="checkbox"/> PDS-Vespetino-2017	63 Backlog 11 Desenvolvimento 4 Verificação / Testes 108 Finalizado

Minhas tarefas (7)

▲ Id	Projeto	Tarefas	Rastreamento de tempo	Data de vencimento
#395	PDS-Vespetino-2017	CasosTeste-MOD02	30h gasto 32h estimado	12 junho 2017
#470	PDS-Vespetino-2017	Desenvolvimento e Testes do Caso de Uso #01 (Pesquisar e ...	28h gasto 28h estimado	26 junho 2017
#472	PDS-Vespetino-2017	Desenvolvimento e Testes do Caso de Uso #03 (Avaliar Estab...	28h estimado	28 agosto 2017
#473	PDS-Vespetino-2017	Desenvolvimento e Testes do Caso de Uso #05 (Exibir reputa...	28h estimado	25 setembro 2017
#474	PDS-Vespetino-2017	Desenvolvimento e Testes do Caso de Uso #06 (Gerenciar re...	28h estimado	25 setembro 2017
#478	PDS-Vespetino-2017	Desenvolvimento e Testes do Caso de Uso #10 (Pesquisar es...	28h estimado	30 outubro 2017
#498	PDS-Vespetino-2017	MOD02_CasosDeUsoBancoLocal(Pesquisar e exibir empresa)	20h estimado	12 junho 2017

Minhas subtarefas (0)

Fonte disponível em: < <https://sbv.ifsp.edu.br/kanboard/?controller=auth&action=login> > Acesso em: 01 de novembro de 2017.

Já em relação a ferramenta redmine, ela apresenta algumas funções diferentes, mas que deve estar sempre atualizada juntamente ao kanbam, pois uma mesma tarefa deve estar presente nas duas plataformas, tanto no kanbam como no redmine. Já que o redmine tem a função mais ligada na questão de prazos e tempo, é possível ver na imagem a baixo por exemplo, os campos que deveriam ser preenchidos para adicionar uma nova tabela ao redmine, no qual a tarefa deve ser toda

especificada, e focar em principalmente na prioridade de produção da tarefa e em quanto tempo ela deve ser resolvida, ou seja, o redmine auxilia os alunos na questão de quais tarefas tem mais prioridade e devem ser resolvidas com maior antecedência ou não, e isso tudo é representado em forma de tabela ou gráfico.

Figura 9: Imagem representando como seria a criação de uma nova tarefa do redmine.

The image shows the 'Nova tarefa' (New task) form in Redmine. The form is organized into several sections:

- Top Navigation:** Tabs for 'Visão geral', 'Atividade', 'Tarefas', and 'Nova tarefa'.
- Form Fields:**
 - Tipo ***: A dropdown menu with 'Bug' selected.
 - Título ***: A text input field.
 - Descrição**: A rich text editor with various formatting icons (bold, italic, underline, strikethrough, link, unlink, list, table, etc.).
 - Situação ***: A dropdown menu with 'Novo' selected.
 - Prioridade ***: A dropdown menu with 'Normal' selected.
 - Atribuído para**: A dropdown menu for assigning the task to a user.
 - Início**: A date field with '2017-11-03' entered.
 - Data prevista**: A date field.
 - Tempo estimado**: A field for estimated time in hours.
 - % Terminado**: A dropdown menu showing '0 %'.
 - Arquivos**: A section with an 'Escolher arquivo' button and a note 'Nenhum arquivo selecionado'. Below it, a link says 'Adicionar outro arquivo (Tamanho máximo: 5 MB)'.
 - Descrição opcional**: A text input field.
- Bottom Navigation:** Buttons for 'Criar', 'Criar e continuar', and 'Pré-visualizar'.

Fonte disponível em: <<http://200.133.203.29/redmine/projects/pds2017>> Acesso em: 01 de novembro de 2017.

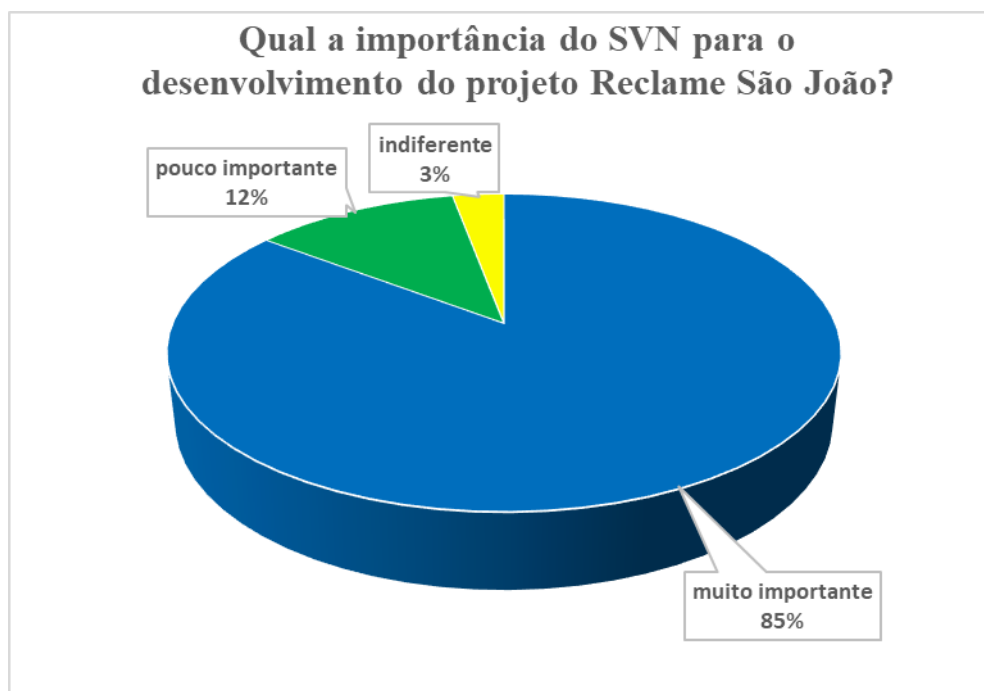
2.1.13 – Pesquisa sobre o SVN

Para o auxílio e o término desta pesquisa, foi bastante relevante a iniciação de uma pesquisa com os envolvidos diretamente com a ferramenta SVN, que no caso são os alunos envolvidos no projeto Reclame São João, do quarto ano de informática. E para ter acesso às informações sobre a opinião de cada aluno, foi necessário a criação de um formulário do google, formado com respostas simples e rápidas, no qual foi passado para os alunos e consequentemente respondido por eles, com um resultado final de 34 respostas registradas. A partir dessas respostas, foi possível analisar a

opinião geral da sala em forma de gráficos, e para ter uma melhor noção disso, algumas perguntas que foram consideradas mais relevantes serão apresentadas a seguir.

Como dito anteriormente, várias perguntas foram feitas a partir do formulário, mas algumas perguntas foram consideradas mais relevantes para serem apresentadas neste trabalho, que se resultam no total em 3, e a seguir é a apresentada a primeira delas, no qual os alunos respondem sobre qual a importância na opinião deles o controle de versões teve para o desenvolvimento do projeto, no qual 85% das respostas disseram ter sido muito importante, e os outros 15% não acharam muito importante ou indiferente, como é apresentado em gráfico a seguir:

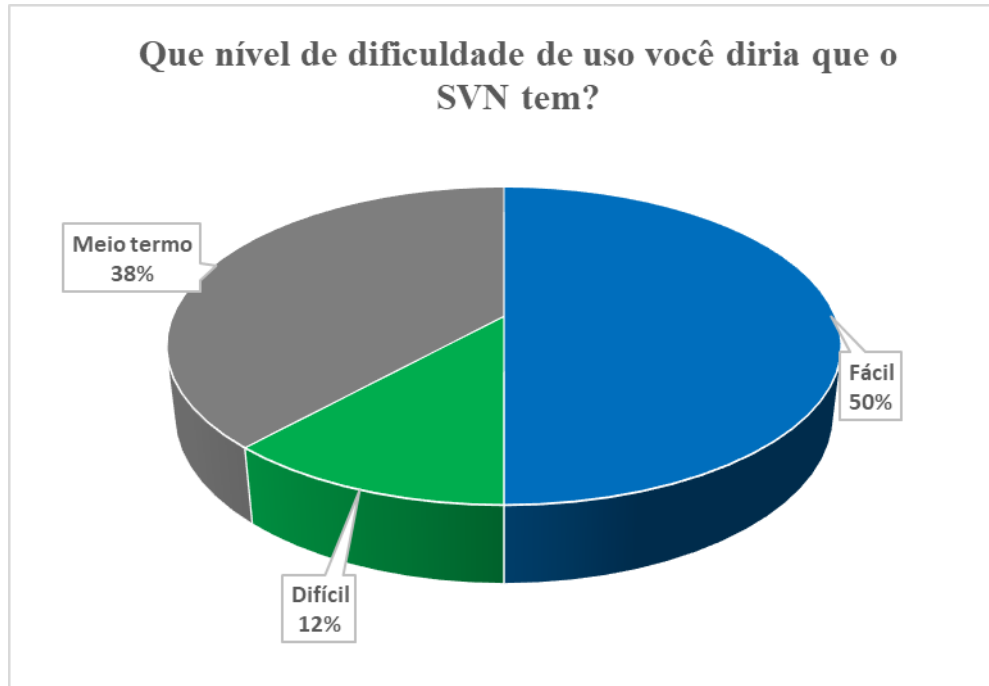
Gráfico 1



A partir do gráfico 1, é possível ver como uma ferramenta para o controle de versões, no caso o SVN, é importante para o desenvolvimento de um projeto, e isso é algo retratado pelos próprios alunos envolvidos no projeto.

O próximo gráfico, apresenta a resposta pelos alunos sobre qual o grau de dificuldades eles consideram que o uso da ferramenta SVN possui, no qual foi possível ver que já existe uma diferença menor entre as respostas, pois metade das respostas disseram ser de fácil uso, enquanto a outra metade ficaria entre um meio termo de dificuldade ou difícil.

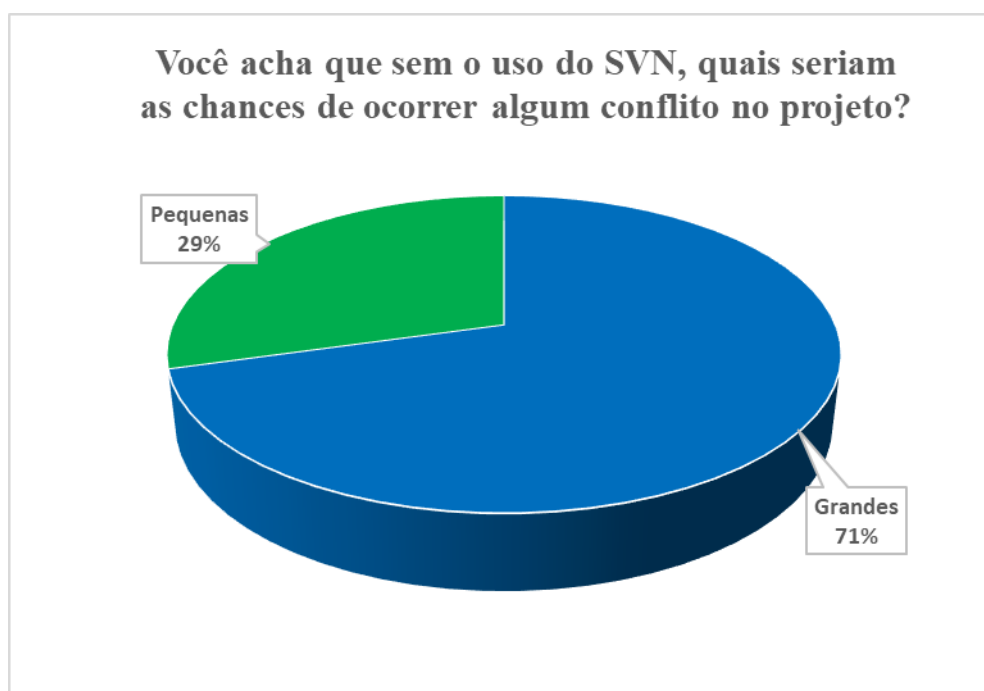
Gráfico 2



A partir do gráfico 2, é possível enxergar que mesmo a sala tendo uma primeira experiência com essa ferramenta de controle de versões, apenas 12% não conseguiram se adaptar completamente com a ferramenta e declarou ter um difícil uso, o que mostra a grande eficiência que a ferramenta pode apresentar em um trabalho.

O próximo e também último o gráfico, mostra as respostas que os alunos deram em relação a pergunta sobre os conflitos que o projeto Reclame São João vir a surgir, caso não existisse um controle de versões, no qual a grande maioria com um número de quase três quartos da sala responderam que existiram grandes chances de acontecer algum conflito sem o uso do controle de versões.

Gráfico 3



Com as informações obtidas com esse gráfico, é possível ver realmente a grande importância que o SVN busca em um projeto, e como os alunos perceberam a influência que o SVN tem com o projeto em relação à organização e controle das informações do projeto, evitando sempre qualquer tipo de grande conflito.

3 Conclusões e Recomendações

O principal objetivo apresentado por essa pesquisa, foi mostrar como a ferramenta de trabalho SVN foi importante para a contribuição do desenvolvimento do projeto Reclame São João, desenvolvido pelos alunos do quarto ano de informática, que poderá futuramente despertar interesses por envolvidos com o assunto ou não.

O propósito principal para que isso tenha sido feito, foi usar um método no qual tivesse a apresentação do funcionamento em si da ferramenta, mostrando todos os seus aspectos e características, desde a explicação sobre o que é realmente um controle de versões, envolvendo assuntos sobre como ele influencia em projetos e trabalhos na vida das pessoas, até a elaboração de gráficos que comprovaram essas características.

Junto a isso, também foi mostrada a explicação sobre qual foi o papel direto da ferramenta SVN com desenvolvimento do projeto Reclame São João, que contou com uma apresentação da pesquisa sobre a sua particularidade com o projeto, no qual foram envolvidos vários aspectos dos comandos da ferramenta junto com seu repositório, que foram características essenciais abordadas durante essa pesquisa.

Com o decorrer do projeto Reclame São João, não ocorreram falhas ligadas ao tema abordado dessa pesquisa, ou seja, o repositório SVN foi de sucesso total com o auxílio da reprodução do projeto, o que foi de grande incentivo e conforto para a reprodução dessa dissertação, e que certamente contribuiu com os resultados finais obtidos com esse tratado acadêmico, que apresenta de forma prática os conceitos básico sobre a ferramenta, e que eventualmente, atrai os interessados pelo assunto ou as pessoas interessadas em conhecer as mecânicas utilizadas por trás do site depois de finalizado.

As principais dificuldades que essa pesquisa apresentou durante seu desenvolvimento, foi em relação a própria inovação dos assuntos que foram apresentados no último ano do curso, ou seja, a falta de experiência que os alunos possuíam em relação ao assunto e que caso já tivessem sido apresentados e familiarizados antes, não seria algo tão inovador e revolucionário, o que com certeza não resultaria em algum tipo de limitação tão grande no aspecto relacionado ao esforço dos alunos, e então seria possível ter tido uma apresentação com um grau de dificuldade bem menor em relação a desenvolvimento e produção prática.

Em geral, o ideal a ser apresentado nessa monografia foi muito bem trabalhado, no qual se teve um trabalho com ótimos referenciais e formatação, apresentação com grande influência dos dados retirados do próprio projeto, mostrando dominação e controle do tema, e que futuramente

pode ser retrabalhado por outros interessados pelo meio, levando em conta que as orientações expressas nesse documento devem ser colocadas em práticas em um ambiente de trabalho real, ou seja, essa pesquisa teve uma representação de como a ferramenta de trabalho SVN foi importante para o apoio dos alunos do quarto ano de informática do Instituto Federal de São Paulo, campus que localizado em São João da Boa Vista, mas que pode ser trabalhado por qualquer outro usuário em qualquer local do planeta e que queira colocar em prática ou aprimorar o tema discutido nesse trabalho.

4 Referências Bibliográficas

[1] Moraes, Rodolfo. **CollabDev: Gerenciador de Repositórios para Ambientes Colaborativos de Desenvolvimento**, Univem – 02 de dezembro de 2013. Disponível em: <<http://aberto.univem.edu.br/handle/11077/991>>. Acesso em: 21 de setembro de 2017.

[2] Freitas, Daniel. **Análise comparativa entre Sistemas de Controle de Versões**, Universidade Federal de Juiz de Fora – 01 de dezembro de 2010. Disponível em: <<http://www.ufjf.br/getcomp/files/2013/03/An%C3%A1lise-Comparativa-entre-Sistemas-de-Controle-de-Vers%C3%B5es-Daniel-Tannure-Menandro-de-Freitas.pdf>>. Acesso em: 29 de setembro de 2017.

[3] Palestino, Caroline. **Estudo de Tecnologias de controle de versões de softwares**, Acervo Digital – 02 de dezembro de 2015. Disponível em: <<http://www.acervodigital.ufpr.br/handle/1884/41087>>. Acesso em: 06 de outubro de 2017.

[4] Feitosa, Samuel. **Tutorial Subversion**, Samuca – 12 de abril de 2007. Disponível em: <<https://samuca.wordpress.com/2007/04/12/tutorial-subversion>>. Acesso em: 11 de outubro de 2017.

[5] Tortoise, **Conceitos básicos de controle de versão**, TortoiseSvn. Disponível em <https://tortoisesvn.net/docs/nightly/TortoiseSVN_pt_BR/tsvn-basics.html>. Acesso em 30 de outubro de 2017.