

**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

**Controle de Versões dos arquivos desenvolvidos no *Projeto Mais Saúde*  
*São João***

Aluno: Larissa Rodrigues Bruno

Prontuário: 1520431

São João da Boa Vista – SP

2018

## Resumo

Este trabalho aborda a utilização de~~o~~o objetivo geral a ferramenta de Controle de Versão SVN no *Projeto Mais Saúde São João*, visando destacar os pontos negativos e positivos de sua adoção. Sendo dividido em uma ~~série~~ca de etapas dentro do objetivo específico, como: apresentação da ferramenta SVN; apresentação da utilização das operações básicas do Controle de Versão dentro do *Projeto Mais Saúde São João*, que comporta a explicação das seguintes operações: *Checkout*, *Update*, *Add*, *Commit* e *Show Log*; e análise dos pontos positivos e negativos da ferramenta e seus impactos no projeto. Será possível identificar como aspecto positivo, a possibilidade do desenvolvimento de um grande projeto integrado, realizado em infraestruturas diferentes. Entretanto, existem~~será relatado alguns~~ aspectos negativos como a não compreensão do conceito da ferramenta, ocasionando em erros em sua utilização, como nas operações de *Commit*, *Add*, *Update*, exclusão e outras – feitas de forma incorreta. Por fim, os principais resultados destacados~~obtidos~~ pelo uso da ferramenta SVN no projeto foram o controle de versões dos arquivos, o controle do histórico de *Commit* de cada usuário, a consolidação e segurança dos novos arquivos inseridos no repositório e o auxílio para o trabalho em conjunto de todos os membros da equipe.

## Sumário

1	Introdução .....	4
1.1	Objetivo Geral .....	6
1.2	Objetivos Específicos .....	6
2	Desenvolvimento .....	7
2.1	Levantamento Bibliográfico.....	7
2.1.1	Controle de Versão .....	7
2.1.2	Arquitetura de Controle de Versão Centralizado .....	7
2.1.3	Arquitetura de Controle de Versões Distribuído .....	8
2.1.4	Ferramenta Subversion - SVN.....	9
2.1.5	Ferramenta Subversion - GIT .....	10
2.2	Etapas para o Desenvolvimento da Pesquisa .....	11
2.2.1	Apresentar o SVN no <i>Projeto Mais Saúde São João</i> .....	11
2.2.2	Apresentar a operação <i>Checkout</i> no SVN do <i>Projeto Mais Saúde São João</i> .....	12
2.2.3	Apresentar a operação <i>Update</i> no SVN do <i>Projeto Mais Saúde São João</i> .....	13
2.2.4	Apresentar a operação <i>Add</i> no SVN do <i>Projeto Mais Saúde São João</i> .....	14
2.2.5	Apresentar a operação <i>Commit</i> no SVN do <i>Projeto Mais Saúde São João</i> .....	15
2.2.6	Apresentar a operação <i>Show Logs</i> no SVN do <i>Projeto Mais Saúde São João</i> ..	16
2.2.7	Análise dos Pontos Positivos e Negativos da adoção do SVN no <i>Projeto Mais Saúde São João</i> .....	17
3	Conclusões e Recomendações .....	19

## 1 Introdução

Vive-se hoje em um mundo cada vez mais tecnológico, onde constantemente são encontradas soluções para diversos problemas por meio da tecnologia em plataformas webs, aplicativos celulares, desenvolvimento de sistemas embarcados [1] e outros recursos computacionais, sendo utilizados de múltiplas formas, com diversos fins [2] e em diferentes áreas, como a cultural, tecnológica, social, entre outras.

Como exemplo disso, novos recursos em plataformas *mobiles* são disponibilizados para soluções na área da saúde, como ferramentas que trazem lembretes para beber água, fazer exercícios, medir calorias em alimentos e/ou controlar a ficha pessoal médica do usuário, como o *Apple Watch*<sup>1</sup> – por exemplo. No entanto, a maioria desses aplicativos por serem pagos, excluem aqueles com menos recursos de usufruírem dos serviços que, sempre disponíveis na palma da mão, trazem facilidades como a economia de tempo e o bem-estar físico do usuário [3].

Pensando nisso, a disciplina de Projeto de Desenvolvimento de Sistemas (PDS) – contida na grade dos 4º anos do curso Técnico em Informática Integrado ao Ensino Médio, do Instituto Federal de Educação, Ciência e Tecnologia de São Paulo campus São João da Boa Vista – visa desenvolver a cada ano um sistema que atenda às necessidades da comunidade externa [4].

O seguinte Trabalho Final de Curso tratará, portanto, do sistema *Mais Saúde São João* desenvolvido integralmente por quarenta e nove alunos – esses que concluirão o ~~e~~Curso ~~técnico~~ Técnico integrado-Integrado em 2018. Desta forma, a criação de todo o *software*, desde o desenho do esquema lógico, passando pela documentação e a codificação, foram dirigidos pelo coordenador do projeto Prof. Breno Lisi Romano, visando suprir na cidade de São João da Boa Vista, SP, Brasil, a necessidade de uma aplicação ~~mobile~~ gratuita que contenha em sua plataforma as áreas de nutrição e atividade física organizadas em um design de rede social para melhor navegação dos usuários [5].

Pensando em uma organização mais eficaz do sistema, visto que participam aproximadamente cinquenta alunos, o *Projeto Mais Saúde São João* foi dividido em nove módulos, contendo cada um deles, objetivos específicos dentro de diversas áreas – todas integradas entre si. Assim, a relação entre os módulos e suas principais funcionalidades estão especificadas a seguir:

- O Módulo I responsável pelos Usuários
- O Módulo II responsável pela Rede Social
- O Módulo III responsável pelos *Checkups*

1 <<https://www.apple.com/br/apple-watch-series-4/health/>>

- O Módulo IV responsável pelos Treinos
- O Módulo V responsável pelos Resultados dos Treinamentos
- O Módulo VI responsável pelas Ferramentas Esportivas
- O Módulo VII responsável pelo Plano Alimentar/Cardápio
- O Módulo VIII responsável pelo Diário de Bordo Nutricional
- O Módulo IX responsável pelas Ferramentas Nutricionais

Desta forma, as funcionalidades desenvolvidas ao longo do projeto foram distribuídas entre as equipes de cada módulo, conforme mostra a Figura 1.

Figura 1. Macro requisitos dos Módulos do *Projeto Mais Saúde São João*

MAIS SAÚDE SÃO JOÃO - PRODUTO A SER ENTREGUE				
SUBSISTEMAS	MÓD. 1 - USUÁRIOS	MÓD. 2 - REDE SOCIAL	MÓD. 3 - CHECKUPS	MÓD. 4 - TREINOS
<b>SUBSISTEMA 1</b> USUÁRIOS	• ESPECIFICAÇÃO DE CADASTRO FÍSICAL • LOGIN E SAÍDA DA REDE • "SOBRE" DO PROJETO E DA INSTITUIÇÃO PSP • GESTÃO DE ADMINISTRADORES • GESTÃO DE PATOLOGIAS E MEDICAÇÕES • VISUALIZAÇÃO DE PATOLOGIAS E MEDICAÇÕES	• GESTÃO DE MENSAGENS COM FOTOS • OPÇÃO "QUIRIRY" PARA MENSAGENS PUBLICADAS • GESTÃO DE COMENTÁRIOS NAS MENSAGENS PUBLICADAS • OPÇÃO "DENUNCIAR" PARA MENSAGENS PUBLICADAS • PUNIÇÃO A USUÁRIO DENUNCIADO	• GESTÃO DE TESTES RELACIONADOS A SAÚDE DO USUÁRIO • ÍNDICE DE MASSA CORPÓREA • TAXA DE GORDURA • BATIMENTOS CARDÍACOS • SATURAÇÃO DE OXIGÊNIO NO SANGUE (SPO2) • PRESSÃO SANGÜÍNEA • VISÃO • CAPACIDADE PULMONAR E TAXA RESPIRATORIA • CLASSIFICAÇÃO DOS RESULTADOS DOS TESTES • DETERMINAÇÃO APROXIMADA DA SAÚDE DO USUÁRIO • GESTÃO DE RELATÓRIOS TABULARES E GRÁFICOS DOS RESULTADOS DOS TESTES	• GESTÃO DA INTERAÇÃO USUÁRIO - EDUCADOR FÍSICO • AGENDAMENTO DE CONSULTAS PRESENCIAIS • GESTÃO DE DADOS REFERENTES A CONSULTA • GESTÃO DE FICHAS DE TREINAMENTO • GESTÃO DE MEDIDAS CORPORAIS E DOBRAS CUTÂNEAS • ENVIO DE NOVAS FICHAS DE TREINAMENTOS VIA E-MAIL • GESTÃO DE SUPLEMENTOS ALIMENTARES
<b>SUBSISTEMA 2</b> ATIVIDADES FÍSICAS	• EDIÇÃO DE DADOS PESSOAIS DOS USUÁRIOS • GERENCIAMENTO DO CONTROLE DE ACESSO • RECUPERAÇÃO DE SENHA • PÁGINAS DE RECEPÇÃO • FALE CONOSCO COM DÚVIDAS E FEEDBACK	• GESTÃO DE ORIENTAÇÕES NUTRICIONAIS E ATIVIDADES FÍSICAS • GESTÃO DE DICAS NUTRICIONAIS E ATIVIDADES FÍSICAS • NOTIFICAÇÕES DE NOVAS POSTAGENS		
<b>SUBSISTEMA 3</b> NUTRIÇÃO				
<b>MÓD. 5 - RESULTADOS DOS TREINAMENTOS</b>	<b>MÓD. 6 - FERRAMENTAS ESPORTIVAS</b>	<b>MÓD. 7 - PLANO ALIMENTAR/ CARDÁPIO</b>	<b>MÓD. 8 - DIÁRIO DE BORDO NUTRICIONAL</b>	<b>MÓD. 9 - FERRAMENTAS NUTRICIONAIS</b>
• GESTÃO DE DESEMPENHOS DAS FICHAS DE TREINAMENTO • GESTÃO DE DESEMPENHOS DE PROGRAMAS DE TREINAMENTO PRONTOS • GESTÃO DE HISTÓRICO DE DESEMPENHO DE TREINAMENTO • GESTÃO DE GASTO CALÓRICO DIÁRIO • RELATÓRIOS TABULARES E GRÁFICOS DE GASTO CALÓRICO VS CONSUMO CALÓRICO • RELATÓRIOS DE FADIGA MUSCULAR POR GRUPO MUSCULAR • RELATÓRIOS TABULARES E GRÁFICOS DE DOBRAS CUTÂNEAS	• GUIA DE EXERCÍCIOS FÍSICOS PESQUISA E VISUALIZAÇÃO • GUIA DE SUPLEMENTOS ALIMENTARES • GESTÃO DE UNIDADES DE MEDIDA DE ATIVIDADES FÍSICAS • GESTÃO DE TIPOS DE EXERCÍCIOS FÍSICOS • GESTÃO DE PROGRAMAS PRONTOS DE TREINAMENTO • GESTÃO DE TESTES PSICOLÓGICOS • GESTÃO DE LEMBRETES PARA AVISO DE CADASTRO DE RESULTADOS DE TREINAMENTO	• GESTÃO DA INTERAÇÃO USUÁRIO - NUTRICIONISTA • AGENDAMENTO DE CONSULTAS PRESENCIAIS • GESTÃO DE DADOS REFERENTES A CONSULTA • GESTÃO DE CALORIAS DIÁRIAS NECESSÁRIAS • GESTÃO DE CARDÁPIOS PERIÓDICOS INDIVIDUAIS • RELATÓRIOS TABULARES E GRÁFICOS DE CALORIAS DOS CARDÁPIOS • ENVIO DE NOVOS CARDÁPIOS AO USUÁRIO VIA E-MAIL	• CÁLCULO DO PESO IDEAL DO USUÁRIO • GESTÃO DE DIÁRIO DE BORDO NUTRICIONAL INDIVIDUALIZADO • GESTÃO DE ALIMENTOS FAVORITOS • RELATÓRIOS TABULARES E GRÁFICOS DE CALORIAS CONSUMIDAS VS CALORIAS RESTANTES (DIÁRIO) • RELATÓRIOS TABULARES E GRÁFICOS DE CALORIAS POR REFEIÇÃO E NUTRIENTES (DIÁRIO) • HISTÓRICO TABULAR DE NUTRIENTES E CALORIAS • ATUALIZAÇÃO DE PESO DO USUÁRIO • HISTÓRICO POR GRÁFICO DE PESO	• PESQUISA CALÓRICA DE ALIMENTOS • GESTÃO DE UNIDADES DE MEDIDA DOS ALIMENTOS • GESTÃO DE TIPOS DE ALIMENTOS • GESTÃO DE INFORMAÇÃO NUTRICIONAL DOS ALIMENTOS • GESTÃO DE "MEUS ALIMENTOS" • GESTÃO DE RECEITAS E SEU COMPARTILHAMENTO • AUTORIZAÇÃO PARA PUBLICAR RECEITAS • GESTÃO DE LEMBRETES PARA AVISO DE CADASTRO DE DIÁRIO DE BORDO

Fonte: Prática de Desenvolvimento de Sistemas (PDS) [6]

Observando-se a complexidade da estrutura do *Projeto Mais Saúde São João* e sua distribuição para a integração posterior, muitos problemas de desenvolvimento de *software* poderiam ocorrer pela falta de Controle de Versão, como os códigos subscritos, as alterações perdidas – feitas por outro membro da equipe –, as dificuldades em saber distinguir quais alterações foram efetuadas em um programa, quando e por quem foram feitas., entre outras. ~~enfim~~ Enfim.

todos os problemas em manter o histórico das versões do sistema, realizado por todos os usuários ao mesmo tempo. Em virtude disso, decidiu-se pela utilização do modelo de Controle de Versão Centralizado no projeto.

Portanto, será abordado neste trabalho o uso do Sistema de Controle de Versão *Subversion* (SVN) e suas principais operações e funcionalidades – essenciais para a viabilidade do projeto.

### **1.1 Objetivo Geral**

O objetivo geral deste trabalho é apresentar a aplicação de Controle de Versão Centralizado no *Projeto Mais Saúde São João*, através da ferramenta SVN, visando destacar os pontos positivos e negativos de sua adoção.

### **1.2 Objetivos Específicos**

Os objetivos específicos necessários para a clara compreensão e realização do trabalho foram divididos em subtópicos, abordando os principais assuntos e operações utilizadas pela ferramenta SVN no *Projeto Mais Saúde São João*. Entre elas estão:

1. Apresentação do SVN
2. Apresentação da utilização das Operações básicas do Controle de Versão no *Projeto Mais Saúde São João*, sendo:
  - 2.1 *Checkout*
  - 2.2 *Update*
  - 2.3 *Add*
  - 2.4 *Commit*
  - 2.5 *Show Log*
3. Análise dos pontos positivos e negativos e os impactos decorrentes

## 2 Desenvolvimento

O desenvolvimento deste trabalho tem como objetivo explicar a utilização, a importância e as atividades desenvolvidas pela ferramenta SVN e suas operações dentro do *Projeto Mais Saúde São João*, utilizando-se, para isso, o levantamento bibliográfico e as etapas para o desenvolvimento da pesquisa.

### 2.1 Levantamento Bibliográfico

Por esse intermédio, apresentam-se os conceitos de suma importância para o pleno entendimento e desenvolvimento da pesquisa, incluindo os principais tipos de Controle de Versão, sendo eles Centralizado e Distribuído, além de algumas de suas ferramentas específicas, como o GIT e o SVN.

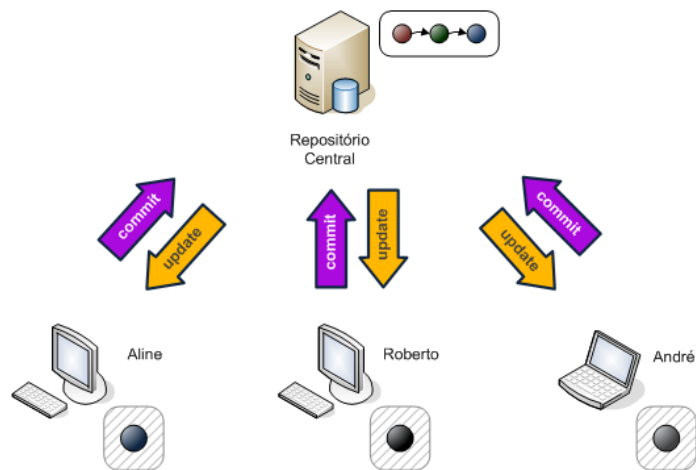
#### 2.1.1 Controle de Versão

O Controle de Versão é utilizado de forma recorrente na Engenharia de *Software*, tendo como objetivo organizar projetos por meio do gerenciamento de diferentes versões de um documento. Assim, é possível acompanhar o histórico de desenvolvimento, desenvolver paralelamente e até resgatar o sistema de um ponto já fixo – sem alterar a versão principal. O Controle de Versão é responsável por administrar as variáveis da informação, sendo muito utilizado por desenvolvedores; é o modo mais eficaz e seguro de garantir a manutenção do código-fonte – principalmente quando se trata de grandes equipes de projetos [7].

#### 2.1.2 Arquitetura de Controle de Versão Centralizado

O Sistema Centralizado possui apenas um servidor central, havendo, entretanto, várias cópias de trabalho em diferentes máquinas – baseados na arquitetura cliente-servidor [8]. É recomendado para empresas que trabalham com uma rede local, pois, dessa forma, a comunicação é centralizada no servidor, como mostra a Figura 2.

Figura 2. Representação de Sistema Centralizado



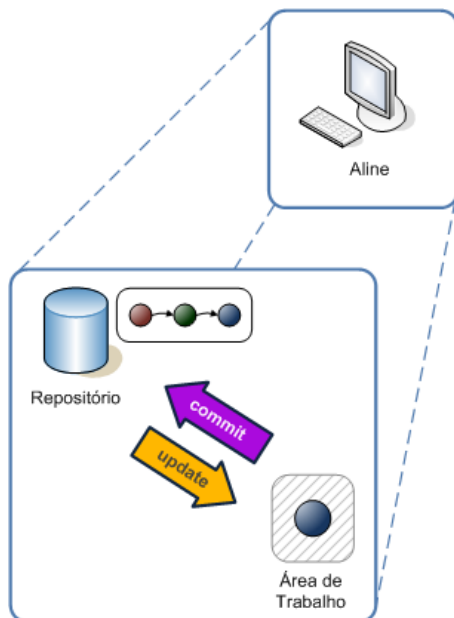
Fonte: Conceitos Básicos de Controle de Versão de *Software* [8]

### 2.1.3 Arquitetura de Controle de Versão Distribuído

No Controle de Versão Distribuído há vários servidores autônomos e independentes, que realizam as operações de *Checkin* e *Checkout* na própria máquina do usuário. É um sistema mais rápido, porém exige maior conhecimento da ferramenta, podendo inicialmente atrapalhar os desenvolvedores [7]. É, desta forma, recomendado para equipes grandes em que seus desenvolvedores trabalhem em diferentes locais, como mostra a Figura 3.



Figura 3. Representação de Sistema Distribuído



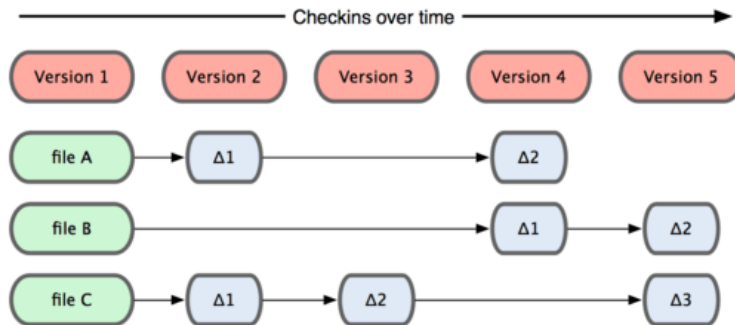
Fonte: Conceitos Básicos de Controle de Versão de *Software* [8]

#### 2.1.4 Ferramenta Subversion - SVN

O Apache Subversion (SVN) é um *software* livre de Controle de Versão e revisão de informações. Foi desenvolvido com o objetivo de manter todo o histórico corrente de versões de pastas e documentação, com o código aberto para desenvolvedores [9].

Um servidor de SVN – também chamado de "repositório" – mantém sempre uma versão atualizada e o histórico de todo o código. Assim, cada desenvolvedor pode acessar o servidor SVN utilizando um *software* que é chamado cliente SVN. O *software* necessita de um repositório central para que os usuários façam o *Checkout* e *Commit* dos arquivos versionados [8], conforme a Figura 4.

Figura 4. Representação do Esquema de versões no SVN

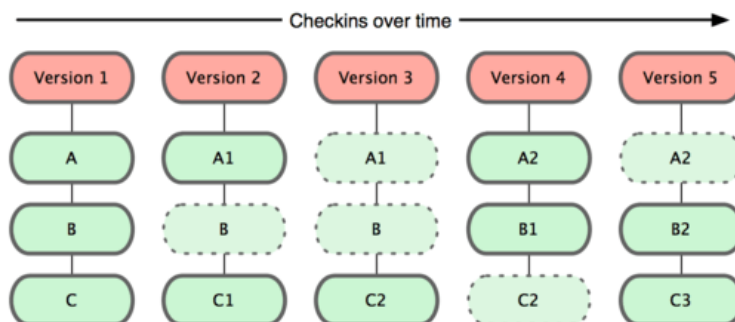


Fonte: Estudo de Tecnologias de Versões de Softwares [7]

### 2.1.5 Ferramenta GIT

O GIT trabalha considerando os dados como um conjunto de *snapshots* (captura de algo em um determinado instante, como em uma foto) de um minissistema de arquivos. A cada vez que o arquivo é salvo ou consolidado (*Commit*), o estado do projeto no GIT é semelhante à captura da imagem de todos os arquivos no momento, armazenando uma referência para essa captura. Desta forma, se nenhum arquivo foi alterado, a informação não é novamente armazenada – apenas um *link* é inserido para o arquivo idêntico ao anterior que já foi armazenado [8]. A Figura 5 representa como o GIT lida com seus dados:

Figura 5. Representação do Esquema de versões no GIT



Fonte: Estudo de Tecnologias de Versões de Softwares [7]

Geralmente, a maior parte das operações no GIT precisam apenas de recursos e arquivos locais para operar, uma vez que há o histórico do projeto no disco local do usuário. Assim, praticamente nunca é necessária a informação de outros computadores – dando a impressão de que a maior parte das operações é quase instantânea.

Nele, também é possível criar áreas completamente separadas para testes ou para projetos diferentes, desfazendo alterações que estão com problemas ou ainda voltando para a versão que estava estável, possibilitando o trabalho em equipe de maneira mais simples e segura. Pode-se assim, criar e editar documentos de projetos nos quais diversas pessoas estão contribuindo simultaneamente, sem o risco de as alterações serem sobrescritas. Por mais complexo que isso possa parecer, o GIT mantém tudo em ordem para evitar problemas para os desenvolvedores.

Portanto, o GIT é um programa de Controle de Versão de documentos que possibilita trabalhar sempre em um mesmo diretório, fazendo alterações no projeto, gravando documentação e comentários; e o sistema registra tudo o que for salvo e aprovado [10].

## **2.2 Etapas para o Desenvolvimento da Pesquisa**

Através deste, foram exibidas, exemplificadas e explicadas todas as etapas necessárias para concluir o objetivo geral do trabalho.

### **2.2.1 Apresentar o SVN no Projeto *Mais Saúde São João***

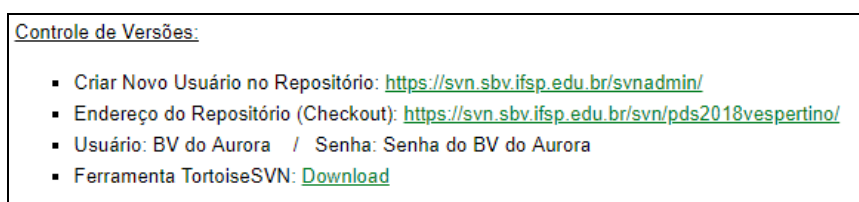
Antes de mais nada, é fundamental ressaltar que a escolha da ferramenta SVN foi feita pelo coordenador do projeto prof. Breno Lisi Romano – exclusivamente – sem participação dos alunos envolvidos no desenvolvimento do projeto, pois esses não tiveram ao longo do curso nenhuma disciplina ou introdução anterior aos diferentes *softwares* e ao próprio conceito e arquitetura de Controle de Versão, não estando aptos, portanto, a contribuir com a escolha da ferramenta utilizada no projeto.

Desta forma, em uma das primeiras aulas da disciplina de PDS, após a explicação da ideia do *Projeto Mais Saúde São João*, foram introduzidas e explicadas as técnicas que seriam utilizadas e de que forma os módulos seriam divididos; assim, a ferramenta SVN e o conceito de Controle de Versão foram apresentados pela primeira vez aos alunos envolvidos. Logo após, as principais operações foram explicadas passo a passo e de que forma seriam utilizadas, além da apresentação do que, de fato, é a ferramenta, com base na sua recorrente utilização dentro da Engenharia de *Software*.

O prof. Breno Lisi Romano, responsável pela disciplina e coordenação do projeto gerenciou a criação inicial do repositório e as demais funções necessárias até que os alunos pudessem realizar

o *Checkout* e começar a criação de suas pastas correspondentes e de seus módulos. Para que o *Checkout* fosse possível, o usuário de *login* no SVN de cada aluno foi vinculado ao seu usuário na ferramenta acadêmica Aurora, que permite aos alunos da instituição diversas vantagens e utilizações, identificando-os no sistema interno desta. Posteriormente, foi disponibilizado um local para acesso das informações vinculadas ao SVN, dentro do site pessoal da disciplina de PDS, como: criação de um novo usuário no repositório, endereço para *Checkout*, acesso pelo usuário e *link* para baixar a ferramenta SVN, conforme a Figura 6.

Figura 6. Captura das informações contidas no site

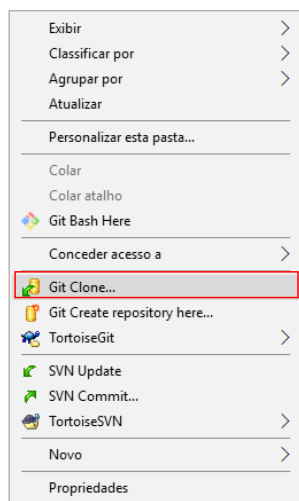


Fonte: A autora – *print* de tela do site [4]

### 2.2.2 Apresentar a operação *Checkout* do SVN no Projeto Mais Saúde São João

O *Checkout* no SVN é definido pelo “Ato de fazer *download* de um projeto para a máquina local, de modo que seus arquivos estejam vinculados ao SVN e passíveis de manipulação. O projeto para o qual será feito o *Checkout* deve existir no repositório” (LIMA, 2018)[X]. É, desta forma, uma das primeiras etapas realizadas normalmente. Quando clicado o botão direito do *mouse* em uma das pastas do SVN, a operação pode ser vista, conforme a Figura 7.

Figura 7. Visualização da operação *Git Clone* (*Checkout*)

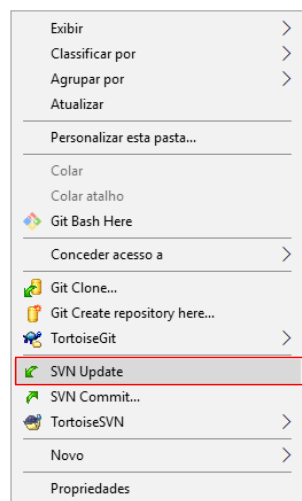


Fonte: A autora — *print de tela*

### 2.2.3 Apresentar a operação *Update* do SVN no *Projeto Mais Saúde São João*

A operação de *Update* é responsável por efetuar na máquina local do usuário as atualizações salvas no servidor do SVN [9]. EsseO servidor serve como um repositório de modificações, permitindo a qualquer momento resgatar o código a um estado anterior já salvo no servidor SVN e auxiliando, desta forma, o usuário na verificação das versões já combinadas realizadas em períodos diferentes, para a criação de uma única versão. A operação de *Update* pode ser visualizada, clicando com o botão direito do *mouse* em uma das pastas do SVN, conforme a Figura 8.

**Figura 8. Visualização da operação de *Update***

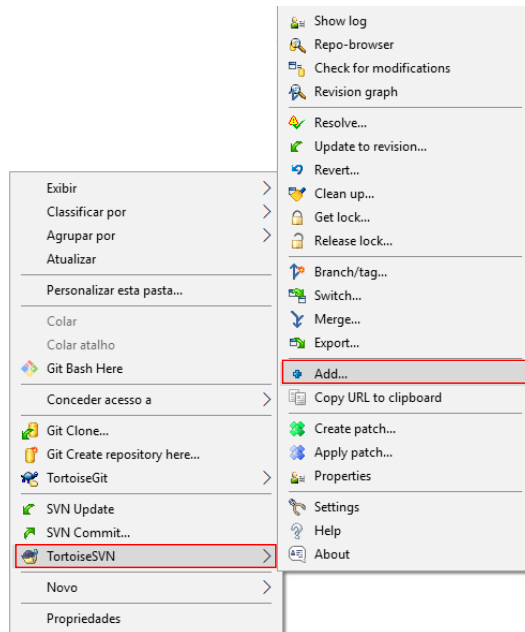


Fonte: A autora — *print de tela*

#### **2.2.4 Apresentar a operação *Add* do SVN no Projeto Mais Saúde São João**

O *Add* é a operação responsável por adicionar arquivos, diretórios ou *links* da cópia de trabalho para o repositório, fazendo a inclusão no próximo *Commit* [9]. Para que a operação seja visualizada basta clicar com o botão direito do *mouse* em uma das pastas do SVN, conforme a Figura 9.

Figura 9. Visualização da operação de Add



Fonte: A autora — print de tela

## 2.2.5 Apresentar a operação Commit do SVN no Projeto Mais Saúde São João

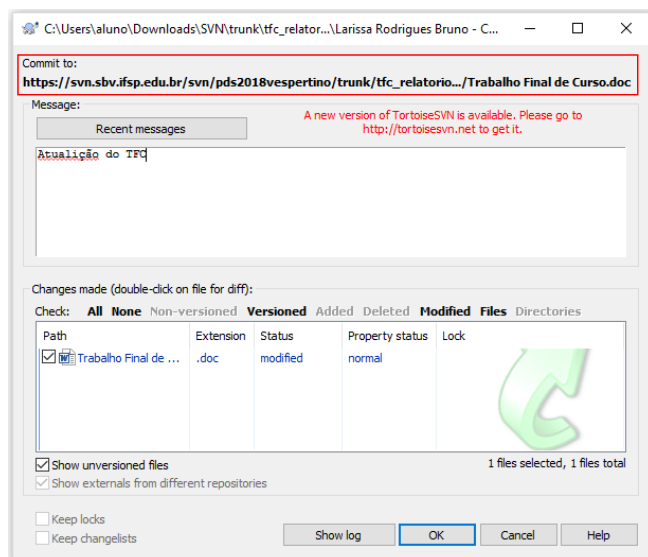
O *Commit* é a operação responsável por enviar as alterações realizadas na máquina local para o servidor SVN [9]. Ela é realizada por arquivo, permitindo, desta forma, que o SVN consiga agrupar as alterações feitas, possibilitando assim, voltar a uma revisão anterior, caso o código seja quebrado. Ou seja, quando o servidor SVN recebe uma nova versão, ele realiza várias comparações do novo código em relação a última versão e, caso haja alguma discrepância entre linhas de código, o próprio servidor SVN tentará resolver o conflito, conforme a Figura 10. Simões (2012) afirma que:

Se arquivos diferentes foram alterados, ou se classes diferentes de um mesmo arquivo foram alteradas, por exemplo, o servidor resolverá automaticamente o conflito, de forma transparente para os usuários, realizando um *merge* dos arquivos, e produzindo automaticamente um head version que é a união de ambos. Caso os dois desenvolvedores tenham, por exemplo, alterado as mesmas linhas nas mesmas classes de um código, o servidor pode não conseguir resolver o conflito. Neste caso, o usuário será alertado (no momento do *Commit*) sobre o conflito, e será solicitado a resolvê-lo manualmente.

(SIMÕES, 2012)

Comentado [AI1]: Merge - agrupamento

Figura 10. Visualização da operação de *Commit*



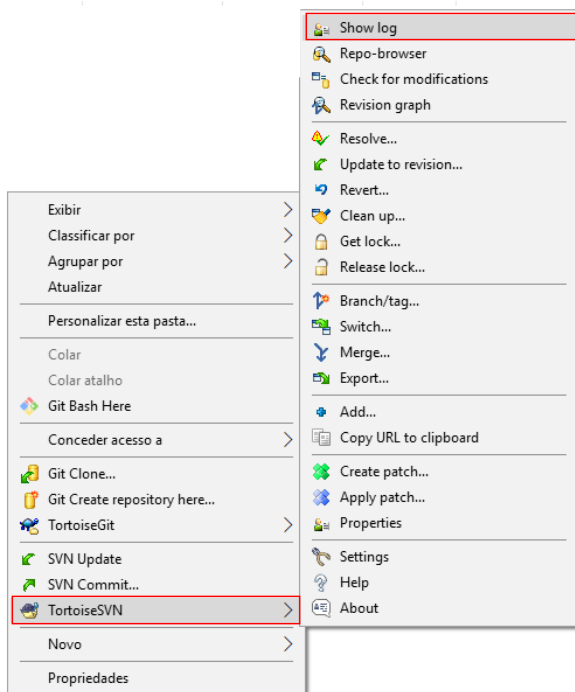
Fonte: A autora — *print de tela*

### 2.2.6 Apresentar a operação *Show Logs* do SVN no Projeto *Mais Saúde São João*

A operação de *Show Logs* é responsável por exibir as informações de data e autor anexadas às revisões; e as modificações ocorridas nelas. Desta forma, a operação retorna um histórico de registros de todas as revisões, com suas alterações atribuídas a cada usuário e até mesmo o horário da modificação, contando ainda com a mensagem inserida no *Commit* daquela revisão [9]. A visualização da operação pode ser encontrada clicando com o botão direito do *mouse* em uma das pastas do SVN, conforme a Figura 11.



Figura 11. Visualização da operação de *Show Logs*



Fonte: -A autora—*print de tela*

### 2.2.7 Análise dos Pontos Positivos e Negativos da adoção do SVN no *Projeto Mais Saúde São João*

A utilização de ferramentas que agilizam ou simplificam o desenvolvimento e a coordenação de grandes projetos trará sempre aspectos positivos e negativos em seu uso.

De forma semelhante, o uso da ferramenta SVN no *Projeto Mais Saúde São João* não fugiu dessa compreensão e trouxe em si situações de comodidade no desenvolvimento como também de conflitos – principalmente nos primeiros meses de sua utilização.

Assim, como já citado no subtópico 2.2.1, pelo desconhecimento e não utilização anterior da ferramenta por parte dos alunos envolvidos, muitos erros foram cometidos ao longo do desenvolvimento, como a não compreensão do conceito da ferramenta, levando os alunos a criar versões diferentes dos documentos dentro das pastas, ou ainda clonar pastas inteiras dentro de suas próprias pastas por não compreender e/ou não querer realizar o *Commit* de um ou outro código

atualizado, acarretando assim, em uma memória de repositório que exigia grande espaço e por vezes tornava lento o processo de *Update*.

Desta forma, a Curva de aprendizagem [11] da utilização e adaptação à ferramenta foi baixa, o que desgastou o processo de desenvolvimento, atrasando as datas de entrega das funcionalidades do *Projeto Mais Saúde São João* como um todo.

Entretanto, para um projeto que envolveu cerca de 50 alunos, seria inviável desenvolver com qualidade e segurança – não correndo o risco de códigos serem perdidos e tempo ~~jogado~~ fora desperdiçado - sem a utilização de uma ferramenta para auxiliar e controlar as versões, como já citado ao longo deste trabalho.

Sem dúvida, o SVN cumpriu a função esperada com sucesso, permitindo a alteração segura dos códigos, o resgate das versões, o controle do histórico e as demais operações básicas de que dispõem.

### 3 Conclusões e Recomendações

O objetivo geral deste trabalho foi apresentar a utilização da ferramenta SVN no *Projeto Mais Saúde São João*, visando destacar os pontos negativos e positivos de sua adoção. Foi possível, através dela, concluir com êxito a proposta do projeto que, com o Controle de Versão, possibilitou o trabalho de 49 alunos que, distribuídos em duas turmas diferentes, produziam ocasionalmente no mesmo ambiente - por falta de infraestrutura capaz de comportar semanalmente essa demanda.

Para a melhor compreensão, dividiu-se o objetivo geral em objetivos específicos, como: apresentação da ferramenta SVN; apresentação da utilização das operações básicas do Controle de Versão dentro do *Projeto Mais Saúde São João*, que comportou a explicação das seguintes operações: *Checkout*, *Update*, *Add*, *Commit* e *Show Log*; e análise dos pontos positivos e negativos da ferramenta e seus impactos no projeto.

Concluiu-se que os principais resultados obtidos da ferramenta SVN foi o controle de versões dos arquivos no projeto, o controle do histórico de *Commit* de cada usuário, a consolidação e segurança dos novos arquivos inseridos no repositório e o auxílio para o trabalho em conjunto de todos os membros da equipe.

Além disso, foi possível identificar como aspecto positivo, a facilidade e possibilidade do desenvolvimento de um grande projeto, com 49 participantes, produzindo integrados, entretanto, em infraestruturas separadas na maior parte do tempo. Somente através do uso da ferramenta tornou-se viável o controle do andamento estável e seguro do sistema, sem perdas e danos que não pudessem ser reparados, e comunicação dos desenvolvedores na codificação, sem subscrição de códigos.

Foi possível relatar também, entretanto, alguns aspectos negativos como a não compreensão do conceito da ferramenta, ocasionando em erros na utilização por parte dos próprios membros do projeto, como nas operações de *Commit*, *Add*, *Update*, exclusão e outras – feitas de forma incorreta. Sabe-se que esses aspectos foram consequência da curva de aprendizagem baixa [11], que exigiu certo tempo – prejudicial – para a adaptação e aprendizagem dos alunos sobre a utilização da ferramenta SVN.

Sugere-se, assim, para trabalhos futuros, a inclusão de uma disciplina no curso Técnico em Informática Integrado ao Ensino Médio, capaz de atender a necessidade de ensino sobre o Controle de Versão e outras ferramentas que possam auxiliar no desenvolvimento de *software*, para um melhor desempenho na disciplina de PDS. Desta forma, os alunos participantes do desenvolvimento do projeto e o coordenador responsável por ele, poderão discorrer sobre a melhor ferramenta a ser utilizada para Controle de Versão ao longo do desenvolvimento dos projetos nos próximos anos.

## Referências Bibliográficas

1. EMBEDDED ARCHITECTS TECNOLOGIA (Brasília). **O que é um sistema embarcado**. Disponível em: <<http://www.embarc.com.br/p1600.aspx>>. Acesso em: 25 out. 2018.
2. SANTA CATARINA. FUNDAÇÃO EDUCACIONAL DA REGIÃO DE JOINVILLE. (Org.). **O impacto das novas tecnologias na sociedade**. 2018. Disponível em: <<https://www.inovapark.com.br/o-impacto-das-novas-tecnologias/>>. Acesso em: 20 set. 2018.
3. JONES, Julia Mandaro Lavinias; FEITOSA, Alina Coutinho Rodrigues. **Aplicativos (apps) em saúde: o que precisamos saber?** 2018. Disponível em: <<http://www.isaudebahia.com.br/noticias/detalhe/noticia/aplicativos-apps-em-saude-o-que-precisamos-saber/>>. Acesso em: 20 set. 2018.
4. SÃO PAULO. INSTITUTO FEDERAL DE EDUCAÇÃO CIÊNCIA E TECNOLOGIA DE SÃO PAULO. (Comp.). **Técnico Integrado em Informática**. Disponível em: <<https://www.sbv.ifsp.edu.br/index.php/component/content/article/64-ensino/cursos/168-tecnico-integrado-informatica>>. Acesso em: 22 mar. 2018.
5. ROMANO, Breno Lisi. **Prática de Desenvolvimento de Sistemas (PDS)**. Disponível em: <<https://sites.google.com/site/blromano/disciplinas/pds2014>>. Acesso em: 25 set. 2018.
6. JOÃO, Equipe do Projeto Mais Saúde São. **Termo de Abertura do Projeto Mais Saúde São João**. Disponível em: <<svn.sbv.ifsp.edu.br/svn/pds2018vespertino/trunk/documentacao/comum/TermoAberturaProjeto.doc>>. Acesso em: 25 set. 2018.
7. PALESTINO, Caroline Munhoz Corrêa. **Estudo de Tecnologias de Versões de Softwares**. 2015. 72 f. TCC (Graduação) - Curso de Gestão da Informação, Universidade Federal do Paraná, Curitiba, 2015. Disponível em: <<https://acervodigital.ufpr.br/handle/1884/41087>>. Acesso em: 18 out. 2018.
8. DIAS, André Felipe. **Conceitos Básicos de Controle de Versão de Software — Centralizado e Distribuído**. Disponível em: <<https://blog.pronus.io/posts/conceitos-basicos-de-controle-de-versao-de-software-centralizado-e-distribuido/>>. Acesso em: 18 out. 2018.
9. COLLINS-SUSSMAN, Ben; FITZPATRICK, Brian W.; PILATO, C. Michael. **Controle de Versão com Subversion: Para Subversion 1.4**. Califórnia: Tba do Brasil,

2002. Disponível em: <<http://creativecommons.org/licenses/by/2.0/>>. Acesso em: 23 out. 2018.
10. CHACON, Scott; STRAUB, Ben. **Pro Git book**. Mountain View: Apress, 2009. Disponível em: <<https://git-scm.com/book/pt-br/v2>>. Acesso em: 23 out. 2018.
11. PEREIRA, Lucila Conceição. **Curva de aprendizagem**. Disponível em: <[https://www.infoescola.com/administracao/\\_curva-de-aprendizagem/](https://www.infoescola.com/administracao/_curva-de-aprendizagem/)>. Acesso em: 01 nov. 2018.