



Universidade Federal de Campina Grande

Centro de Engenharia Elétrica e Informática

Curso de Graduação em Engenharia Elétrica

IGOR DE MEDEIROS MACIEL

RELATÓRIO DE ESTÁGIO SUPERVISIONADO

Campina Grande, Paraíba
Junho de 2015

IGOR DE MEDEIROS MACIEL

RELATÓRIO DE ESTÁGIO SUPERVISIONADO REALIZADO NO
LABORATÓRIO DE INTERFACE HOMEM MÁQUINA DA
UNIVERSIDADE FEDERAL DE CAMPINA GRANDE

*Relatório de Estágio Supervisionado submetido
à Unidade Acadêmica de Engenharia Elétrica
da Universidade Federal de Campina Grande
como parte dos requisitos necessários para a
obtenção do grau de Bacharel em Ciências no
Domínio da Engenharia Elétrica.*

Área de Concentração: Controle e Automação

Orientadora:

Professora Maria de Fátima Queiroz Vieira, PhD.

Campina Grande, Paraíba
Abril 2016

IGOR DE MEDEIROS MACIEL

RELATÓRIO DE ESTÁGIO SUPERVISIONADO

*Relatório de Estágio Supervisionado submetido à
Unidade Acadêmica de Engenharia Elétrica da
Universidade Federal de Campina Grande como parte
dos requisitos necessários para a obtenção do grau de
Bacharel em Ciências no Domínio da Engenharia
Elétrica.*

Aprovado em / /

Professor Avaliador
Universidade Federal de Campina Grande
Avaliador

Professora Maria de Fátima Queiroz Vieira, PhD.
Universidade Federal de Campina Grande
Orientador, UFCG

SUMÁRIO

1	Introdução.....	5
1.1	Objetivos do Estágio.....	5
1.2	O LIHM.....	5
2	Atividades Desenvolvidas.....	7
2.1	Desenvolvimento de Instalador para o Gerador de Cenários.....	7
2.1.1	Arquitetura do Gerador de Cenários.....	7
2.1.2	Escolha de Software Instalador.....	9
2.1.3	Construção de Executável.....	9
2.1.4	Criação de template do instalador.....	9
2.1.5	Interface do Instalador.....	10
2.2	Reestruturação da rede e intranet do LIHM.....	12
2.3	Sistema de controle de versão SVN.....	13
2.3.1	Migração e melhorias no Sistema de Versionamento de Software.....	14
2.4	Suporte técnico ao LIHM.....	15
2.5	Sistema de Monitoramento de Treinamentos.....	16
2.5.1	Arquitetura proposta para o sistema.....	17
3	Considerações Finais.....	19
	Bibliografia.....	20
	APÊNDICE A – Tutorial para construção de instalador do gerador de cenários.....	21
	APÊNDICE B– Estrutura da rede do LIHM.....	25

1 INTRODUÇÃO

Este relatório aborda as atividades do estágio supervisionado desenvolvido pelo aluno do curso de Engenharia Elétrica da Universidade Federal de Campina Grande (UFCG), Igor de Medeiros Maciel, realizado no Laboratório de Interface Homem-Máquina (LIHM).

O LIHM faz parte da Unidade Acadêmica de Engenharia Elétrica do Centro de Engenharia Elétrica e Informática da UFCG entre 20 de Abril de 2015 e 05 de Junho de 2015 totalizando uma carga horária de 188h. na mesma Universidade de 20 de Abril de 2015 a 05 de Junho de 2015 totalizando uma carga horária de 188h.

Durante o estágio, foram realizadas atividades de apoio ao projeto de pesquisa e desenvolvimento Gerador de Cenários para o Treinamento de Operadores no Ambiente SAGE/SIMULOP para a **Companhia Hidro Elétrica do São Francisco (CHESF)**; além de tarefas de melhoria da infraestrutura de rede e de segurança de dados do LIHM.

1.1 OBJETIVOS DO ESTÁGIO

- Desenvolver Instalador para o Gerador de Cenários;
- Desenvolvimento do Sistema de Monitoramento de Treinamentos.
- Reestruturação da rede e intranet do LIHM;
- Migração e melhorias no sistema de controle de versão SVN;
- Suporte técnico ao LIHM.

1.2 O LIHM

O Laboratório de Interfaces Homem-Máquina (LIHM) – foi criado em 1986 no âmbito do DEE da UFCG com o propósito de apoiar a pesquisa acadêmica na área de interfaces com o usuário de produtos e sistemas. Em 2010, o LIHM lançou o selo de qualidade em Usabilidade, atribuído a produtos que tenham passado por um processo de avaliação e certificação de Usabilidade, visando aprimorar a sua interface com o usuário para torná-los mais competitivos no mercado. Atualmente, o LIHM trabalha com pesquisa e desenvolvimento de e presta serviços de avaliação de produtos e sistemas para o mercado nacional e internacional.

O laboratório, situado no bloco CI do Departamento de Engenharia Elétrica da Universidade Federal de Campina Grande, é liderado pela professora Dra. Maria de Fátima Queiroz Vieira, e conta com uma equipe composta por três pesquisadores doutores, um mestrando, um estagiário e quatro bolsistas de graduação.

Normalmente são realizadas duas reuniões semanais com toda a equipe para atualizar, traçar metas, apresentar resultados, formar equipes e solucionar eventuais problemas. Ferramentas de gestão de projeto são utilizadas para organizar e administrar das atividades do laboratório.

Atualmente o LIHM tem como principal atividade um projeto de Pesquisa e Desenvolvimento com a Companhia Hidroelétrica do São Francisco (CHESF) que consiste em várias melhorias ao sistema de treinamento de operadores de subestações.

O LIHM conta com três pesquisadores de pós-graduação cinco de graduação. Possui um laboratório de usabilidade onde são realizados testes de usabilidade de Software de clientes, monitorando o comportamento do usuário durante o uso das ferramentas com propósito de identificar erros e possíveis melhorias na interface.

2 ATIVIDADES DESENVOLVIDAS

Durante o período de estágio foram desempenhadas diversas atividades, tais como: desenvolvimento do Instalador do Gerador de Cenários; reestruturação da rede e intranet do LIHM; migração e melhorias no sistema de controle de versão SVN, suporte técnico ao LIHM e desenvolvimento de Sistema de Monitoramento de Treinamentos.

2.1 DESENVOLVIMENTO DE INSTALADOR PARA O GERADOR DE CENÁRIOS

Um dos principais projetos do laboratório é o Gerador de Cenários para o Treinamento de Operadores no Ambiente SAGE/SIMULOP, projeto P&D contratado pela empresa CHESF - Companhia Hidroelétrica do São Francisco. Consiste no desenvolvimento de um Gerador de Cenários para o SIMULOP, ambiente de simulação de eventos do sistema supervisório utilizado pela CHESF, o SAGE - Sistema Aberto para Gerenciamento de Energia, desenvolvido pela CEPEL - Centro de Pesquisas de Energia Elétrica. Com interface desenvolvida em JAVA e cenários armazenados em um banco de dados MySQL.

Uma das dificuldades enfrentadas nos testes de versões preliminares do Gerador de Cenários era a instalação de uma versão funcional da ferramenta na máquina em que os testes seriam realizados. Os pacotes JRE (*Java Runtime Enviroment*) eram instalados manualmente, o responsável pela instalação acessava a aplicação diretamente do código fonte e executava a partir da IDE. O banco de dados com os cenários de teste também precisavam ser importados para um destino local ou a conexão precisava ser estabelecida durante a configuração. Com o objetivo de automatizar este processo e reduzir as chances erros, foi desenvolvido um instalador para o Gerador de Cenários.

Para realizar a tarefa foi necessário: Entender a arquitetura do Gerador de Cenários e a estrutura de classes do código fonte e a maneira que este se comunicava com banco de dados; Eleger Software mais adequado; Modificar código para possibilitar a criação de um executável do aplicativo Java e pacotes a serem instalados; Criar *template* para o instalador; Desenvolver Interface do Instalador; Elaborar documentação guia de como gerar o instalador para novas versões.

2.1.1 ARQUITETURA DO GERADOR DE CENÁRIOS

O Gerador de Cenários desenvolvido pelo LIHM para a CHESF é uma aplicação que objetiva aperfeiçoar o processo de criação de cenários para treinamento de operadores no sistema SAGE.

A partir de uma interface gráfica que fornece acesso rápido a representação dos elementos e modificação do estado destes o usuário pode gerar um cenário simulado utilizado nos treinamentos. A aplicação também conta com uma Base de Casos na qual, cenários de treinamento são armazenados. Esta base de casos é armazenada em um banco de dados no servidor da empresa com o intuito de melhorar o acesso, compartilhamento e a gestão de um acervo de cenários que reproduzem tanto situações onde o sistema elétrico opera em condições normais, quanto em situação de contingência. Os cenários serão elaborados por tutores de operadores de subestações e centros de operação do sistema, da empresa CHESF. Há também um armazenamento local dos cenários na ferramenta Gerador de Cenários que permite interromper o trabalho a qualquer momento e salvar o cenário em um arquivo local estruturado na forma de uma ontologia no formato *.owl*.

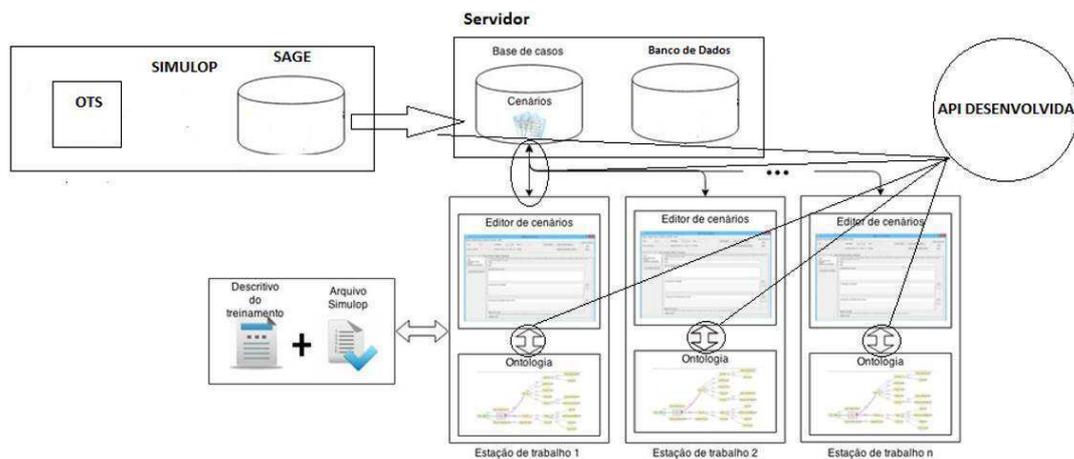


Figura 1 - Representação da Arquitetura do Gerador de Cenários(RANGEL, Gustavo, 2015)

Existem dois modos de utilização do gerador de cenários: online e off-line. No modo online, é necessária uma conexão autenticada com o servidor da empresa para acessar a base de casos, através da qual a aplicação realiza um mapeamento das informações contidas no banco de dados e gera uma instância destes dados no arquivo *.owl* presente na máquina local. Desta forma, durante a criação ou edição de um cenário, a busca de dados será realizada a partir do arquivo owl local. Sem a necessidade de comunicação frequentemente com o servidor. Após a finalização do cenário, pode-se salvá-lo na base de casos do servidor para que outros tutores tenham acesso a ele, tanto para consulta, quanto para a criação de outros cenários a partir daquele. No modo off-line, não é necessário passar por nenhum tipo de controle de acesso. O arquivo *.owl* base contém todas as informações necessárias para a criação de um novo cenário. Este arquivo é atualizado todas as vezes que a ferramenta é conectada no modo online. Caso este arquivo não esteja atualizado, será necessário fazer o login e

esperar que o mapeamento do banco de dados seja completado. A aplicação permite que o arquivo seja salvo localmente, ou na base de casos conforme o diagrama UML representado na Figura 22

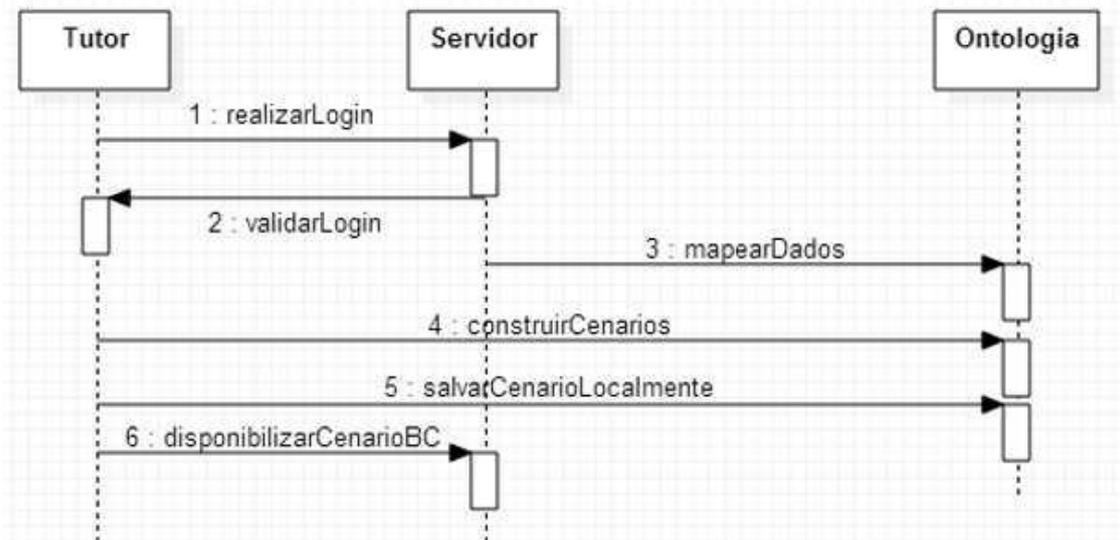


Figura 2- Diagrama UML de sequência para salvar e disponibilizar um cenário na base de casos

2.1.2 ESCOLHA DE SOFTWARE INSTALADOR

Após estudo e teste de vários softwares que geram instaladores, optou-se pelo Advanced Installer, por possuir várias funcionalidades específicas mais adequadas à arquitetura da aplicação em relação às demais ferramentas, dentre as quais se destacaram:

- i. Possibilidade de criação de *templates* do instalador;
- ii. Integração com aplicações Java;
- iii. Facilidade em adicionar SDK's e JRE's em uma única instalação;
- iv. Configuração automática de conexões a Bancos de dados remotos mais modernos.

2.1.3 CONSTRUÇÃO DE EXECUTÁVEL

O principal arquivo a ser instalado é uma versão executável da aplicação Java. Para possibilitar a construção deste arquivo foi necessário realizar diversas modificações no código fonte e tratamento de bugs, além de reconfigurar as conexões com o banco de dados. Utilizando uma ferramenta da IDE NetBeans foi possível construir uma versão executável da aplicação Java com todos os recursos funcionais.

2.1.4 CRIAÇÃO DE TEMPLATE DO INSTALADOR

O uso de um *template* visa padronizar e agilizar o procedimento de criação do instalador. Salvando todas as customizações, apontadores de caminhos dos arquivos em um projeto de instalador que é facilmente atualizado para adaptar-se a novas versões do código. Foi elaborado um tutorial de como gerar um instalador a partir do projeto e com uma versão atualizada do código contido no APÊNDICE A – deste documento.

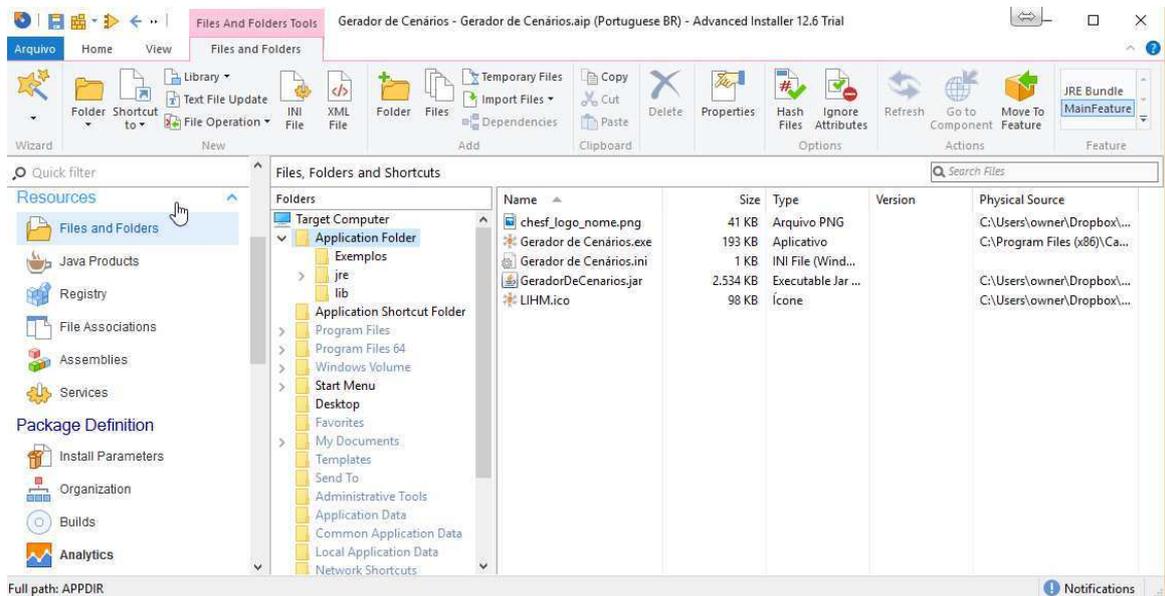


Figura 3 – Projeto do instalador no Advanced Installer.

2.1.5 INTERFACE DO INSTALADOR

A construção do projeto de instalador do template tem como saída dois arquivos de setup que podem ser entregues ao cliente de maneira menos complicada em um disco ou driver auto executável, bastando executá-los para instalar todos os pacotes necessários, arquivos de exemplo e o Gerador de Cenários em um diretório apropriado. A versão final da interface de instalação é apresentada nas Figura 4 e 5.



Figura 4 - Interface do instalador intuitiva e personalizável com o design coerente com os sistemas operacionais atuais

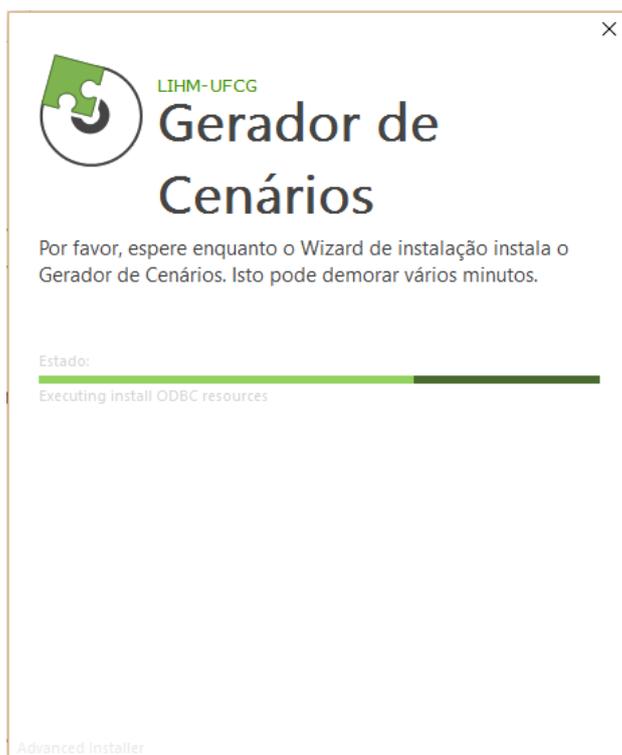


Figura 5 - Tela de Progresso da instalação

desses sistemas. Se trata de um sistema genérico que poder gerenciar qualquer coleção de arquivos. De código fonte ao acervo de uma biblioteca virtual.

Em nosso laboratório utilizamos um sistema SVN para dar apoio ao desenvolvimento de software. Entretanto, deixávamos de explorar algumas das funcionalidades mais úteis da ferramenta, e por problemas de configuração na rede, muitas vezes o repositório de dados era inacessível. Para solucionar este problema, uma das atividades do estágio consistiu na migração do repositório e algumas melhorias no sistema.

2.3.1 MIGRAÇÃO E MELHORIAS NO SISTEMA DE VERSIONAMENTO DE SOFTWARE

Todas as versões do software desenvolvido se encontravam em apenas um computador e eram alteradas diretamente na fonte por apenas um usuário genérico, limitando o uso das funcionalidades do SVN. Por exemplo, a detecção de inserção de *bugs* no código e o *blaming* que aponta o desenvolvedor responsável pelas modificações em cada linha de código e pedaços do arquivo. Conforme apresentado na Figura 8.

Revision	Actions	Author	Date	Message
546		JoseLucas	sexta-feira, 19 de junho de 2015 16:27:00	Fixing hibernate configuration file (hibernate.cfg.xml).
545		JoseLucas	sexta-feira, 19 de junho de 2015 14:48:40	Arquitetura 1.0.
544		Igor	sexta-feira, 19 de junho de 2015 12:07:39	Atualização do Instalador com modificações no código fonte de Login.java para corrigir problemas com a referência ao diretório do shiro.ini O arquivo shiro.ini foi transferido para a
543		Renan Wiana	quarta-feira, 17 de junho de 2015 20:11:45	Verifiquei que quando concertei o salvamento local após ter carregado um cenário local, acabei danificando o salvamento de um cenário novo(recém criado).
542		Andre	quarta-feira, 17 de junho de 2015 16:11:44	adding nova IHM de busca de sinalizações, correção de bug de confirmar mais de 1 equipamento;
541		JoseLucas	terça-feira, 16 de junho de 2015 11:54:46	Resolução de conflitos na migração do NetBeans. Criação de nova camada de dados.
540		Renan Wiana	segunda-feira, 15 de junho de 2015 18:10:57	Juntamente com Andre, nós: Corrigido a necessidade de um Tutor logado para salvar localmente; E o carregamento de um cenário local; Um dos problemas ao salvar era o carrega
539		Renan Wiana	quinta-feira, 11 de junho de 2015 17:57:40	O botão da criar cenário da tela inicial não estava sendo habilitado após fechar o editor de cenários.
538		Alex	quinta-feira, 11 de junho de 2015 01:56:08	Documentação e Refatoração

Figura 8 –Registro de modificações nas versões SVN

Para possibilitar o acesso ao código a partir de outras máquinas (via intranet e internet) e fornecer mais segurança e resiliência ao código, foi feita a migração do banco de dados SVN para o computador Servidor(ATLANTIDA) que permanece ligado, criado um servidor SVN com controle de acesso, além da elaboração de uma documentação tutorial de como utilizar de maneira correta o sistema de versionamento SVN.

Para hospedar o novo repositório de dados, foi utilizado o software VisualSVN server, no qual foram criados usuários pessoais com diferentes níveis de acesso aos dados e criado um acesso seguro via internet que registra quem e quando acessou o repositório.

O acesso via internet passou a ser feito por meio de uma URL hospedada no domínio do laboratório (*lihm.ufcg.edu.br/svn*).

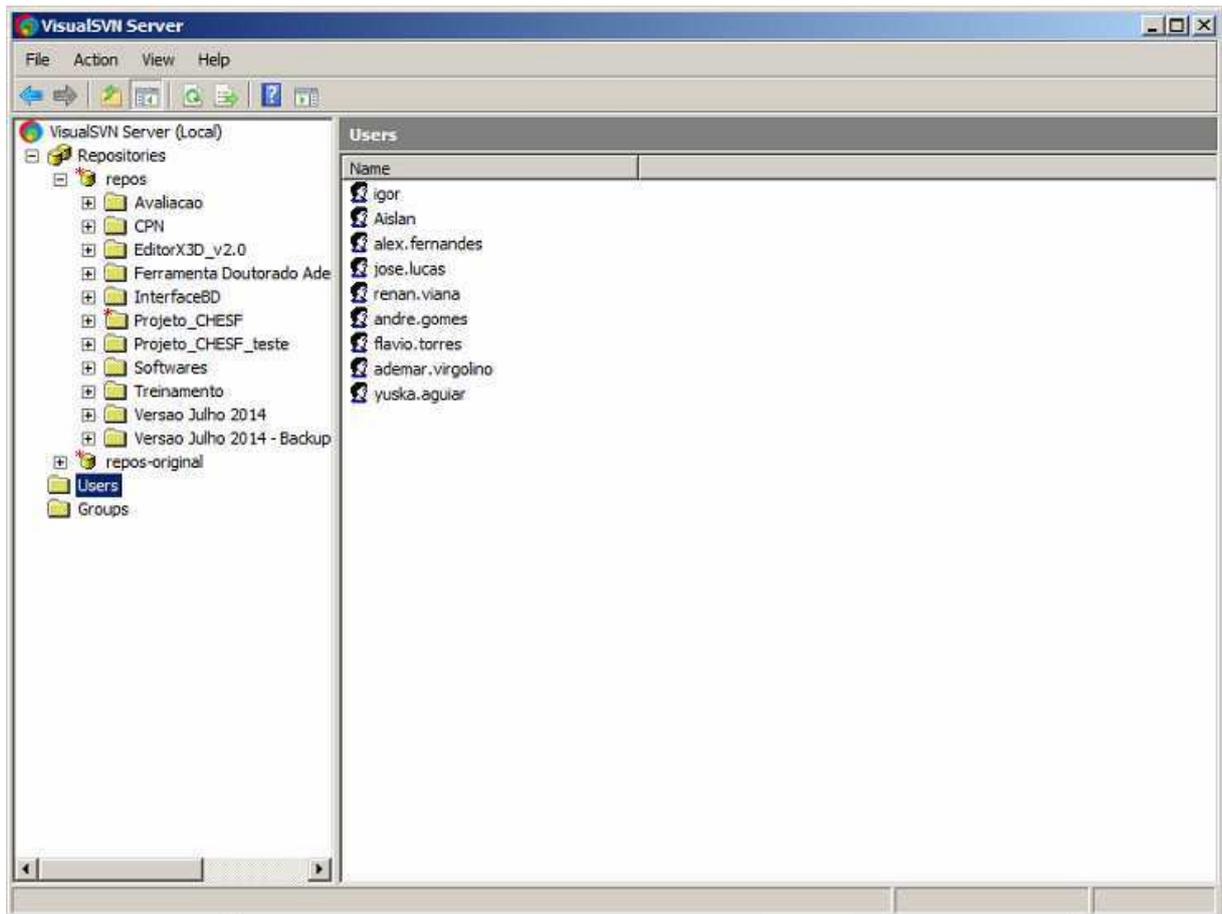


Figura 9 -Repositórios do novo servidor SVN a esquerda e usuários de acesso à direita.

2.4 SUPORTE TÉCNICO AO LIHM

Uma das atividades do estágio consistia em fornecer suporte técnico ao Laboratório na integração dos equipamentos já existentes à nova infraestrutura de rede e na configuração e instalação de novos equipamentos. Também foi fornecido suporte aos bolsistas com questões de desenvolvimento JAVA, utilização do SVN, manipulação de bancos dados MySQL, integração da rede, boas práticas de programação e elaborados documentos tutoriais.

O LIHM estava passando por um período de ampliações, adquirindo novos equipamentos que precisaram ser configurados de acordo com os requerimentos da rede e que possuísem ferramental de desenvolvimento necessário para que trabalhar de forma integrada nos sistemas já existentes.

Devido a recentes furtos de equipamentos ocorridos no laboratório vizinho e a necessidade de monitorar a produtividade dos bolsistas, foi proposto e desenvolvido um sistema de vigilância com acesso remoto.

Utilizando câmeras IP, a rede sem fio, e o software *opensource* iSpy foi elaborado e configurado o sistema de maneira que todas as imagens e áudio ficam armazenados no servidor por

um determinado período de tempo. Por meio do sistema as câmeras também podem ser utilizadas para realizar vídeo conferências e monitorar reuniões remotamente.

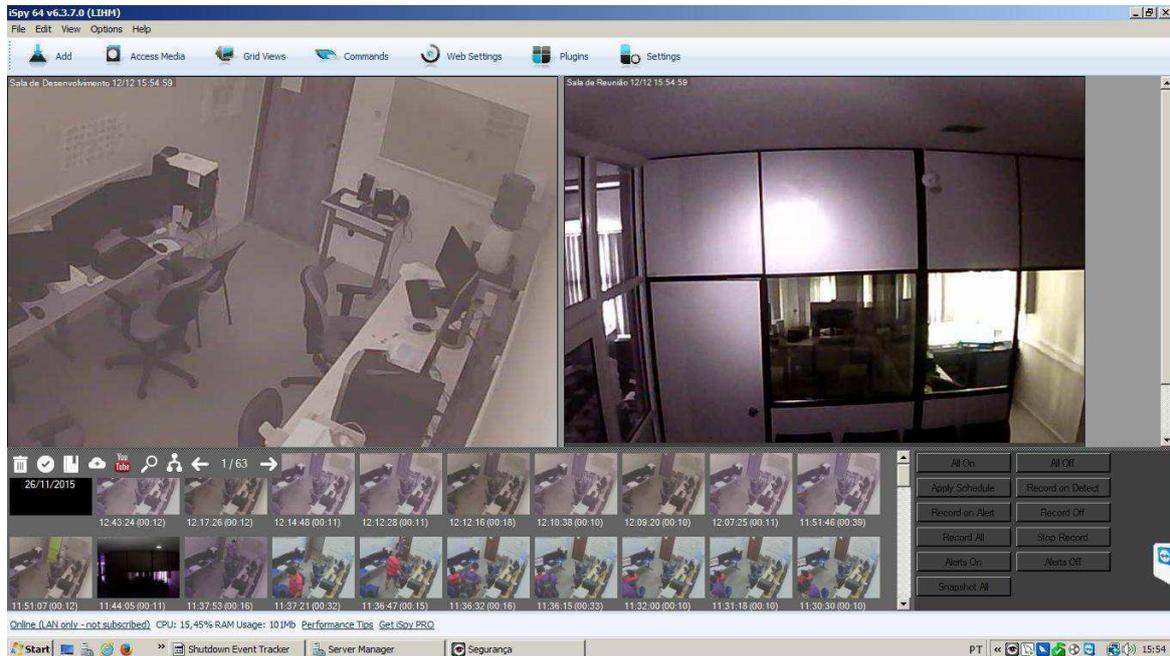


Figura 10- Sistema de Vigilância do LIHM

2.5 SISTEMA DE MONITORAMENTO DE TREINAMENTOS

Uma outra parte do projeto com a CHESF consistia em desenvolver um sistema de monitoramento em tempo real dos treinamentos. Nos treinamentos atuais, um operador em treinamento prático é acompanhado por um operador experiente, denominado o “Sombra”, o qual observa as atividades do treinando e registra em vídeo suas ações para posterior discussão.

Visando otimizar os treinamentos e dar mais visão do Tutor sobre o que acontece no campo a partir da sala de comando foi proposto um sistema que provê uma visão do ponto de vista do operador no campo através de uma câmera.

Foi feita uma visita técnica à subestação Campina Grande II para fazer um reconhecimento do ambiente onde os treinamentos ocorrem e dimensionar um sistema adequado às condições e disposição de obstáculos do campo. Durante a visita foi demonstrada uma rotina de treinamento e foram levantadas possíveis restrições. A área a ser monitorada durante os treinamentos possui vários obstáculos estreitos e a sala de comando se encontra em um prédio ao leste do pátio, conforme demarcado na Figura 11.



Figura 11 –Área a ser monitorada(em verde) e Sala de Comando(em vermelho)

Após debater diversas soluções, optamos por uma câmera GoPro fixada no capacete do operador pois esta gravará em alta resolução em sua memória interna enquanto transmite um sinal de vídeo esses dados em tempo real à sala de comando através de um link de vídeo portátil semelhante aos sistemas de visão de primeira pessoa, muito utilizado no aeromodelismo.

2.5.1 ARQUITETURA PROPOSTA PARA O SISTEMA

O sistema deve ser portátil, robusto e confiável. Composto por duas partes principais, o subsistema de aquisição e transmissão de imagem (cTx) e o subsistema de recepção e exibição da imagem (dRx). O sistema sugerido FPV para aquisição deve ser um de 1,2 ou 1,3GHz com transmissor com potência de ao menos 1200 mW, possuir qualidade sinal superior nas condições, um bom alcance e por trabalhar numa faixa frequência menor que os demais sistemas avaliados, o que significa que seu sinal tem um poder de penetração maior, eliminando a necessidade de uma antena externa ao prédio da sala de comando.

O cTx é formado pela câmera GoPro montada em seu suporte de cabeça sobre o capacete conforme demonstrado na Figura 12, uma bateria de Polímero de Lítio conectada para alimentar o transmissor e a câmera e o transmissor Tx. A conexão da saída de vídeo da câmera ao transmissor é feita de um cabo mini USB que segue o esquema apresentado na Figura 13. A bateria e o transmissor podem ser fixados na cinta externa ou diretamente no capacete com uma fita adesiva de velcro.



Figura 12 – Montagem de Câmera do cTx

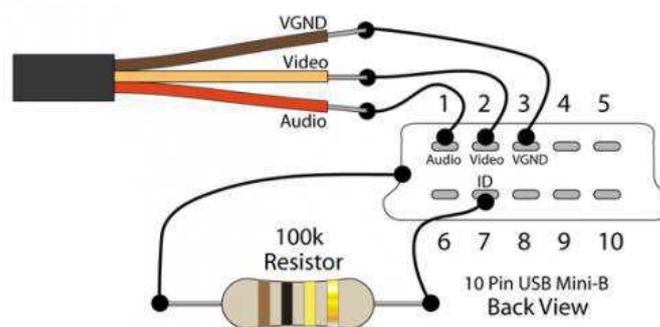


Figura 13 – Pinagem do cabo micro usb da GoPro para saída de vídeo analógica

Para visualizar o vídeo ao vivo, o sinal da saída de vídeo do Rx é conectado ao computador do tutor através de uma placa de captura de vídeo USB. Pode também ser conectado diretamente à entrada analógica de uma televisão disponível na sala de comando com um cabo RCA, ou a ambos utilizando um splitter.

3 CONSIDERAÇÕES FINAIS

As atividades desenvolvidas durante o período de estágio foram realizadas com êxito, onde foi possível aplicar alguns dos conhecimentos adquiridos ao longo do curso. Os dois meses de trabalho desenvolvidos junto Laboratório de Interface Homem-Máquina permitiram consolidar conhecimentos abordados durante o curso de Engenharia Elétrica; sendo as disciplinas mais presentes Técnicas de Programação, Informática Industrial, Sistemas de Processamento de Áudio e Vídeo e Redes de Computadores. Desenvolver atividades de estágio supervisionado no âmbito acadêmico foi relevante dos pontos de vista técnico e profissional. O acompanhamento de pesquisadores experientes expandiu o conhecimento sobre a atuação acadêmica. Além de o projeto permitir uma interação com o cliente para entender a flexibilidade da especificação de um sistema ao longo de seu desenvolvimento.

Por fim, pode-se afirmar com que o estágio atendeu as suas expectativas ao possibilitar novos aprendizados e a consolidação dos conhecimentos anteriormente obtidos ao longo dos anos de universidade.

BIBLIOGRAFIA

COLLINS-SUSSMAN, Ben; PILATO, C. Michael; FITZPATRICK, Brian W.. **Version Control with Subversion: For Subversion 1.7 (Compiled from r5064)**. Stanford: [s.n.], 2011. SVN book Disponível em: <http://svnbook.red-bean.com/en/1.7/svn-book.pdf>. Acesso em: 20/05/2015.

LEON-GARCIA, Alberto; WIDJADJA, Indra. **Communication Networks: Fundamental Concepts and Key Architecture**. 2. ed. Toronto: McGraw-Hill, 2001.

RANGEL, Gustavo. **Descrição e armazenamento de cenários de treinamento de operadores a partir de uma ontologia de domínio**. Campina Grande, 2015

SIERRA, Kathy; BATES, Bert. **Head First Java**. 2. ed. New York: O'Reilly Media, 2005.

ABNT. **NBR 14724 - Informação e documentação — Trabalhos acadêmicos — Apresentação**. Associação Brasileira de Normas Técnicas. [S.l.]: ABNT. 30 dez. 2005. p. 9.

ABNT. **NBR 6034 - Informação e documentação - Índice - Apresentação**. Associação Brasileira de Normas Técnicas. [S.l.]: ABNT. 2005. p. 4.

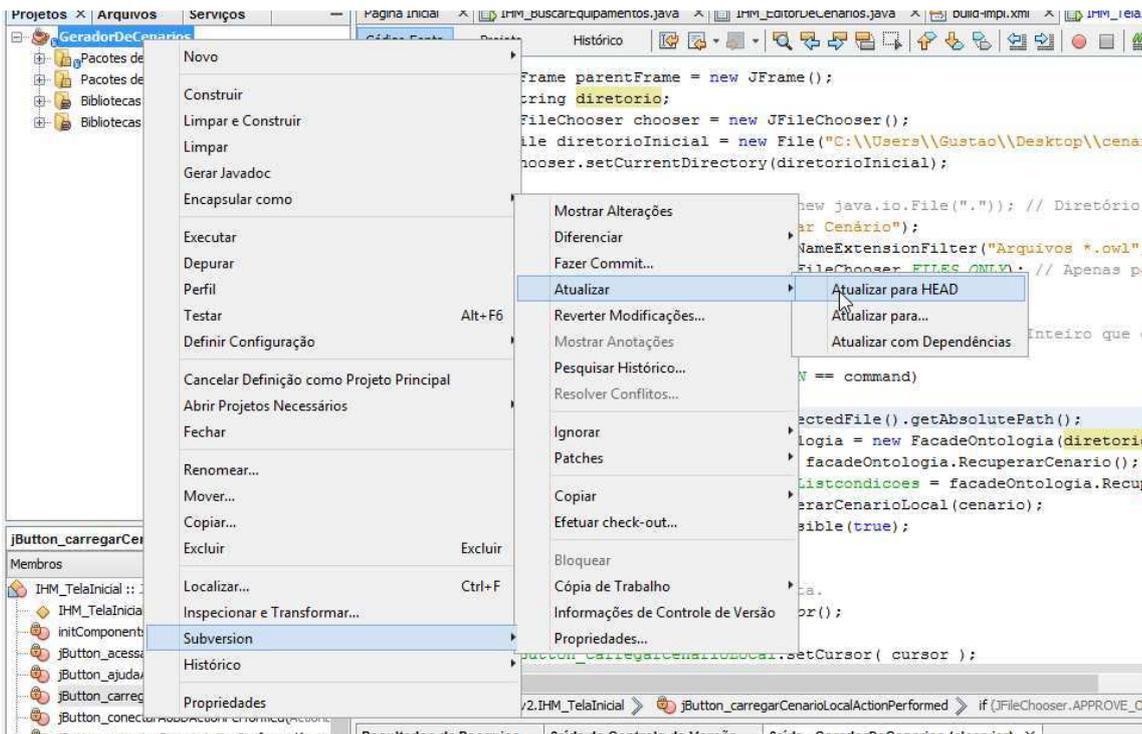
APÊNDICE A – TUTORIAL PARA CONSTRUÇÃO DE INSTALADOR DO GERADOR DE CENÁRIOS

Introdução

Este é um tutorial com a descrição passo-a-passo do procedimentos necessários para construir o instalador para novas versões da aplicação Gerador de Cenários a partir da ferramenta Advanced Installer, baseando-se no *template* do projeto de instalador já existente no diretório do projeto na pasta Instalador.

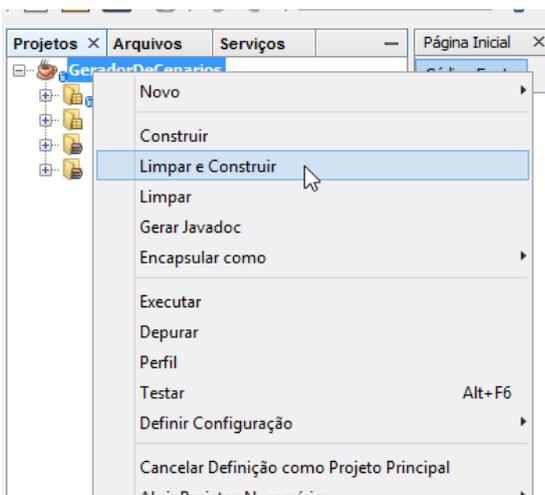
Passo 1 – Atualizar projeto para versão desejada

Atualizar código do projeto a partir do servidor SVN para a versão a ser instalada pelo novo instalador.



Passo 2 - Construção do .jar no NetBeans

No NetBeans, limpar e construir o projeto. Para criar o arquivo *GeradorDeCenários.jar* em "*PASTADOPROJETO*"\dist.



Atenção! Certificar-se que as bibliotecas também foram copiadas para "*PASTADOPROJETO*"\dist\lib.

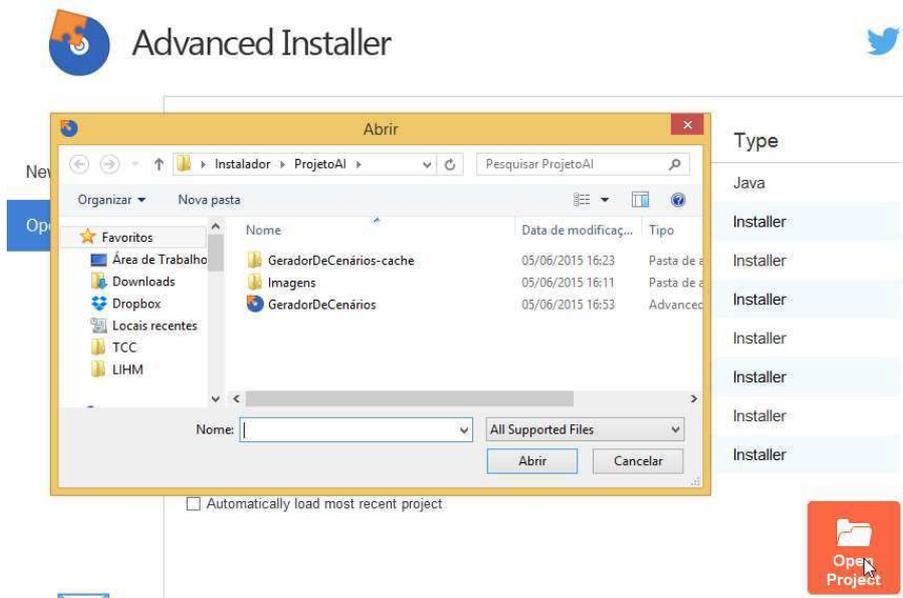
```

Resultados da Pesquisa | Saída do Controle de Versão | Saída - GeradorDeCenarios (clean.jar) x
Note: Recompile with -Xlint:unchecked for details.
6 warnings
Copying 48 files to C:\Users\Igor\Dropbox\LIHM\ProjetoCHESF\build\classes
Copied 1 empty directory to 1 empty directory under C:\Users\Igor\Dropbox\LIHM\ProjetoCHESF\build\classes
compile:
Created dir: C:\Users\Igor\Dropbox\LIHM\ProjetoCHESF\dist
Copying 1 file to C:\Users\Igor\Dropbox\LIHM\ProjetoCHESF\build
Copy libraries to C:\Users\Igor\Dropbox\LIHM\ProjetoCHESF\dist\lib.
Building jar: C:\Users\Igor\Dropbox\LIHM\ProjetoCHESF\dist\GeradorDeCenarios.jar
To run this application from the command line without Ant, try:
java -jar "C:\Users\Igor\Dropbox\LIHM\ProjetoCHESF\dist\GeradorDeCenarios.jar"
jar:
CONSTRUÍDO COM SUCESSO (tempo total: 4 segundos)

```

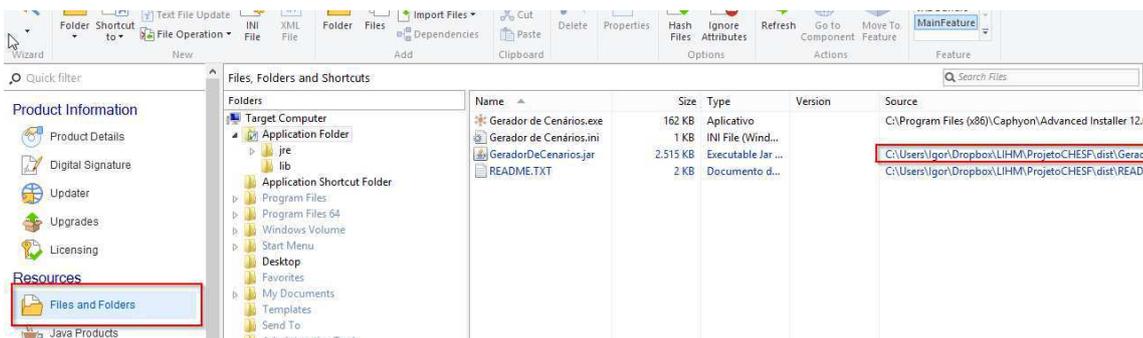
Passo 3 – Abrir projeto do Instalador

Abrir o programa Advanced Installer instalado na máquina SUMERIO, abrir o projeto salvo em “PASTADOPROJETO”\Instalador\ProjetoAI (arquivo *.aip)



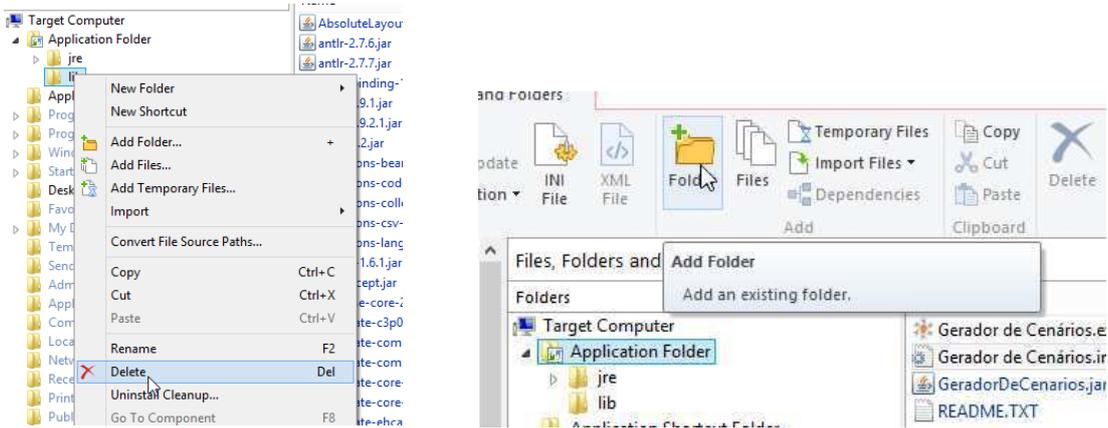
Passo 4 – Verificar fontes do arquivo .jar

Na aba Files and Folders, verificar se o caminho do arquivo *GeradorDeCenarios.jar* é o mesmo do .jar construído no passo 2. Caso contrário, é necessário editar para que aponte ao novo arquivo. Para isto, selecione o arquivo, clique no botão properties e edite o caminho.



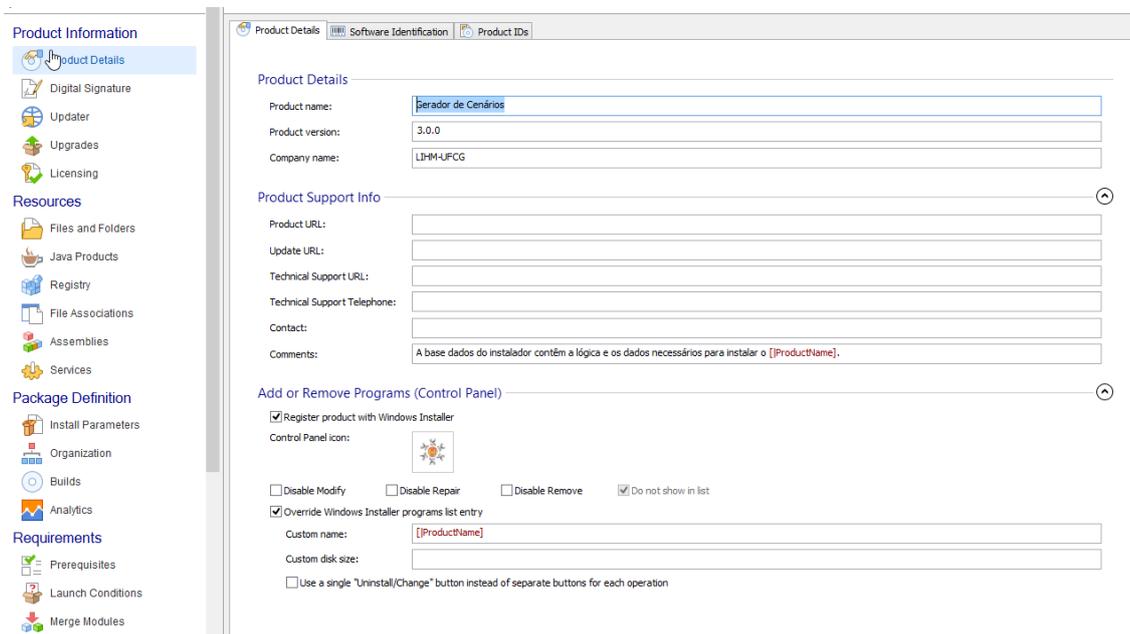
4 Passo 5– Verificar fontes da pasta \lib

Faça o mesmo para verificar se o destino da pasta “PASTADOPROJETO”\dist\lib está correto. Se não, remova a pasta e adicione-a novamente apontando para o caminho correto.



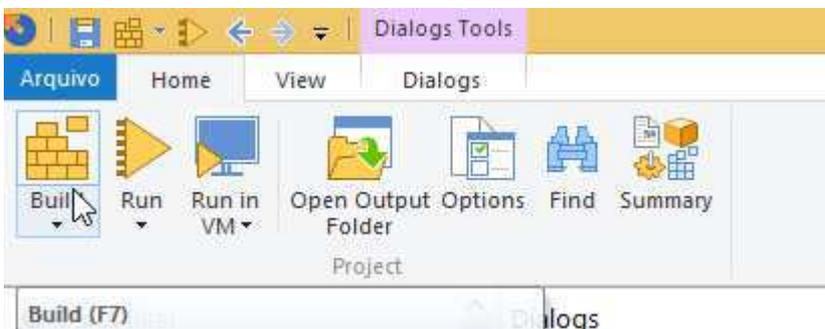
5 Passo 6 – Atualizar dados de versão

Atualizar os detalhes da versão do produto na aba *Product Details*.



6 Passo 7

Após atualizar os caminhos para o novo *.jar* construído, e sua respectiva biblioteca, é necessário construir um novo instalador clicando em build.



Os arquivos do instalador serão criados em “PASTADOPROJETO”/Instalador/Setup Files

Atenção! Para que a instalação seja feita corretamente, deve-se abrir o executável e os dois arquivos (*.msi* e *.exe*) devem estar na mesma pasta.

Funcionalidades a serem adicionadas.

Adicionar os Termos de Uso

Conexão com o banco de dados remoto da CHESF

Caso necessário na versão final, isto pode ser feito seguindo as instruções no menu ODBC do Advanced Installer ou diretamente no código

Associação dos arquivos do tipo .owl com a aplicação.

Para isto, é necessário saber qual a função chamada quando carregamos cenário local. E implementar o comando que a chame quando abrir o arquivo .owl.

Para esclarecer qualquer dúvida, enviar e-mail para: igormmaciel@gmail.com

APÊNDICE B– ESTRUTURA DA REDE DO LIHM

Dados de rede:

- Máscara: 255.255.255.0
- Gateway: 150.165.61.254
- DNS1: 150.165.61.1
- DNS2: 150.165.1.1

Faixa de IP: 150.165.61.60-200

- 60-69 : equipamentos de infra estrutura(servidor, roteadores, etc.);
- 70-79 : Impressoras e dispositivos de saída;
- 80-109: Computadores;
- 110-119: Câmeras;
- 120-200 - Faixa do DHCP do wi-fi.

Wireless:

- SSID: LIHM
- Senha:

1.1 DADOS DOS DIPOSITIVOS

Computador	Usuário	IP	Obs.
ROTEADOR OIW	Sala de Pesquisa	Omitido	DHCP .120-.200 Usuário: Senha:
ROTEADOR LINKSYS	Prof. Fátima	Omitido	Usuário: Senha:
ATLANTIDA	Servidor Vermelho	Omitido	
IMPRESSORA HP8610	LIHM	Omitido	.70
IMPRESSORA SCX-3400	Prof. Fátima	Omitido	.71
ASTECA	André Gomes	Omitido	
INCA	DISPONÍVEL	Omitido	
VIKING	Sala de Controle 2	Omitido	
SUMERIO	Igor Maciel	Omitido	
BIZANTINO	DISPONÍVEL	Omitido	
MAIA	DISPONÍVEL	Omitido	
GODO	MORTO	Omitido	
ROMANO	Alex Fernandes	Omitido	
ASSIRIO	José Lucas	Omitido	
BABILONIO	Sala de Testes 1	Omitido	
FENÍCIO	MORTO	Omitido	
CHOSON	MORTO	Omitido	
TEUTAO	Sala de Controle 1	Omitido	
HUNO	Ademar Neto	Omitido	
SARRACENO	Sala de Testes 2	Omitido	
PERSA	Prof. Fátima	Omitido	
SIMULOP-2	Hélio	Omitido	
SIMULOP-1	Renan	Omitido	
--	LIVRE	Omitido	
IPCAM 01	LIHM-Sala de Desenvolvimento	Omitido	Usuário: Senha:
IPCAM 02	LIHM-Sala de reuniões	Omitido	Usuário: Senha:
CAMERA 01	LIHM- Sala de testes	Omitido	Usuário: Senha:

ANEXO – TUTORIAL DE COMO UTILIZAR O SVN

TUTORIAL DE COMO UTILIZAR O SVN TORTOISE

Neste tutorial é descrito os passos básicos para utilizar o TortoiseSVN, dentre eles baixar o projeto pela primeira vez, fazer atualização do projeto e por fim submeter alterações.

1. BAIXAR OS ARQUIVOS PELA PRIMEIRA VEZ

Para o primeiro uso é necessário seguir os seguintes passos:

1. Instalar o TortoiseSVN no computador.
2. Liberar acesso ao servidor para isto você deve:
 - 2.1. Abrir o Windows Explorer (atalho: simbolo do windows +E) e digitar no campo principal: \\150.165.61.62 (Figura 1)



Figura 1 - Liberar acesso ao SVN (windows explorer)

- 2.2. Irá abrir uma tela de autenticação digitar login: "Administrator" e senha: "!ntr@l12" (ambos sem as aspas).
- 2.3. Marcar a opção: lembrar credenciais e confirmar (se você não marcar esta opção, sempre que iniciar o Windows irá precisar liberar novamente o acesso ao servidor).
- 2.4. Pronto acesso liberado, agora é configurar o TortoiseSVN para baixar os arquivos.
3. Criar uma pasta no diretório onde deseja "baixar" os arquivos do projeto.
4. Clicar com o botão direito sobre a pasta e selecionar a opção "SVN Checkout" como ilustrado na Figura 1.

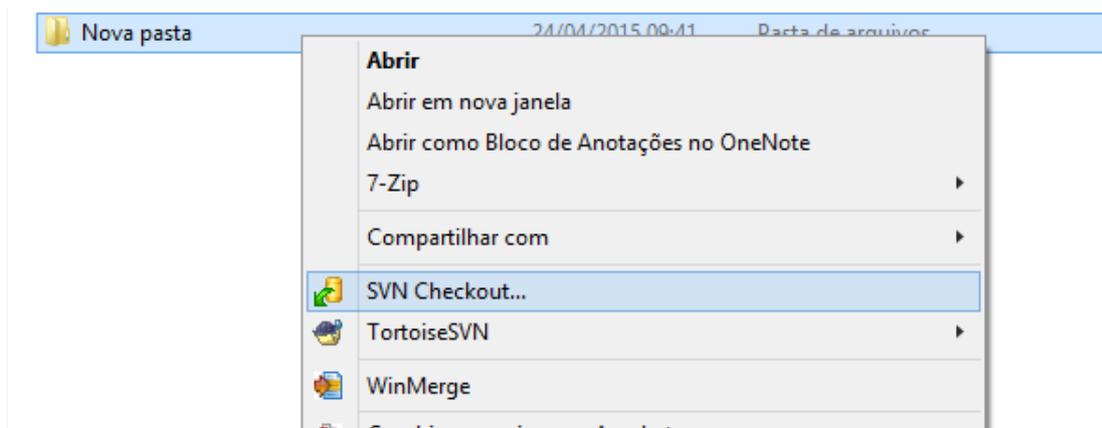
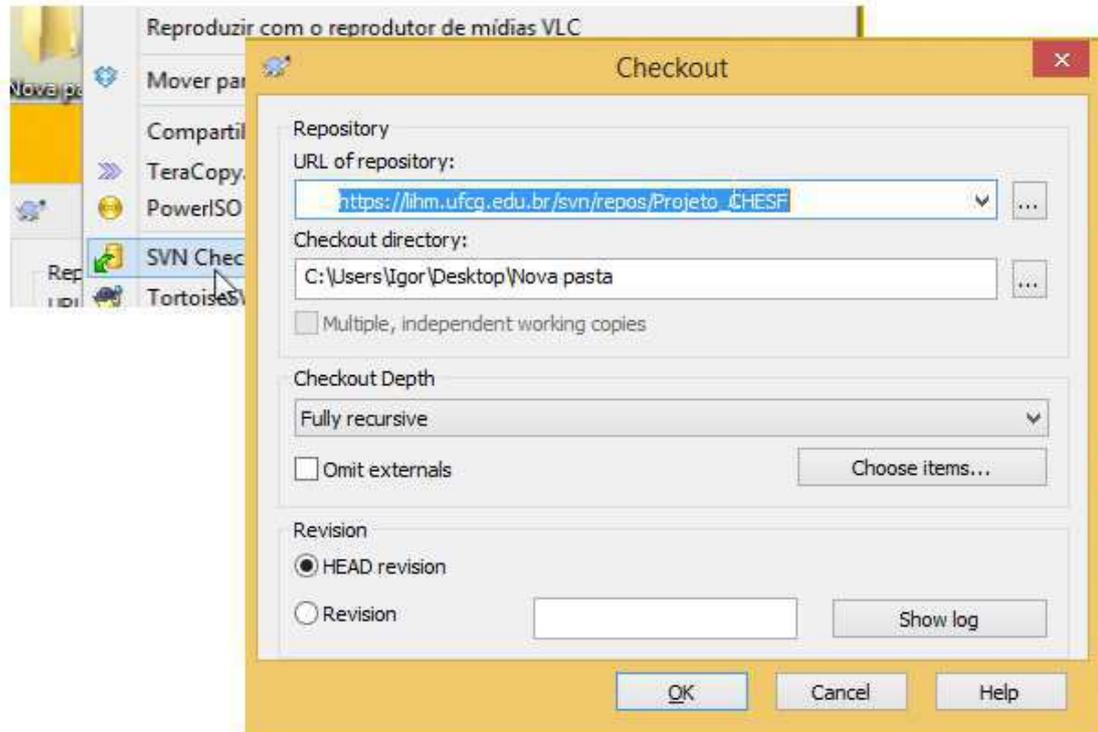


Figura 2 - Passo inicial: SVN checkout

5. Na próxima tela digitar no campo de "URL of repository" o seguinte endereço: https://lihm.ufcg.edu.br/svn/repos/Projeto_CHESF/GeradorDeCenarios (Figura 3). Para quem

quer fazer atualização de tudo que esta no servidor relacionado ao projeto, como por exemplo, versões anteriores, deve fazer o Checkout de um nível acima no caso do link será: https://lihm.ufcg.edu.br/svn/repos/Projeto_CHEsf.



6. Figura 3 - Inserir caminho da pasta do projeto no servidor

7. Confirmar pressionando OK, será necessário que você faça login com seu usuário e senha fornecidos pelo administrador do repositório.
8. Agora é esperar pois irá demorar em torno de 15 minutos para ele "baixar" todos os arquivos para o seu pc.

Após concluir o *checkout* você pode abrir o projeto no Netbeans, para isto basta abrir o Netbeans, clicar na opção abrir projeto, selecionar a pasta do projeto e confirmar. Para um bom funcionamento do sistema você deve sempre atualizar (*update*) sua versão com a do servidor e sempre que tiver uma versão estável deve submeter (commit) ao servidor.

ATUALIZAR PASTA DO PROJETO (FAZER *UPDATE*)

Sempre antes de iniciar um novo código você deve atualizar o seu para a versão mais recente do servidor, para isto basta clicar com o botão direito sobre a pasta do projeto no Windows Explorer e clicar na opção "*SVN Update*", como ilustrado na Figura 4.

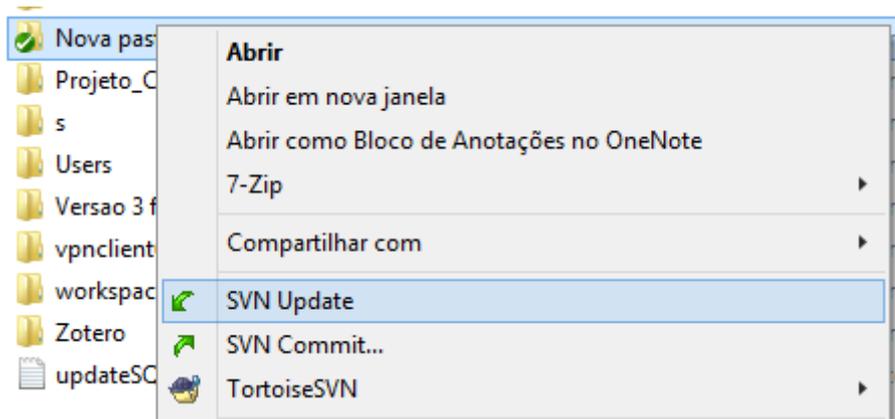


Figura 4 - Tela de seleção do SVN Update

SUBMETTER VERSÃO ESTÁVEL E TESTADA DO CÓDIGO (FAZER *COMMIT*)

Para submeter uma nova versão do projeto o procedimento é o mesmo da atualização, clicar com o botão direito sobre a pasta do projeto no Windows Explorer e clicar na opção "*SVN Commit*", como ilustrado na Figura 5. A única diferença é que ao fazer uma submissão (*commit*) você deve inserir mensagem ("*Message*") significativa, explicando o que você alterou, acrescentou ou corrigiu, para que os demais tenham o controle e entendam o que foi alterado de uma versão para outra (vide campo preenchido como exemplo na Figura 6).

Outro detalhe importante que deve ser observado é que ao criar novos arquivos (.java, .xml, bibliotecas) você deve selecionar na tabela "*changes made*" (destacado em vermelho na Figura 6) para que esses novos arquivos sejam adicionados ao projeto, caso contrário ele irá apenas atualizar os alterados e os criados ficaram apenas no computador de quem criou e conseqüentemente surgirá um erro no computador dos demais ao atualizarem.

Observação: quando for adicionar uma lib você deve selecionar a opção colocar caminho relativo e adicionar a lib na pasta de projeto\lib.

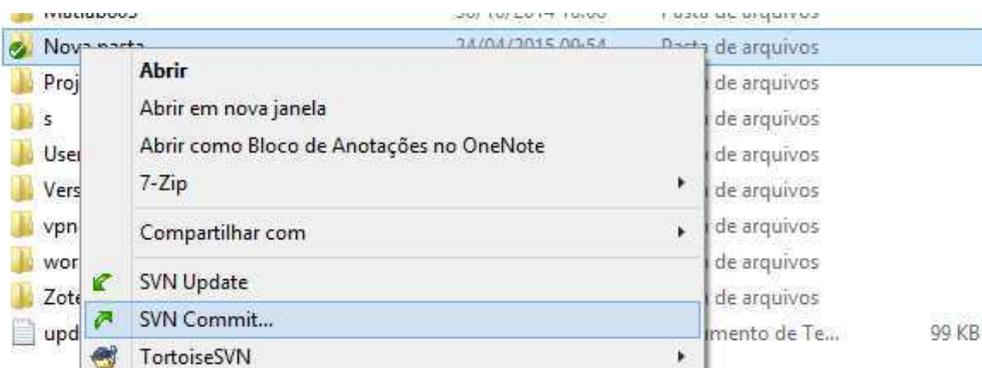


Figura 5 - Tela de seleção do SVN Commit

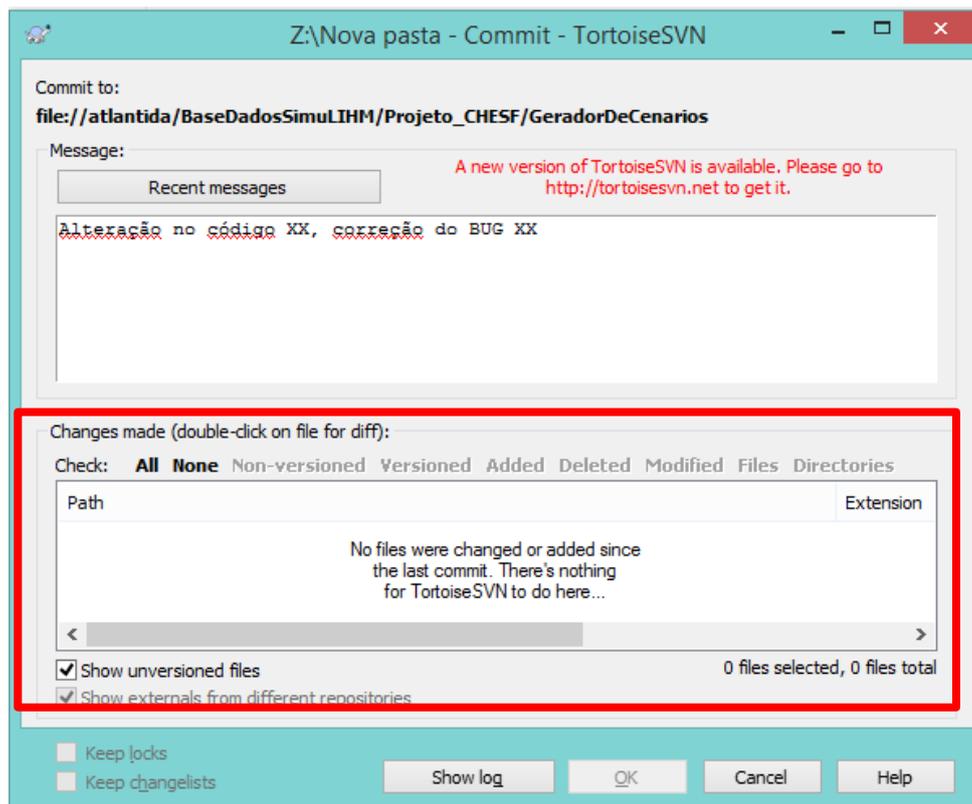


Figura 6 - Tela de submissão de arquivos no servidor (Commit)