

Iniciação Científica

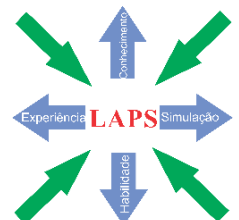
"Relatório de Implementação"

Autor: Nome

Orientador: Rodolfo Rocha Vieira Leocádio



Laboratório de Práticas Simuladas
LAPS – EMED
Abril de 2020



Sumário

1 - Especificação de requisitos	3
1.1 - Interação cliente x empresa	3
1.2 - Proposta ao cliente	3
2 - Serviços oferecidos pelo produto	3
3 - Requisitos de Interface	3
3.1 - Interfaces de Usuário	3
3.2 - Interfaces de Hardware	4
3.3 - Interfaces de Software	4
4 - Requisitos Funcionais	4
4.1 - Requisito 1 –	4
5 - Requisitos Não Funcionais	4
5.1 - Usabilidade	4
5.2 - Confiabilidade	5
5.3 - Desempenho	5
5.4 - Manutenibilidade	6
5.5 - Portabilidade	6
5.6 - Requisitos Legais	7
5.7 - Requisitos de Segurança	7
5.8 - Outros Requisitos Não Funcionais	7
6 - Diagramas de Caso de Uso	7
6.1 - UC000: Nome do primeiro caso de uso	8
7 - Testes	9
7.1 - Configuração dos Testes	9
7.2 - Atividades e Eventos	10
7.3 - Incidentes	10
7.4 - Variações	10
7.5 - Abrangência	10
7.6 - Sumário dos Resultados	10
7.7 - Avaliação	10
8 - Validação	10
8.1 - Funcionalidade	10
8.2 - Plano de validação	11
8.3 - Resultados da validação	11
8.4 - Conclusões da validação	11
9 - Controle de versões	11
10 - Referências	11
1 - Especificação de requisitos	

1.1 - Interação cliente x empresa

Relatar o que o cliente precisa.

1.2 - Proposta ao cliente

Nome do produto e de seus componentes principais.

Missão, visão e valores do produto.

Limites do produto.

Benefícios do produto.

2 - Serviços oferecidos pelo produto

Inclua um diagrama de contexto do sistema. Trata-se de um diagrama com casos de uso que mostre as interfaces do sistema em seu ambiente de operação.

Identifique as principais funções que o produto desempenhará.

Identifique as principais relações entre os atores que interagem com o sistema.

Descreva de forma concisa e objetiva cada ator. O objetivo é descrever os diferentes papéis que o usuário pode desempenhar quando interage com o sistema.

3 - Requisitos de Interface

3.1 - Interfaces de Usuário

Descreva os detalhes de layout das interfaces com o usuário do sistema. Para cada interface com o usuário (tela, janela ou caixa de diálogo) é necessário descrever os campos nelas presentes e os comandos de acesso, por exemplo, clique com mouse, botões na barra de ferramentas e/ou teclas de atalhos.

Exemplo de roteiro:

Janela Principal.

- Layout: O software deverá executar a partir de uma janela do Windows apresentando na área central o mapa topográfico do terreno em questão. Haverá uma barra de ferramentas contendo as funções onde o usuário poderá invocar.
- Relacionamento com outras interfaces: Da interface principal será possível visualizar a interface de cadastro de equipamentos, a interface do painel de controle, a interface do inspetor de objetos e a interface de atualização de mapas topográficos.
- Campos: Serão apresentados os seguintes campos:

Barra de ferramentas:

- Atualização de terreno – contém lista de mapas topográficos;
- Barra de status – ativa/desativa a exibição da barra de status;
- Painel de Controle – exhibe/oculta a janela com a listagem de equipamentos;
- Inspetor de Objetos – exhibe/oculta a janela do inspetor de objetos;

Área de desenho: região da tela onde é exibido o mapa topográfico do terreno.

Barra de Status: apresenta informações sobre a execução do programa e coordenadas de pontos indicados no terreno.

- Comandos: A Tabela 1 apresenta os comandos relacionados com a interface principal do software.

Tabela 1 – Lista de Comandos: Janela Principal

Nome	Ação	Atalho
Atualização de terreno	Lista os mapas topográficos disponíveis para que seja escolhido aquele que ser exibido na área de desenho.	Barra de ferramentas
Exibição da barra de status	Exibe ou esconde ou a barra de status.	Barra de ferramentas
Painel de controle	Exibe ou esconde o painel de controle.	Barra de ferramentas e teclas [ctrl] + [P]
Inspetor de objetos	Apresenta o estado das variáveis monitoradas sobre o objeto selecionado.	Barra de ferramentas

3.2 - Interfaces de Hardware

Descreva a comunicação do sistema com algum hardware, se houver. Para cada interface de hardware é necessário descrever os atores que a utilizam, a fonte de dados de entrada, o destino dos dados de saída e o formato dos dados que nela trafegam, além dos relacionamentos com as demais interfaces do sistema.

3.3 - Interfaces de Software

Descreva a comunicação do sistema com algum outro software. Para cada interface de software é necessário descrever os atores que a utilizam, a fonte de dados entrada o destino dos dados de saída e o formato dos dados que nela trafegam, além dos relacionamentos com as demais interfaces do sistema.

4 - Requisitos Funcionais

Descreva as ações fundamentais através das quais o produto aceita e processa as entradas especificadas, gerando as respectivas saídas. É feito o detalhamento desses requisitos, em nível suficiente para o projeto do produto, de seus testes de aceitação e de seu manual de usuário.

4.1 - Requisito 1 –

Descreva os requisitos, prioridade relativa, nível de dificuldade e critério de aceitação.

5 - Requisitos Não Funcionais

Descreva os requisitos não funcionais do sistema. Os requisitos não funcionais, em geral, estabelecem o nível de qualidade que os requisitos funcionais devem atender, ou condicionam e ambientam tais requisitos. Eles definem, por exemplo, critérios de usabilidade, portabilidade e desempenho que devem ser satisfeitos pelos requisitos funcionais.

5.1 - Usabilidade

São os requisitos que definem as facilidades de uso do sistema, o nível de consistência dos dados apresentados e de documentação.

Exemplo de roteiro:

Conformidade com padrões: O sistema deve seguir o padrão XYZ – Qualidade de Serviços de Informação na empresa ABC e, quando necessário, o RUP – Rational Unified Process. Estes padrões podem ser adaptados.

Nível de habilidade do usuário: O sistema deve atender desde ... até os funcionários da empresa ABC, assim, o nível dos usuários que utilizarão o sistema deve ser considerado variado. Por este motivo, deve-se optar por interfaces com recursos gráficos e formulários de fácil compreensão, para que as tarefas possam ser realizadas no menor tempo e com o menor custo de treinamento.

Presença de ferramentas de auxílio: O sistema deve prover ao usuário helps on-line, menus, tool tips, enfim, documentação com o intuito de auxiliar em sua utilização.

Treinamento: Haverá necessidade de treinamento para usuários com variados níveis de habilidade, desde ... até analistas, supervisores e gerentes de risco da empresa ABC. Assim, haverá necessidade de treinamento para os variados níveis, que serão identificados na próxima fase.

Dicas para o Usuário: O sistema deve fornecer orientações a respeito do significado de cada campo apresentado na tela, visando auxiliar seu preenchimento ou utilização.

5.2 - Confiabilidade

São os requisitos que definem a confiabilidade do sistema. Englobam aspectos como previsibilidade, acurácia de resultados, tolerância a falhas, recuperabilidade, dentre outros.

Exemplo de roteiro:

Integridade dos dados: O sistema deve manter a integridade dos dados residentes nas estações servidoras e também nas aplicações clientes para cadastros que são realizados off-line e depois transmitidos em lotes para as estações servidoras. Assim, é necessário manter, com integridade, as bases de dados centrais (estações servidoras) e locais (aplicações cliente).

Controle de redundância: O sistema deve possuir controle de redundância dos dados, uma vez que muitas informações estarão duplicadas nas estações servidoras e nas aplicações clientes. Visa-se desta forma, manter as bases de dados das estações servidoras e as bases de dados das aplicações clientes atualizadas e sincronizadas.

Disponibilidade: O sistema deve estar disponível 24 horas por dia, 7 dias por semana.

Medidas de tempo entre falhas: Pode ocorrer uma falha do sistema a cada ano. O tempo total de falha esperado no ano é de 4 horas.

Medidas de tempo de reparo: Após uma falha, o sistema pode permanecer indisponível, no máximo, por um período de meia hora.

Máximo de defeitos ou taxa de defeitos: A ser identificado na próxima fase.

Rotinas operacionais: O backup dos dados sensíveis deve ser realizado diariamente. Será adotada a abordagem de backup incremental ou diferencial nos dias úteis e um backup total nos finais de semana. Os demais dados serão copiados de forma integral nos finais de semana. Dados sensíveis são os dados considerados imprescindíveis para operação do sistema, serão identificados na fase posterior, a fase de modelagem de dados.

5.3 - Desempenho

Relate os requisitos que melhor definem o desempenho do sistema. Engloba considerações sobre tempo de resposta, taxa de serviço, velocidade de processamento, eficiência, precisão, consumo de recursos computacionais e volume de produção.

Exemplo de roteiro:

Tempo de resposta para uma transação: O sistema implementa serviços utilizando a tecnologia Web. Serviços implementados na Web devem ter um tempo de resposta mínimo, não

podendo ultrapassar os 10 segundos que já são praticados por corporações de grande porte que mantêm serviços na Web.

Número de usuários do sistema distribuído ao longo do tempo: Atualmente a Empresa & possui cerca de 2 mil funcionários na área T e cerca de 30 mil parceiros cadastrados. Planeja-se um crescimento do número de parceiros em torno de 5% ao ano. Já o número de funcionários cresce a uma taxa menor, podendo variar entre 1% e 3% de crescimento anual.

Estimativas de transações com o banco de dados: Estima-se para o sistema que se tenha aproximadamente 800 mil operações de cadastro (consulta, alteração, inserção e exclusão) por ano. O crescimento destas operações não deve ultrapassar a 5% ao ano.

Quantidade de acessos simultâneos: O sistema deve suportar até 1 mil usuários simultâneos e uma taxa de 1% de crescimento anual.

Quantidade de transações por segundo: Supondo 21,5 milhões de operações com o banco anualmente. Extrapolando outras 21,5 milhões de operações que não acessam ao banco, tem-se cerca de $43/6,912 = 6,23$ transações por segundo.

Capacidade: O sistema deve acomodar não simultaneamente os 32 mil usuários e simultaneamente os 7 mil usuários definidos anteriormente. Além disto, o sistema deve ter a capacidade de alocar recursos extras, proporcionais à taxa anual de crescimento de usuários.

Área de armazenamento: Supondo que uma operação de cadastro, seja esta de Produto ou Serviço, ocupe 1024 bytes (1KB). Já uma operação de @ ou \$ ocupa, em média, 10K. A área de armazenamento mínima anual deve ser de 11 GBytes para cadastros e outros 110Gbytes para @'s e \$'s. A área de espaço ocupada não leva em conta a fonte do sistema e outros dados necessários ao sistema como banco de regras, banco de técnicas de IC, banco de configurações de técnicas de IC, banco com os metadados do sistema, o Data Warehouse do sistema, entre outros. Este espaço será definido na segunda fase do projeto, após a modelagem de dados.

5.4 - Manutenibilidade

São os requisitos que definem a capacidade do sistema de suportar mudanças, evoluções e reparos. Definem a testabilidade, extensibilidade, adaptabilidade, compatibilidade, entre outros.

Exemplo de roteiro:

Padrões de programação: Será adotado e adaptado o Java Programming Guideline, proposto por Scott Ambler.

Padrões Gerais: Os layouts do sistema, por exemplo, Web ou DESKTOP, deve seguir o mesmo padrão adotado para o sistema RTW da empresa XYZ, feito em Java. A arquitetura do sistema será baseada no modelo MVC (Model View Controller).

Ferramentas de avaliação de impacto: Deverá ser adotado mecanismos propostos pela disciplina de Gerência de Configuração e Mudanças do RUP.

Características de extensibilidade de linguagem adotada: Os componentes a serem desenvolvidos para o sistema devem possuir extensibilidade, ou seja, devem facilitar a adição de novas características que se fizerem necessárias.

Facilidades de instalação: O sistema deve possuir mecanismos de facilitação de instalação. Ressalta-se, ainda, que esta característica é fundamental, pois os usuários são pessoas com pouco conhecimento de informática. Assim, por exemplo, os módulos desktop dos cadastros devem possuir uma facilidade de instalação similar à do sistema RTW da empresa XYZ.

Elaboração e Distribuição de novas versões: Deverá existir um gerenciamento para controlar a elaboração de novas versões do subsistema. Também deverá haver um controle para a distribuição das últimas versões para as partes interessadas. Poderá ser adotado mecanismos propostos pela disciplina de Gerência de Configuração e Mudanças do RUP.

Interoperabilidade: O sistema deve considerar dados de entrada e saída que são mantidos em sistemas externos, ou seja, sistemas que estão fora do domínio do sub-sistema. Assim, deverá haver

interoperabilidade entre o sistema e sistemas legados. As regras para este inter-relacionamento deverão ser estabelecidas na próxima fase.

Mídia de Armazenamento: A mídia de armazenamento para os dados persistentes e temporários do subsistema serão discos rígidos (HD). Já para os dados de backup, a mídia será definida na próxima fase.

5.5 - Portabilidade

Descreva as restrições relacionadas a infraestrutura tecnológica necessária para o desenvolvimento, implantação e a utilização do sistema.

Exemplo de roteiro:

Linguagem de programação – Java, HTML e JavaScript;

Banco de dados – Oracle 10g;

Servidor de Aplicação – JBoss;

Sistema Operacional – RedHat 5.0 e

Configuração de Hardware, Software Básico, Rede de Comunicação de Dados e Instalações Físicas para o desenvolvimento, as simulações e os testes do sistema na UFOP em conjunto com os representantes da empresa XYZ – A ser definido posteriormente.

Outros aspectos da infraestrutura serão levantados na segunda fase do projeto.

5.6 - Requisitos Legais

Descreva os requisitos não funcionais derivados da legislação que regula a construção do sistema, que restringem ou controlam de alguma maneira o seu desenvolvimento. Por ser a atividade fiscal plenamente vinculada, se faz necessário observar os dispositivos da legislação pertinentes a cada caso. No tocante ao projeto, os requisitos legais a serem atendidos em primeiro lugar são os dispositivos normativos editados pela LEI RRR que dispõem sobre segurança de dados, entre elas segurança física e lógica dos mesmos, requisitos quanto a cadastramento de usuários e sistemas, além do uso de extratores e acesso ao ambiente externo.

5.7 - Requisitos de Segurança

Descreva os requisitos que definem a política de segurança adotada para o sistema.

Exemplo de roteiro:

Mecanismos ou sistemas de controle de acesso – A princípio foi definido que o sistema vai utilizar os serviços do sistema NNN para controle acesso e autenticação de usuários.

Transações e perfis de acesso – O sistema vai se valer do perfil, transação e âmbito do usuário para permitir o acesso a serviços internos. O perfil pode ser um leque de opções razoáveis, dependendo da política a ser adotada. As transações são as descritas e implementadas no sistema. Os âmbitos podem ser: individual, local, regional e nacional.

Sigilo – Sabe-se que o sistema e os dados manipulados por este são de sigilo absoluto, mas os critérios e procedimentos que garantam tal sigilo serão especificados em detalhes na próxima fase do projeto.

5.8 - Outros Requisitos Não Funcionais

Descreva aqui os requisitos identificados para o sistema que não estão mapeados dentro da classificação adotada no corpo desse documento.

Exemplo: cálculos estatísticos, condições de Entrega, orçamentária, financeira, econômicos, distribuição, monitoramento, replicação de Bases de Dados, dentre outros.

6 - Diagramas de Caso de Uso

Detalhe os relacionamentos existentes entre os casos de uso e os atores do sistema. Cada caso de uso identificado no diagrama de contexto pode dar origem a um ou mais requisitos funcionais. Todos os requisitos funcionais devem ter o fluxo básico e os fluxos alternativos de seus casos de uso especificados. No entanto, enquanto os fluxos descrevem os detalhes de cada caso de uso, os diagramas de casos de uso descrevem os relacionamentos dos casos de uso entre si e com os atores que interagem com o sistema. Interações entre os atores não devem ser documentadas nos diagramas.

São duas as possíveis relações entre os casos de uso: inclusão ou extensão. A relação de extensão indica que o comportamento do caso de uso estendido pode ser ou não inserido no caso de uso extensor. A notação é uma seta pontilhada partindo da extensão para o caso de uso estendido com a etiqueta <<extend>>. A relação de inclusão implica que o comportamento do caso de uso incluído é obrigatoriamente inserido no comportamento do caso de uso inclutor. A notação é uma seta pontilhada que aponta para o caso de uso incluído com o estereótipo <<include>>.

Recomenda-se o uso de diagramas para detalhar partições do diagrama de contexto, mostrando grupos correlatos de casos de uso primários e os atores; diagramas que mostrem todos os casos de uso de um determinado ator; diagramas para mostrar um certo caso de uso e seus relacionamentos; ou diagrama de uma versão do sistema ou protótipo.

Exemplo de roteiro:

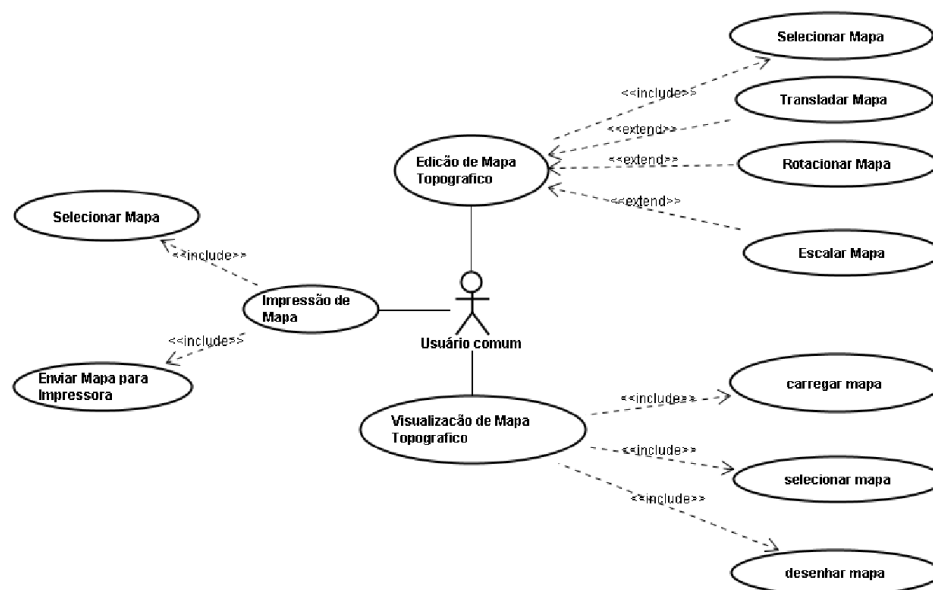


Figura 1 – Diagrama de casos de uso: Manipulação de mapas topográficos por usuários comuns.

Descreva o fluxo básico e os fluxos alternativos dos casos de uso.

6.1 - UC000: Nome do primeiro caso de uso

- Fluxo Básico de Eventos: O caso de uso descreve interações entre um usuário e o sistema. Se forem trocadas informações, seja específico sobre o que é transmitido de um lado para outro. Por exemplo, não é muito esclarecedor dizer que o usuário digita informações do cliente se elas não forem definidas. É melhor dizer que o usuário digita o nome e o endereço do cliente.

Fluxos alternativos simples podem ser apresentados no texto do fluxo de básico. Se forem usadas apenas algumas sentenças para descrever o que acontece quando há uma alternativa, faça isso diretamente dentro do fluxo. Se o fluxo alternativo for mais complexo, utilize uma seção separada para descrevê-lo.

O fluxo complexo de eventos deve ser melhor estruturado em sub-fluxos. Ao fazer isso, a meta principal deve ser aprimorar a clareza do texto. Os sub-fluxos podem ser chamados muitas vezes de muitos lugares. Lembre-se de que o caso de uso pode executar sub-fluxos em sequências opcionais, em loops ou mesmo vários ao mesmo tempo.

- **Fluxos Alternativos:** Comportamentos alternativos do sistema são descritos em uma subseção do fluxo básico. Os fluxos alternativos geralmente descrevem como as exceções que ocorrem no fluxo principal serão tratadas pelo sistema. Exemplo de roteiro:

< A1 Primeiro Fluxo Alternativo >

Descreva o fluxo alternativo, exatamente como qualquer outro fluxo de eventos.

< A2 Segundo Fluxo Alternativo >

Pode haver, e muito provavelmente haverá, vários fluxos alternativos em cada área de funcionalidade. Mantenha cada fluxo alternativo separado para aprimorar a clareza do documento.

- **Pré-Condições:** Insira as condições prévias do caso de uso. Uma condição prévia de um caso de uso descreve o estado em que sistema deve estar antes de um caso de uso ser executado. Pode haver várias condições prévias. Mantenha cada pré-condição separada para tornar o documento claro.

- **Pós-Condições:** Insira as condições posteriores do caso de uso. Uma condição posterior de um caso de uso descreve a lista de estados possíveis que o sistema pode estar imediatamente após um caso de uso ter sido concluído. Pode haver várias condições posteriores. Mantenha cada pós-condição separada para tornar o documento claro.

- **Pontos de Inclusão:** Insira os pontos de inclusão do caso de uso. Definição local do ponto de inclusão do caso de uso.

- **Pontos de Extensão:** Insira os pontos de extensão do caso de uso. Definição local do ponto de extensão do caso de uso.

- **Requisitos Especiais:** Um requisito especial é, geralmente, um requisito não funcional que é específico de um caso de uso, mas não é fácil ou naturalmente especificado no texto do fluxo de eventos do caso de uso. Exemplos de requisitos especiais incluem requisitos legais e reguladores, padrões de aplicativos e atributos de qualidade do sistema a ser construído incluindo requisitos de utilidade, confiabilidade, desempenho ou suportabilidade. Adicionalmente, outros requisitos como sistemas e ambientes operacionais, requisitos de compatibilidade e restrições de projetos devem ser capturados nesta seção.

7 - Testes

7.1 - Configuração dos Testes

Exemplo de roteiro:

Tabela 2 – Parâmetros configurados para testes.

N ^o	Hardware	Software	Ferramentas	Componentes
1	Core 2 Duo 6400, 2,13 GHz - 3,25GB RAM ATI Radeon 4560, 512 MB	Windows XP SP2	Teste manual	-
2	Core 2 Duo E7300, 2,66 GHz - 3,5GB RAM NVidia GeForce 9600 GSO, 512 MB	Windows XP SP2	Teste manual	-

Tabela 3 – Participantes dos testes.

Função	Responsabilidades
Programador Júnior	Instalação e configuração do sistema para os testes Assessoria ao testador na especificação dos testes Assessoria à elaboração do relatório dos testes
Programador Sênior	Instalação e configuração do sistema para os testes Assessoria ao testador na especificação dos testes Assessoria à elaboração do relatório dos testes
Analista de Sistema	Planejamento dos testes Especificação dos testes Realização dos testes Assessoria à elaboração do relatório dos testes de integração

7.2 - Atividades e Eventos

Exemplo de roteiro:

Tabela 4 – Resumo das atividades e eventos.

Casos de uso	Ação do usuário	Resultado Esperado	Conforme		Incidentes	Período
			S	N		
Transladar mapa via teclado	Clicar tecla W	Mover o mapa para cima e atualizar as coordenadas na Barra de Status	x			10/03/2009
	Clicar tecla S	Mover o mapa para baixo e atualizar as coordenadas na barra de status	x			10/03/2009
Edição de mapa topográfico	Selecionar Mapa	Apresentar o mapa selecionado na janela principal	x			10/03/2009
	Mover o cursos	Os parâmetros da transformação selecionada são desenhados na janela principal	x			10/03/2009
Barra de Ferramentas	Clicar o botão esquerdo do mouse no botão "Abrir"	Abre Caixa de Diálogo "Abrir Arquivo"	x			10/03/2009
	Clicar o botão esquerdo do mouse no botão "Salvar"	Abre Caixa de Diálogo "Salvar Arquivo"	x			10/03/2009

7.3 - Incidentes

7.4 - Variações

7.5 - Abrangência

A cobertura pode ser considerada satisfatória dentro do previsto para estes testes do módulo de imersão virtual.

7.6 - Sumário dos Resultados

O produto passou nos testes, com correção de alguns bugs de implementação.

7.7 - Avaliação

Levando-se em conta os resultados alcançados, a bateria de testes pode ser considerada de cobertura satisfatória.

8 - Validação

8.1 - Funcionalidade

Descreva a funcionalidade que deseja validar.

8.2 - Plano de validação

Descreva o processo de validação

8.3 - Resultados da validação

Exponha os resultados da validação.

8.4 - Conclusões da validação

Conclua sobre a validação discutindo os resultados.

9 - Controle de versões

10 - Referências

De acordo com a ABNT.

Este relatório foi elaborado seguindo os materiais didáticos do prof. Tiago Garcia de Senna Carneiro.

<https://www.terralab.com.br/doku.php?id=terralab:curso:tecprog:index>