

PLANO GERAL DE GESTÃO DE CONFIGURAÇÃO

Versão 2.6

Responsável: José David Fumo

Gestor de Configuração

josedavidfumo@gmail.com

Erick Paulo Samuel Mahanjane

Gestor de Projectos

erick.mahanjane@gmail.com

Luthermilla Mucula dos Reis Ecole

ÍNDICE

1. Introdução.....	4
1.1. PROPÓSITO	4
1.2. DEFINIÇÕES, SIGLAS E ABREVIACÕES.....	4
2. Gestão de Configuração de <i>Software</i>	5
2.1. ORGANIZAÇÃO, RESPONSABILIDADES E INTERFACES.....	5
2.1.1. Comitê de Controle de Configuração (CCC)	5
2.1.2. Patrocinador do Projecto	6
2.1.3. Gestor de Projectos	6
2.1.4. Gestor de Tecnologia	6
2.1.5. Gestor de Configurações.....	7
2.1.6. Líder de Equipe	7
2.1.7. Gestor de <i>Marketing</i> (ou Analista de Qualidade).....	7
2.1.8. Desenvolvedores	8
2.2. FERRAMENTAS, AMBIENTES E INFRAESTRUTURA.....	8
2.2.1. Ferramentas Utilizadas.....	8
2.3. ENDEREÇO DO REPOSITÓRIO.....	9
3. Programação de Gestão de Configuração	9
3.1. IDENTIFICAÇÃO DA CONFIGURAÇÃO	9
3.1.1. Itens de Configuração.....	9

3.1.2.	Padrão de nomeação dos arquivos e pastas.....	15
3.1.3.	Estrutura de pastas padrão do Projecto	19
3.1.4.	Permissões de acesso.....	24
3.1.5.	<i>Baseline</i>	25
3.2.	GESTÃO DE CONFIGURAÇÃO DE BASE DE DADOS	27
3.3.	CONTROLE DE CONFIGURAÇÕES E MUDANÇAS	27
3.3.1.	Solicitação de Mudança e Aprovação	27
3.4.	REGISTO E COMUNICAÇÃO DO STATUS DA CONFIGURAÇÃO	30
3.4.1.	Processo de <i>Backup</i>	30
3.4.2.	Expurgo de Projectos Inativos	30
3.4.3.	Auditorias.....	30
4.	Marcos de Projecto.....	31
5.	Recursos e Treinamento	32
5.1.	RECURSOS.....	32
5.2.	TREINAMENTO.....	33
5.2.1.	Conceitos Básicos e Boas Práticas	33
6.	Subcontratação e Controle de Versão do Fornecedor.....	34
7.	Esclarecimentos	34

Tabela de Figuras

Figura 1 - Estrutura Padrão de Pastas Por Cliente**Error! Bookmark not defined.**

Figura 3 – Processo geral de desenvolvimento com GC**Error! Bookmark not defined.**

Figura 2 - Fluxo de Aprovação de Solicitação de Mudança 29

1. Introdução

1.1. Propósito

O propósito do Plano de Gerenciamento de Configuração é descrever como a gestão de configuração (GCO) será realizada durante todo o ciclo de vida do projecto. Isto inclui a documentação de como a GCO é gerida, os papéis e responsabilidades das pessoas envolvidas, como as mudanças são feitas no item de configuração (IC), e como a comunicação de todos os aspectos da GC são realizadas entre os participantes do projecto.

Sem um plano de gestão de configuração documentado é provável que um IC seja perdido ou trabalho desnecessário seja feito por falta de versão e controle de documentos. Um plano de gestão de configuração é importante para todos os projectos, especialmente para projectos de tecnologia da informação (TI).

Nessa ordem de ideias, o presente documento visa estabelecer a gestão de configurações do sistema de compras *online*, *iXitolo*, por forma a garantir que o produto final seja entregue de acordo com uma especificação inicialmente estabelecida e homologada por todas as partes interessadas e envolvidas.

1.2. Definições, Siglas e Abreviações

Nesta secção são descritas as definições, siglas e abreviações utilizadas neste documento e em comunicações diversas nos projectos, de modo a facilitar o entendimento de termos que serão abordados durante ao discorrer do documento.

- **IC:** Item de configuração;
- **GC:** Gestão de Configuração;
- **CCC:** Comitê de Controle de Configuração;
- **GP:** Gestor de Projectos;

- **GI:** Gestor de Infraestrutura;
- **Checkout:** Baixar do repositório inteiramente o Projecto ou uma pasta dele;
- **Commit:** Enviar para o repositório os arquivos locais modificados;
- **Update:** Actualizar (sincronizar) localmente arquivos específicos ou pastas;
- **GT:** Gestor de Tecnologia;
- **GM:** Gestor de *Marketing*;
- **Marketing:** relativo a actividades orientadas a entender e atender as necessidades do cliente;
- **CCM:** Controle de Configuração e Mudança;

2. Gestão de Configuração de Software

2.1. Organização, Responsabilidades e Interfaces

2.1.1. Comitê de Controle de Configuração (CCC)

O comitê de controle de configuração (CCC), do inglês *Configuration Control Board*, é composto por patrocinador, gestor de projecto (GP), gestor de configuração, gestor de tecnologia (GT), gestor do negócio (GN), gestor de *marketing* (GM) e líderes de equipa. Suas responsabilidades são:

- Definir as diretrizes da GC;
- Definir os ICs padrão a serem considerados em todos os projectos relacionado com o sistema;
- Analisar e aprovar / rejeitar solicitações de mudança de configuração;
- Assegurar que todas as mudanças aprovadas estão sendo aplicadas no plano de GC de conhecimento unanime;
- Definir quando são realizadas auditorias de GC nos projectos; e

- Definir as regras de mudanças por respostas a necessidades no mercado.

2.1.2. Patrocinador do Projecto

O patrocinador do projecto é quem financia o projecto. Em projectos externos geralmente ele é o cliente e em internos é o diretor da área que solicitou o Projecto (Infra, Operação, administrativo, etc). Neste caso, o financiador do projecto é cliente ou supervisor, que poderá fornecer recursos para a futura implementação da plataforma como combinado.

- Participar da reunião do CCC;
- Aprovar qualquer assunto que requer escopo, tempo ou custo adicional aos Projectos; e
- Pode fornecer uma opinião adicional para a equipe desenvolvedora sobre o sistema.

2.1.3. Gestor de Projectos

Suas responsabilidades são:

- Responsabilidade geral por todas as actividades de GC do Projecto;
- Comunicar quaisquer mudanças aprovadas no plano de GC;
- Participar das reuniões do CCC;
- Fechar nova linha-base caso haja mudanças no escopo ou prazo;
- Solicitar configuração de acesso dos utilizadores do Projecto no repositório; e
- Garantir a padronização dos recursos utilizados para o desenvolvimento do projecto.

2.1.4. Gestor de Tecnologia

Suas responsabilidades são:

- Dimensionar os recursos necessários ao bom funcionamento da GC;

- Garantir a disponibilidade e desempenho dos recursos necessários;
- Executar a configuração de permissões solicitada pelo gestor de Projecto.

2.1.5. Gestor de Configurações

Suas responsabilidades são:

- Reportar solicitações de mudança do processo de GC ao CCC;
- Identificar novos ICs com apoio dos desenvolvedores;
- Solicitar qualquer necessidade de treinamento em GC;
- Criar as BRANCHES, *tags* e realizar os *merges*; e
- Padronizar versionamento de plataforma de desenvolvimento.

2.1.6. Líder de Equipe

Suas responsabilidades são:

- Reportar ao gestor de configurações qualquer mudança de GC descoberta durante a fase de execução do Projecto;
- Identificar ICs candidatos e comunicar ao gerente de configuração;
- Eliminar qualquer dúvida ou impedimento que tenha surgido durante as reuniões;
- Participar das reuniões do CCC; e
- Facultar poder (Todos os gestores podem ser líderes mediante um período devidamente clarificado pela equipe desenvolvedora, e segundo aspectos previamente acordados).

2.1.7. Gestor de *Marketing* (ou Analista de Qualidade)

- Assegurar que a equipe de desenvolvimento está seguindo os padrões e métodos de GC definidos pelo CCC;

- Orientar os desenvolvedores quanto as boas práticas de GC;
- Fornecer padrões e *templates* de configuração as equipes de Projecto;
- Fornecer qualquer necessidade de treinamento em GC;
- Participar das reuniões do CCC.

2.1.8. Desenvolvedores

Suas responsabilidades são:

- Identificar ICs candidatos e comunicar ao Líder de equipe;
- Reportar ao Líder de equipe qualquer mudança de GC descoberta durante a fase de execução do Projecto;
- Explorar a criatividade colectiva para desenvolvimento de *software*; e
- Seguir todos os padrões e métodos de GC na implementação do Projecto.

2.2. Ferramentas, Ambientes e Infraestrutura

2.2.1. Ferramentas Utilizadas

O controle de versões de *software*, para o presente sistema, será feito através de um servidor GIT com o cliente *web* GitHub (ferramenta que disponibiliza a utilização do GIT).

O GIT (<https://git-scm.com/>) é um sistema de controle de versão distribuído livre e de código aberto projectado para lidar com tudo, desde projectos pequenos a muito grandes com velocidade e eficiência. O GIT é fácil de aprender e tem uma pequena pegada com um desempenho rápido. É utilizada em diversas empresas e projectos como Google, Linux, Facebook, Android, Microsoft, Twitter, linkedin entre outras.

A ferramenta cliente *web* padrão definida para aceder o repositório é o GitHub (<https://github.com/>). É uma plataforma de desenvolvimento inspirada na maneira como o utilizador trabalha. Ele permite que programadores, utilitários ou qualquer utilizador cadastrado na plataforma contribuam em projectos privados e/ou Open Source de qualquer lugar do mundo. GitHub é amplamente utilizado por programadores para divulgação de seus trabalhos ou para que outros programadores contribuam com o projecto, além de promover fácil comunicação através de recursos que relatam problemas ou mesclam repositórios remotos (*issues*, *pull request*).

2.3. Endereço do Repositório

O repositório do GIT estará disponível internamente e também por acesso externo. O servidor se encontra nas dependências da Matriz, sendo somente utilizado através de ferramentas que disponibilizam a utilização do GIT, nesse caso o GitHub. Segue abaixo os endereços:

- <https://github.com/search?utf8=%E2%9C%93&q=Trabalho+do+grupo+1%2C+3o+ano%2C+iXitolo&type=> Endereço de pesquisa resultante do projecto
- <https://github.com/emahanjane/EngenhariaSoftware> : Endereço oficial do repositório.

3. Programação de Gestão de Configuração

3.1. Identificação da Configuração

3.1.1. Itens de Configuração

Os itens de configuração (IC) são os vários tipos de arquivos que devem ser incluídos no repositório para ter seu histórico de mudanças controlado.

Commented [S1]:

iXitolo LTDA ME
Contacto: ixitolo@gmail.com



Tanto os documentos como os arquivos-fonte que compõem um produto de *software* são Itens de Configuração (IC), assim como também são as ferramentas de *software* necessárias para o desenvolvimento.

Nem todos os arquivos precisam ser controlados, por exemplo, um executável gerado pelo código fonte não deve estar no repositório, visto que sua fonte geradora já está sendo controlada no repositório.

Abaixo segue a lista de ICs que devem ser considerados em todos os Projectos.

Item de Configuração (IC)	Descrição	Responsável	Quando é incluído?
Documento de requisitos	Especifica todas as funcionalidades e relações existentes entre as várias entidades do sistema.	Erick S. Mahanjane	11.08.2017
Proposta técnica e financeira	Representa uma sugestão de implementação de <i>software</i> focalizado em termos técnicos e financeiros	Erick S.Mahanjane	29.09.2017
Dicionário de dados	Ferramenta que auxilia a organização dos tabelas e itens de gestão de banco de dados, para permitir o controlo de ACID	Luthermilla Ecole	24.11.2017
Código fonte	Representa o conjunto de linhas que fazem com que o artefacto do produto seja apresentado e explorado	José David	24.11.2017
Cronograma	Representa um instrumento de planeamento e controle semelhante a um diagrama, em que são definidas e detalhadas minuciosamente as actividades a serem executadas durante um período estimado.	Vânia Chenridza, Albertina Mondjane	20.11.2017
Diagrama de classe	Constitui uma representação da estrutura e relações das classes que servem de modelo para objectos	Erick S. Mahanjane	11.08.2017
Plano de gestão de Metodologia Ágil de Desenvolvimento	Planificação que permite acelerar a entrega do produto de um determinado projecto, porém priorizando a qualidade e iteração entre a equipa desenvolvedora	Erick S. Mahanjane, Luthermilla Ecole	20.08.2017
Repositório distribuído	Local de armazenamento de onde pacotes de software podem ser recuperados e instalados em um computador	José David, Ercílio Samo	31.10.2017
Bases de Dados	Conjunto de arquivos relacionados entre si com registos sobre pessoas, lugares ou qualquer outro item de armazenamento de dados. Representam colecções organizadas de dados que se relacionam de forma a criar algum sentido (Informação) e dar mais eficiência durante uma	José David, Luthermilla Ecole	20.10.2017

	pesquisa ou estudo		
Casos de uso	Tipo de classificador representando uma unidade funcional coerente provida pelo sistema, subsistema, ou classe manifestada por sequências de mensagens intercambiáveis entre os sistemas e um ou mais actores	Erick S. Mahanjane, Luthermilla Ecole e Vânia Cherindza	11.08.2017
Diagrama de sequência	Representa a sequência de processos (mais especificamente, de mensagens passadas entre objectos) num programa de computador.	Erick S. Mahanjane	11.08.2017
Diagrama de casos de uso	Descreve a funcionalidade proposta para um novo sistema que será projectado, sendo uma excelente ferramenta para o levantamento dos requisitos funcionais do sistema	Erick S. Mahanjane	11.08.2017
Plano de gestão de configuração	descreve todas as atividades do Gerenciamento de Controle de Configuração e Mudança (CCM) que serão executadas durante o ciclo de vida do produto ou do projeto	José Fumo e Ercílio Samo	09.11.2017
Requisitos não funcionas	Requisitos relacionados ao uso da aplicação em termos de desempenho, usabilidade, confiabilidade, segurança, disponibilidade, manutenção e tecnologias envolvidas	Albertina Mondjane e Vânia Cherindza	11.08.2017
<i>Framework</i>	Abstração que une códigos comuns entre vários projetos de <i>software</i> provendo uma funcionalidade genérica	José Fumo e Luthermilla Ecole	24.11.2017
Diagrama de Actividade	Gráfico de fluxo, mostrando o fluxo de controle de uma actividade para outra e são empregados para fazer a modelagem de aspectos dinâmicos do sistema	Vânia Cherindza	11.08.2017
Diagrama de componentes	Permite a modelagem física de um sistema, através da visão dos seus componentes e relacionamentos entre os mesmos	Albertina Mondjana	11.08.2017

Diagrama de instalação	Descreve os componentes de hardware e software e sua interação com outros elementos de suporte ao processamento	Albertinha Mondjana	11.08.2017
Diagrama de implantação	Utilizados para visualizar a topologia dos componentes físicos de um sistema em que os componentes de <i>software</i> são implantados	Vânia Cherindza	11.08.2017
Diagrama de Estados	Mostra os possíveis estados de um objecto e as transações responsáveis pelas suas mudanças de estado	Erick S. Mahanjane	11.08.2017
Diagrama de Implementação	Mostram a distribuição de <i>hardware</i> do sistema, identificando os servidores como nós do diagrama e a rede que relaciona os nós	Luthermilla Ecole	11.08.2017
Relatório de testes	Ferramenta utilizada para verificar e testar o <i>software</i> em concordância com o pressuposto inicialmente	Ercílio Samo	22.11.2017
Relatório de Progresso	Ferramenta que serve para acompanhar o desenrolar das funcionalidades e do disposto no sistema	Ercílio Samo	22.11.2017
Plano de implementação	Planificação de como será realizada a implementação do <i>software</i> , identificando aspectos como servidor, quantidade de <i>hosts</i> conectados entre outros	José Fumo e Erick S. Mahanjane	27.11.2017
Plano de implantação	Planificação de como será realizada a implantação do <i>software</i> , que permite verificar componentes físicos de um sistema	José Fumo	27.11.2017
Manual de Utilizador	Documento que serve de guia para o utilizador, com finalidade de familiarizar este com as funções oferecidas pelo <i>software</i>	Luthermilla Ecole	25.11.2017
Plano de	Planificação de actividades de garantia que o ciclo de vida do sistema continue fornecendo	Erick S. Mahanjane	27.11.2017

iXitolo LTDA ME
Contacto: ixitolo@gmail.com



manutenção	um conjunto de consultorias e outras actividade de manutenção		
------------	---	--	--

Uma vez que o projecto já foi implementado antes, não é necessário um plano específico de gestão de configuração para definir quais itens de configuração o sistema deve considerar na gestão de configuração.

3.1.2. Padrão de nomeação dos arquivos e pastas

Nesta seção é descrito o formato padrão de nomeação dos arquivos e pastas a ser seguido no Projecto.

Item	Padrão de Nome	Local Padrão de Armazenamento no Repositório
Plano Geral de Gestão de Configuração	Formato: Gestão de Configuração- Grupo 1.docx	https://github.com/emahanjane/EngenhariaSoftware
Plano de Gestão de Configuração para Banco de Dados	Formato: Gestão de Configuração para Banco de Dados.docx	
Plano de Gestão de Configuração do Projecto	Formato: iXitolo-PlanoGestãoConfiguração.docx	emahanjane/Engenharia de Software/Projecto-iXitolo/Documentação/Planos
Prospecção		
Proposta Técnica e Financeira	Formato: Proposta Técnica e Financeira- Grupo 1.docx	emahanjane/Engenharia de Software/Projecto-iXitolo/Documentação/Planos
<i>Concept Note</i>	Formato: Concept Note- Grupo 1.docx	emahanjane/Engenharia de Software/Projecto-iXitolo/Documentação/Planos
Escopo do projecto	Formato: Escopo do Projecto- Grupo 1.docx	emahanjane/Engenharia de Software/Projecto-iXitolo/Documentação/Planos
Análise		
Relatório da Reunião	Formato: Relatório da Reunião- Grupo 1.docx	emahanjane/Engenharia de Software/Projecto-iXitolo/Documentação/Relatórios
Relatório de	Formato: Relatório de Progresso 1- Grupo 1.docx	emahanjane/Engenharia de Software/Projecto-

iXitolo LTDA ME
 Contacto: ixitolo@gmail.com



Progresso 1		iXitolo/Documentação/Relatórios
Relatório de Progresso 2	Formato: Relatório de Progresso 2- Grupo 1.docx	emahanjane/Engenharia de Software/Projecto-iXitolo/Documentação/Relatórios
Apresentação Parcial	Formato: Apresentação 1- Grupo 1.pptx	emahanjane/Engenharia de Software/Projecto-iXitolo/Documentação/Apresentações
Planeamento		
Cronograma	Formato: iXitolo- Grupo 1.mpp	
Plano do Projecto	Formato: Projecto- Grupo 1.docx	
Apresentação Parcial	Formato: Apresentação 2- Grupo 1.pptx	emahanjane/Engenharia de Software/Projecto-iXitolo/Documentação/Apresentações
Desenvolvimento		
Escopo do Produto (Documento de Requisitos)	Formato: Escopo do Produto- Grupo 1.docx	emahanjane/Engenharia de Software/Projecto-iXitolo/Documentação/Planos
Documento de escopo	Formato: Escopo do Projecto (actualizado)- Grupo 1.docx	emahanjane/Engenharia de Software/Projecto-iXitolo/Documentação/Planos

iXitolo LTDA ME
Contacto: ixitolo@gmail.com



(actualizado)		
Base de Dados	Encontra-se disposto num conjunto de pastas	
Código fonte	Encontra-se disposto num conjunto de pastas	
Fechamento		
Apresentação Final	Formato: Apresentação Final- Grupo 1.pptx	emahanjane/Engenharia de Software/Projecto-iXitolo/Documentação/Apresentações
Relatório Final	Formato: Relatório Final- Grupo 1.docx	

3.1.2.1. SubProjectos

Alguns novos Projectos surgem por necessidade de se adequar Projectos anteriores. Estes Projectos se caracterizam por envolverem modificações nos requisitos iniciais tratados pelo Projecto “pai” e estas necessidades são descobertas geralmente quando o Projecto pai está em homologação ou logo após ter entrado em produção.

Os motivos desta necessidade são geralmente dois:

1. **Impactos causados por outros Projectos:** uma modificação no Projecto x impacta o requisito do Projecto y;
2. **Novas necessidades descobertas:** durante a homologação, ou em produção, descobre-se que o requisito x deveria tratar uma situação y que não foi considerada.

Nestes casos, a codificação do Projecto deverá refletir o vínculo com o Projecto anterior para que os envolvidos percebam de imediato este vinculado ao ler o código do Projecto. Para isto um subProjecto deverá ser numerado com um segundo conjunto de numeração iniciado do 1.

3.1.3. Estrutura de pastas padrão do Projecto

Esta seção descreve como a estrutura de pastas do Projecto deve estar organizada.

A organização da estrutura foi planejada pelo CCC e foi levado em conta necessidades e problemas de GC como merge, fechamento de linhas base, arquivamento de BRANCHES, entre outros. O local de cada pasta tem um porquê de estar lá e por isto reforçamos a importância de seguir esta estrutura para obter sucesso na GC.

A estrutura completa de pastas padrão do Projecto foi modelada e é ilustrada na **Error! Reference source not found..**

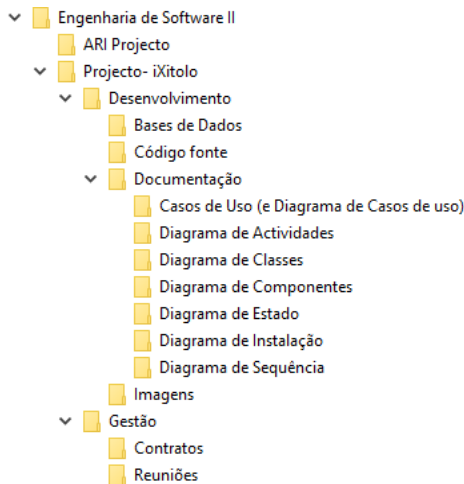


Figura 1. Estrutura completa de pastas padrão do projecto



Figura 2. Subpasta dentro a pasta de Documentação

O arquivo fonte da modelagem se encontra no repositório de versões, no endereço <https://github.com/emahanjane/EngenhariaSoftware>

3.1.3.1. Principais pastas do Projecto

Abaixo é descrito o papel das principais pastas do Projecto.

- **TRUNK:** A TRUNK representa a linha de desenvolvimento mais nova em produção. Ela será utilizada para criação de novas BRANCHES de correção de bugs (*hotfixes*) e novas BRANCHES de desenvolvimento (novos Projectos);
- **BRANCH:** A BRANCH representa uma nova linha de desenvolvimento, um desenvolvimento de melhoria a ser implantado no futuro, ou uma correção (*bugfix*) em produção. Elas representam um Projecto em execução, depois de finalizada ela será fechada;
- **TAG:** A *tag* representa uma versão estável que foi implantada em produção. *Tags* são imutáveis, são uma foto no tempo de uma versão que foi homologada ou correção de bug em produção;
- **DOC:** A pasta doc reúne toda a documentação do Projecto, como documentos de requisitos, casos de uso, cronogramas, planeamentos, aprovações, etc.

3.1.3.2. Ciclo de Vida das BRANCHES

Como dito anteriormente, uma BRANCH representa um desenvolvimento em andamento, tanto um novo desenvolvimento, quanto uma correção. Após o final do desenvolvimento, a BRANCH deve ser colocada em ambiente de homologação para a validação do cliente (solicitante do desenvolvimento).

Após sua validação, deverá ser realizado o merge da BRANCH para a TRUNK. Neste momento, esta modificação da TRUNK deverá ser replicada em todas as BRANCHES activas (em desenvolvimento e/ou homologação).

No final do processo esta BRANCH deverá ser desativada (excluída).

3.1.3.3. Desenvolvimento de novas versões

Novas versões de sistemas deverão ser desenvolvidas em novas BRANCHES. Para isto, antes do início do desenvolvimento do Projecto uma BRANCH deverá ser criada ou a partir da TRUNK ou de uma outra BRANCH que corresponda a necessidade do Projecto. As permissões de acesso da equipe na nova BRANCH deverão ser configuradas.

A nomenclatura para a BRANCH de desenvolvimento será:

UpdateiXitolo<xxx>

Onde <xxx> é o número da versão do RA em desenvolvimento.

3.1.3.4. Correção de bugs em produção

Correções de bugs em produção (*bugfix*) deverão ser realizadas em BRANCHES específicas para este fim.

A BRANCH deverá ser criada dentro da pasta “*branches/manutencao*”, pois a mesma já terá as permissões configuradas para a equipe desenvolvedora responsável pela manutenção. Com isto,

A nomenclatura para a BRANCH de manutenção será:

manutencao<yyyymmdd>_redmine#<

Onde:

- <yyyymmdd> é a data da criação da BRANCH. Ela é propositalmente no formato americano, pois permite uma melhor ordenação e visualização no directório;
- <identificador_redmine> é o nome do Projecto, seguido do tipo e o número da BRANCH aberta no Redmine.

3.1.3.5. Uso da TRUNK

Modificações na TRUNK somente serão permitidas quando BRANCHES forem homologadas pelo cliente, assim um merge trazendo tais modificações será permitido. Apenas o gestor de configurações terá permissões de realizar modificações na TRUNK.

A TRUNK é apenas o ponto focal para se criar novas BRANCHES. Caso *bugs* tenham sido gerados devido ao merge, quando uma cópia dele for criada para um novo desenvolvimento, como ela passará pelas fases de desenvolvimento, testes e homologação, este *bug* provavelmente será descoberto e sanado. No pior dos casos ele terá sido homologado pelo cliente.

3.1.3.6. Uso da TAG

Para evitar transferência de grande volume de dados entre servidor e estação de trabalho, não será adotado a forma convencional de tratamento de TAGs no GitHub. Em vez disso, será utilizado um arquivo texto onde cada *commit* representará uma TAG.

O arquivo deverá ser nomeado como TAG.txt dentro da pasta "tags". Os comentários commitados no arquivo serão o número da revisão correspondente na TRUNK, data do commit e o motivo da modificação (o nome do Projecto ou o motivo da correção do *bug*), ou seja, não precisará escrever dentro do arquivo, apenas no comentário do commit. O acesso será restrito ao gestor de configuração.

O formato do comentário deverá ser:

**"REVISAO:<<número de 6 dígitos>>; DATA:<<DD/MM/AAAA>>;
MOTIVO:<<descrição do motivo (se for comum, apresentar de forma resumida,
de contrário detalhar os factos)>>"**

3.1.3.7. Tratamento da Pasta Documentação

Novas manutenções evolutivas terão sua documentação mantida em nova pasta seguindo o padrão:

`emahanjane/Engenharia de Software/Projecto-iXitolo/Documentação`.

Dentro desta pasta teremos as subpastas para documentos de requisitos, cronograma, planos, modelagens, entre outros, conforme ilustrado na **Error! Reference source not found.**

3.1.4. Permissões de acesso

O acesso às pastas do GitHub será restrito, isto é, somente aqueles envolvidos no Projecto terão acesso ao mesmo.

Todos os utilizadores do GitHub pertencente ao quadro de colaboradores da UEM-Faculdade de Engenharia terão acesso de leitura a todos os Projectos existentes e todas as suas subpastas. Somente não terão acesso às pastas referentes a processos não relacionados a desenvolvimento, como a pasta do administrativo, por exemplo.

Utilizadores do GitHub que não pertencem ao quadro de colaboradores da UEM-Faculdade de Engenharia, estudantes e visitantes, por exemplo, somente terão acesso de leitura ou escrita aos Projectos que estão envolvidos. Ou seja, nos demais Projectos eles não poderão nem visualizar a pasta.

Abaixo são listados os perfis que tem acesso a cada uma das pastas explanadas anteriormente:

- TAG - somente o Gestor de Configuração terá permissão de escrita, pois será de responsabilidade dele actualizar o arquivo que contém as informações das TAGs;
- BRANCH DE DESENVOLVIMENTO – A restante equipa desenvolvedora/testadores somente terão permissão de escrita na BRANCH

do Projecto que está envolvido e durante o período de execução deste. Após concluído o Projecto, suas permissões de escrita deverão ser removidas. Isto irá evitar *commits* indevidos em Projectos que não àqueles que o desenvolvedor está alocado;

- BRANCH DE MANUTENÇÃO – a pasta “*branch/manutencao*” terá configurada as permissões da equipe de manutenção responsável do Sistema *iXitolo*. As BRANCHs criadas dentro desta pasta irão herdar as permissões e com isto não será necessário a definição da mesma em cada criação.
- TRUNK – somente o Gestor de Configuração terá permissão de escrita, para que o mesmo tenha o controle do que irá ou não para o ambiente de produção.

O Gestor de Projectos será responsável por conceder / aprovar permissões de acesso dos utilizadores nos Projectos e o Gestor da Tecnologia será responsável por executar a configuração no sistema de controle de versão.

3.1.5. **Baseline**

O trabalho a ser feito no Projecto deverá se basear na *baseline* de escopo, prazo, custo e qualidade, bem como atender somente a mudanças aprovadas pelo comitê de controle de mudanças (CCM). Para cada mudança aprovada, uma nova *baseline* deve ser fechada.

A tabela a seguir descreve em quais pontos do Projecto a *baseline* será estabelecida, quem as autoriza e o que vai nela.

Observação: Caso o escopo especificado leve mais de 1 mês para ser aprovada pelo cliente, nova análise de impacto deve ser refeita e assim nova linha base de todos os documentos da fase de prospecção devem ser fechadas.

Fase	Arquivo	Responsável	Formato e Local
Análise	Obrigatórios		
	Relatório de progresso	Gestor de Projecto	Formato: <nome do arquivo origem>-baseline-<yyyymmdd>.<extensão>
	Apresentação parcial	Gestor de <i>Marketing</i>	Local: Na mesma pasta do documento que originou a baseline
	Escopo do projecto	Gestor do Projecto	Formato: <nome do arquivo origem>-baseline-<yyyymmdd>.<extensão> Local: Na mesma pasta do documento que originou a baseline
	Escopo do produto	Gestor do Projecto	
	<i>Concept Note</i>	Gestor do Projecto	
	Relatório de Controle	Gestor do Configuração	
Planeamento	Obrigatórios		
	Proposta Técnica e Financeira	Gestor do Projecto	Formato: <nome do arquivo origem>-baseline-<yyyymmdd>.<extensão> Local: Na mesma pasta do documento que originou a baseline
	Plano do Projecto	Gestor do Projecto	
	Plano de Gestão de Configuração	Gestor de Configuração	
Desenvolvimento	Obrigatórios		
	Relatório de testes	Analista do Projecto	Formato: <nome do arquivo origem>-baseline-<yyyymmdd>.pdf Local: Na mesma pasta dos documentos UAT
	Relatório de progresso	Gestor de Projecto	
	TAG	Gestor de Configuração	Deve ser criado conforme descrito na seção 3.1.3.6.
	Guião de apresentação		Formato: <nome do arquivo origem>-baseline-<yyyymmdd>.pdf Local: Na mesma pasta do documento que originou a baseline
	Cronograma		Formato: <nome do arquivo origem>-baseline-<yyyymmdd>.<extensão> Local: Na mesma pasta do documento que originou a baseline

3.2. Gestão de Configuração de Base de Dados

A Gestão de Configuração para Base de Dados é tratado em um documento separado que pode ser encontrado na mesma pasta deste documento no repositório de versões

(<https://github.com/emahanjane/EngenhariaSoftware/blob/master/Documento%20de%20Bases%20de%20Dados.docx>).

3.3. Controle de Configurações e Mudanças

3.3.1. Solicitação de Mudança e Aprovação

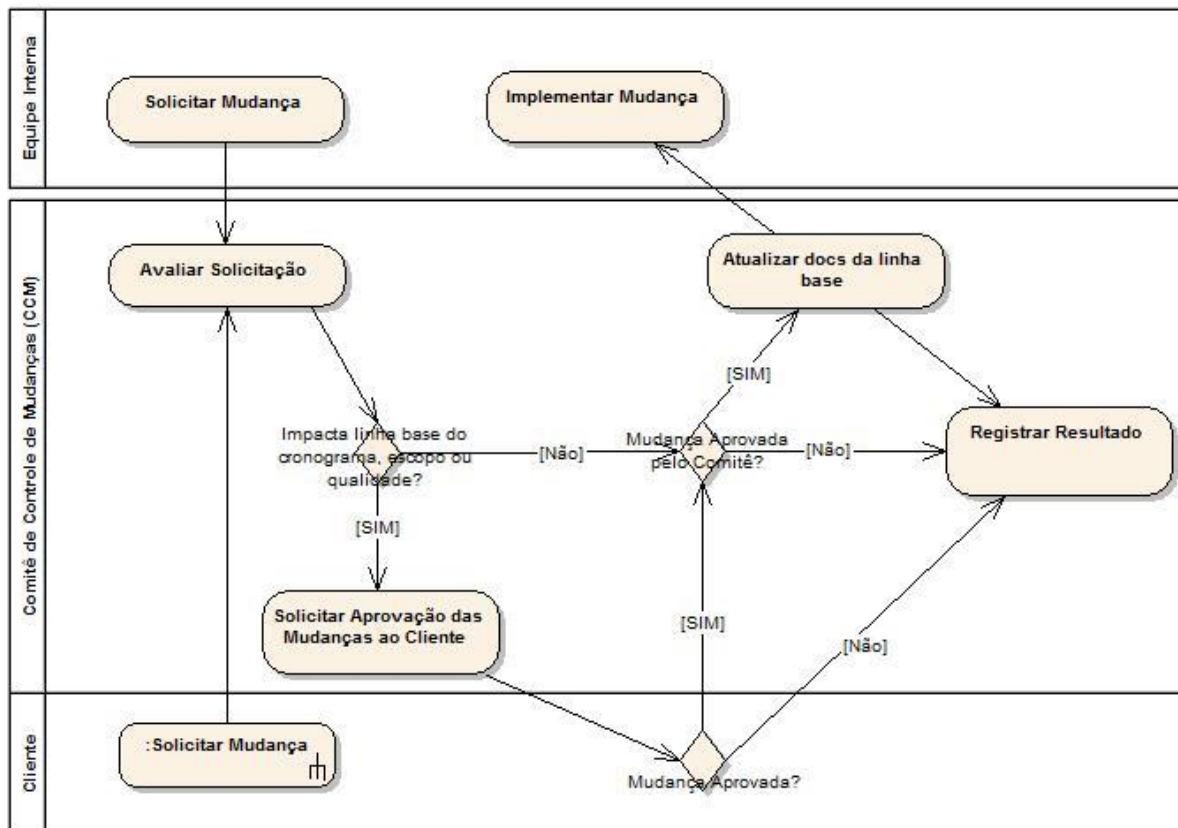
Descreve o processo pelo qual problemas e mudanças são submetidos, revisados e disponibilizados pelo comitê de Controle de Mudanças (CCM).

Solicitações de mudança podem ocorrer a qualquer momento no Projecto (iniciação, planeamento, execução, controle ou finalização). Quanto mais tarde no Projecto ocorrer, mais rígido e exigente deverá ser o processo de aprovação. Solicitações de mudança podem ser quanto ao escopo, cronograma, custos ou qualidade.

As solicitações de mudanças podem vir do cliente ou internamente da equipa desenvolvedora. Independente de onde vem, todas devem ser tratadas pelo CCM.

A seguir é ilustrado o fluxo de aprovação de solicitações de mudança vindas da equipe desenvolvedora interna ou do cliente.

act Processo ASM



iXitolo LTDA ME
Contacto: ixitolo@gmail.com



Figura 3 - Fluxo de Aprovação de Solicitação de Mudança

3.4. Registo e Comunicação do *Status* da Configuração

3.4.1. Processo de *Backup*

Descreve políticas de retenção, *backup*, desastre, e plano de recuperação. Também descreve quais médias devem ser mantidas (*online*, *off-line*, tipo de média e formato).

3.4.1.1. *Backup* do repositório GITHUB

Um *backup* completo do repositório deverá ser realizado mensalmente e *backups* incrementais diariamente. Após realizado o *backup* completo, uma limpeza dos *backups* incrementais anteriores poderá ser executada.

Os *backups* completos deverão ser gravados em média DVD e os *backups* incrementais em HD Externo.

Os *backups* incrementais são automatizados por *scripts shell* e executados através de *Jobs (cron Linux)*. O *script* dos *backups* incrementais está implementado no seguinte endereço do servidor café 192.30.253.113 (mesmo servidor do GITHUB): <https://github.com/emahanjane/EngenhariaSoftware/issues/1>.

3.4.2. Expurgo de Projectos Inativos

A cada 3 anos o repositório deverá ser renovado mantendo revisões dos últimos 2 anos, e Projectos inativos a mais de 2 anos deverão ser removidos.

Caso Projectos antigos precisem ser acedidos, um chamado para restauração do *backup* do Projecto deverá ser aberto com prazo de 1 semana.

3.4.3. Auditorias

O sector de garantia da qualidade do processo terá a responsabilidade de realizar auditorias esporádicas nos Projectos para validar a correta adoção das práticas de gestão de configuração. Qualquer desvio encontrado deverá ser registrado e reportado ao CCC que tomará as devidas acções. As auditorias deverão ser realizadas pelo analista do projecto, afim de certificar que as actividades desenvolvidas estão em concordância com o pressuposto.

4. Marcos de Projecto

Identifica marcos interno e do cliente relacionados ao esforço de GC do Projecto ou produto.

Marco	Interno ou do Cliente	Ações
Aprovação do documento de requisitos com esforço	Cliente	Fechar uma linha base do escopo e elaborar estimativa de execução do escopo aprovado.
Aprovação do esforço de execução do escopo aprovado	Cliente	Fechar uma linha base do esforço e elaborar cronograma de execução.
Aprovação do cronograma	Cliente	Fechar uma linha base do cronograma e divulgar o início da execução do Projecto.
Reunião de rotina com Cliente	Cliente e Interno	Permitir que todas as partes interessadas acompanhem o desenrolar do sistema, para permitir possíveis alterações sem prejudicar o produto final
Entrega do Módulo Clientes	Interno	Possibilitar aos Clientes realizar operações como cadastro, gestão de perfil e entre outras
Entrega do Módulo Compras	Interno	Permitir a realização de compras e vendas pelas partes dos clientes
Entrega do Módulo Administrativo	Interno	Controlar todos os utilizadores directamente conectados ao sistema, de forma a garantir a integridade dos parâmetros do sistema
Entrega do Módulo Transporte	Interno	Garantir a comunicação entre funcionários que prestam serviços a plataforma com os administradores, por forma a garantir melhor

		atendimento aos clientes
Fim da fase de desenvolvimento (implementação e testes)	Interno	Realizar implantação em homologação.
Fim da implantação em homologação	Interno	Realizar a validação do ambiente antes de liberar para o cliente.
Fim da validação do ambiente de homologação	Interno	Liberar para o cliente a homologação do Projecto.
Aceite do Projecto pelo cliente	Cliente	Pegar o aceite formal do cliente, realizar implantação em produção, repassar mudanças para todas as branches em aberto.
Fim do Projecto	Interno	Realizar a reunião de <i>feedback</i> do Projecto e registrar as lições aprendidas.

5. Recursos e Treinamento

Descreve as ferramentas de *software*, pessoal e treinamento requerido para implementar as especificações das actividades de GC.

5.1. Recursos

Segue lista de ferramentas utilizadas nos Projectos.

- *Enterprise Architect* (EA): ferramenta de modelagem de base de dados e *software* que apoia a GC. Através da exportação dos modelos em formato XML (Padrão XMI) é possível controlar versões dos modelos, bem como realizar merge (com certa limitação) nos arquivos.
- *Microsoft Office Word*: ferramenta de processamento de texto utilizada para elaborar documentos de requisitos, casos de uso, propostas, planos de Projecto, de testes, de implantação, entre outros;

- *Microsoft Office Excel*: planilha de processamento de dados utilizada para elaborar estimativas de esforço, cronogramas, controles, etc;
- *Microsoft Office Project*: ferramenta para elaboração e acompanhamento do cronograma de execução do Projecto;
- *Tortoise GITHUB*: ferramenta cliente para acesso e manipulação dos dados do repositório de versão de software;
- *Microsoft Visual Studio e Eclipse*: ferramenta IDE para codificação dos sistemas;
- *Ferramentas SQL Server*: ferramentas para elaboração e consulta de SQL's.
- *Adobe Acrobat Reader DC*: Ferramenta que permite realizar diversas acções sobre arquivos na extensão “.pdf”.

5.2. Treinamento

Para o controle de versão de bases de dados foi feita uma avaliação das ferramentas de modelagem *Power Designer* e *Enterprise Architect* (EA). O EA se saiu melhor na avaliação e será a ferramenta adotada pela UEM- Faculdade de Engenharia para a modelagem de bases de dados de *software*. Um treinamento de uso da ferramenta será necessário aos analistas responsáveis.

Um treinamento quanto aos padrões, diretrizes e práticas de gestão de configuração será realizado em todos os níveis da empresa.

5.2.1. Conceitos Básicos e Boas Práticas

Conceitos básicos e boas práticas podem ser vistos em nossa WIKI no endereço a seguir.

- [Endereço.](#)

Maiores informações sobre controle de versão com GITHUB podem ser vistas no seguinte *link*:

- [Endereço](#)

6. Subcontratação e Controle de Versão do Fornecedor

Fornecedores de desenvolvimento de *software* contratados deverão seguir nosso plano de gestão de configuração, salvar os códigos em nosso repositório de versões, reportar o andamento das actividades em nossa ferramenta de gestão dos Projectos e se reportarem a nossos gestores de Projectos. Assim, conseguindo-se manter a qualidade e integridade do projeto e produto final.

7. Esclarecimentos

Durante a realização do presente plano de gestão de configuração foram apresentadas três entidades. Para melhor percepção destas, a seguir são descritas a sua posição e significado para o projecto.

- **Universidade Eduardo Mondlane:** representa a empresa contratante do projecto, visto que se trata de um plano realizado em uma disciplina do curso de Engenharia Informática da mesma instituição, sendo especificada pela sigla UEM e referida em termos da Faculdade de Engenharia;
- **Supervisor** (ou docente Supervisor): representa o individualizado, cliente, responsável por desempenhar funções de um representante da empresa por forma a verificar, certificar, validar e homologar todos os termos acordados para o projecto. É o regente da disciplina;

iXitolo LTDA ME
Contacto: ixitolo@gmail.com



- **Equipe desenvolvedora:** representa o grupo contratado para realizar as tarefas para alcançar os objectivos. É formado por estudantes que frequentam o curso e que estejam a realizar a disciplina respectiva.