

## ANEXO X - TERMO DE REFERÊNCIA

### PADRÃO DOS AMBIENTES DE CONSTRUÇÃO DE DESENVOLVIMENTO DE SISTEMAS

#### 1. Introdução

Um ambiente é um conjunto de hardware e software, com uma configuração específica, estabelecida para um propósito pré-definido. Com o intuito de documentar e uniformizar os procedimentos no âmbito da Enap, este documento visa a padronização dos ambientes de desenvolvimento, teste, homologação e produção para a construção de sistemas de informação. Para cada um dos ambientes, a plataforma a ser utilizada e os recursos de hardware e software precisam ser registrados, assim como os procedimentos específicos de integração e uniformidade entre os ambientes, cabe ressaltar que a versão dos aplicativos dos servidores de aplicação e base de dados devem ser iguais em todos os ambientes, a fim de garantir homogeneidade, integridade e integração entre os ambientes.

A integração tem sido considerada o elemento central para prover um suporte efetivo ao desenvolvimento de software, pois é ela que define todas as regras e diretrizes que governarão o uso das ferramentas. A integração é responsável pela combinação de ferramentas de forma que elas trabalhem em harmonia em todas as etapas do processo. Existem vários níveis de integração de ferramentas, diferindo no suporte fornecido. Dentre eles, pode-se citar:

- **Integração de Dados:** A chave para integrar ferramentas é a habilidade de compartilhar informação de projeto.
- **Integração de Apresentação:** A integração de apresentação tem como objetivo tornar as interfaces do ambiente de construção de sistemas consistentes, permitindo que o usuário utilize as ferramentas, alternando de uma para a outra, sem sofrer um impacto na interface. Os componentes utilizados são os mesmos, com as mesmas funcionalidades, não havendo a necessidade de aprender como utilizar cada ferramenta.
- **Integração de Controle:** Para integrar um conjunto de ferramentas é preciso ter um foco forte no gerenciamento do processo. Engenheiros de software têm se tornado mais orientados a processos, usando métodos, técnicas e ferramentas para controlar e guiar seu trabalho.
- **Integração de Conhecimento:** Com o aumento da complexidade dos processos de software faz-se necessário considerar informações de natureza semântica, sem as quais a integração não é plenamente obtida. O conhecimento, assim como os dados, deve estar disponível e representado de forma única no ambiente, de forma a poder ser acessado por todas as ferramentas que dele necessitarem, evitando redundância e inconsistências

#### 2. Ambiente de Desenvolvimento

Será o ambiente a ser utilizado pela equipe de desenvolvimento de software, com o objetivo de implementar a solução. No desenvolvimento dos sistemas deverão ser seguidos os padrões de identidade visual ([identidadevisual.enap.gov.br](http://identidadevisual.enap.gov.br), padrão de banco de dados, o Metodologia de Desenvolvimento de Software (MDS) e as normas da política de segurança da Enap.

##### 2.1. Criação e manutenção do projeto de sistema

O projeto de sistema será criado no ambiente de desenvolvimento seguindo o padrão: <http://siglasistema.dev.enap.gov.br>, onde:

- siglasistema = Nome (sigla) do sistema em letras minúsculas.
- dev = Indicação do nome do ambiente de desenvolvimento.

Por exemplo: <http://suap.dev.enap.gov.br>

A base de dados a ser utilizado para o ambiente de desenvolvimento será uma base com as tabelas corporativas e as tabelas específicas do sistema, com os usuários com senha e perfil de desenvolvimento. A nomenclatura para o nome do esquema do banco de dados do projeto será: `db_siglasistema`, onde:

- db = Indicação que se trata de uma base de banco de dados.
- siglasistema = Nome (sigla) do sistema em letras minúsculas.

Somente os desenvolvedores que trabalharão no projeto de sistema serão habilitados para realização de check-out e check-in dos códigos fontes do projeto. Enquanto o desenvolvedor estiver codificando o código do módulo

(funcionalidade), este deverá utilizar o ambiente local de desenvolvimento da enap em **\*.dev.enap.gov.br**. Para cada projeto poderá existir um ambiente específico devidamente configurado:

Somente após os testes iniciais e verificação de consistência da aplicação, que deverá ser realizado o chek-out do módulo (funcionalidade) para o ambiente de desenvolvimento. No caso específico do ambiente de desenvolvimento, é interessante identificar que o desenvolvedor será o responsável pela geração dos releases e pela execução de testes básicos de integração. Estes testes são muito mais simples do que os testes a serem executados no ambiente de homologação. O objetivo é assegurar que o release a ser encaminhado ao ambiente de homologação está funcionando minimamente, isto é, assegurar que a instalação, a configuração inicial e algumas funcionalidades básicas se apresentam sem problemas.

Este repositório e sua infraestrutura completa, são consequência direta das características das ferramentas de desenvolvimento utilizadas e dos processos de desenvolvimento adotados. Segue na tabela abaixo, a configuração para este ambiente de desenvolvimento na Enap:

Servidor de Aplicação – Ambiente de Desenvolvimento			
Endereço	Software	Finalidade	Usuário
<a href="http://siglasistema.dev.enap.gov.br">http://siglasistema.dev.enap.gov.br</a>	Git Lab e Jenkins	- Desenvolvimento do sistema - Geração de releases	Equipe de Desenvolvimento
<a href="http://siglasistema.hom.enap.gov.br">http://siglasistema.hom.enap.gov.br</a>	Git Lab e Jenkins	- Validação das releases	Cliente e Gerente do projeto

Base de Dados - Ambiente de Desenvolvimento			
Base	Software	Finalidade	Usuário
<b>db_siglasistema</b> Base nova, exceto para o esquema corporativos (acesso-segurança, estados, municípios, unidades...) Para a tabela de pessoas (usuários) se houver, realizar a cópia do ambiente de produção e alterar todas as senhas para teste123456.	SQL Developer	- Manutenção da base de dados do projeto	Equipe de Desenvolvimento
	DB Designer	- Elaboração e validação do modelo lógico de dados do do projeto	Analista de Sistema e Analista de Banco de dados

### 2.1.1. Estrutura de Grupos e Projetos para Codificação

Todos os projetos deverão ser hospedados no controle de versão com GITLAB em [https://gitlab.enap.gov.br/grupo\\_departamento/nome\\_do\\_projeto](https://gitlab.enap.gov.br/grupo_departamento/nome_do_projeto).

Os nomes dos projetos deverão seguir a nomenclatura de enap-**NOMEDOPROJETO** e fazer parte do subdiretório (grupo) do departamento responsável, exemplo, **cgti-sistemas** para os projetos destinados ao departamento da **CGTI**. Subdiretório **EscolaVirtual** para os projetos destinados ao departamento da **CGEAD**.

Exemplo: <https://gitlab.enap.gov.br/cgti-sistemas/enap-suap>.

Os nomes dos subdiretórios poderão se adaptar conforme as necessidades a fim de facilitar a navegação e identificação dos projetos de acordo com os seus objetivos. Exemplo de grupos:

- cgti-sistemas: Agrupa todos os projetos de sistemas da coordenação-geral de TI;
- cgti-sistemas-core: Componentes da arquitetura Core dos Sistemas Symfony e Angularjs;
- cgti-identidade-visual: Grupo para manter os padrões de identidade visual para sistemas ou portais;

### 2.1.2. Estrutura para Documentação do sistema

A documentação do sistema deverá ser armazenada e versionada no repositório do Gitlab no na pasta “documentação” de cada projeto, conforme exemplo abaixo:

master

enap-avalienap /

+ v

Name

Last commit

Last update

- 1 - Planejamento
- 2 - Analise e designer
- 3 - Homologação
- 4 - Treinamento e manuais
- 5 - Implantação
- 6 - Metricas
- 7 - Backlog

Os projetos também deverão ser documentados no controle de versão Gitlab no arquivo **README.md** contendo todos os passos necessários para restauração do ambiente e suas dependências.

## 2.2. Versionamento de codificação e documentação

Será utilizado um sistema de versionamento para o controle de códigos-fonte e versionamento de releases (Tags) pelos desenvolvedores, assim como, para a documentação do projeto pelos desenvolvedores da aplicação.

Para cada release de codificação a ser gerada para os ambientes, deve seguir o seguinte padrão: v.X.Y.Z, onde:

- X = Versão que indica mudança que torna incompatível com a versão anterior
- Y = Versão que indica mudança compatível com a versão anterior
- Z = Versão que indica mudanças menores e correções de erros

Por exemplo: SUAP 2.2.0

Para cada release documentar o motivo da mudança, descrevendo as melhorias, correções e procedimentos de implantação caso haja dependências de funcionalidades.

Exemplo:

### SUAP 2.2.0

#### Módulo Educacional

- Incluir campo cor no cadastro de Tipo de Curso (#127)
- Novo recurso para listar classificações de cursos na API do Serviços ENAP (#133)
- Melhoria na apresentação dos horários das aulas (visão do aluno, professor e turma)
- Permitindo que coordenadores de curso cadastrem materiais comuns a todas as turmas
- Melhorias nas telas de listagem e cadastro de cursos (#134)
- Melhoria nas telas de cadastro e vinculação de material em diários (#136)
- Permitindo que Coordenadores de curso editem configurações de avaliação no diário (#138)
- Ajustes de nomenclatura e exibição dos materiais vinculados ao curso nas turmas virtuais do SUAP (#137) Ajustes no admin de t
- Permitindo que professores enviem mensagem para os alunos.

## 3. Ambiente de Homologação

Este ambiente simula o ambiente de produção da Enap. É utilizado para validar o produto antes de colocá-lo em real funcionamento.

O ambiente de homologação deve ser trabalhado e mantido para ser similar ao ambiente de produção. Seu objetivo é oferecer aos futuros usuários do sistema a possibilidade de testar as funcionalidades dos novos produtos de desenvolvimento e encontrar possíveis incorreções de resultado ou comportamento. Estes desvios devem ser relatados à equipe de desenvolvimento responsável pela solução que providenciará as alterações necessárias. Da mesma forma que este ambiente serve para a análise prévia das funcionalidades do novo sistema, ele também deve servir para que as equipes de desenvolvimento e suporte a rede e banco executem seus procedimentos de instalação e atualização antes que os produtos de desenvolvimento sejam implantados em produção.

É importante ressaltar os seguintes pontos:

1. Ambientes de homologação não devem ser utilizados pelas equipes de desenvolvimento para testes.
2. Desenvolvedores não devem possuir acesso privilegiado ao ambiente de homologação. Para evitar que os desenvolvedores alterem configurações ou efetuem cópias não autorizadas de versões de sistema ou massas de dados, os desenvolvedores não devem ter característica de acesso privilegiado ao ambiente de homologação. Isso permite com que o desenvolvedor, ao acessar sistemas em homologação, tenha a mesma experiência que o usuário final, podendo reproduzir eventuais problemas encontrados por ele.
3. O ambiente de homologação será uma cópia do ambiente de produção mais a nova versão do produto ou projeto. Assim os testes do usuário serão muito mais eficientes.

Segue, na tabela abaixo, a configuração para este ambiente.

SERVIDOR DE APLICAÇÃO – Ambiente de homologação		
Endereço	Finalidade	Usuário
http://siglasistema.hom.enap.gov.br	- Realização da homologação (testes) pelos usuários.	Gestor de Negócio, Requisitantes e Usuários
	- Executar os procedimentos de instalação e atualização antes que os produtos de desenvolvimento sejam implantados em produção.	Equipes de suporte a rede e banco de dados.

BASE DE DADOS – Ambiente de homologação			
Base	Software	Finalidade	Usuário
Cópia da base do ambiente de produção, atualizada mensalmente. Em casos excepcionais poderá ocorrer atualização específica para o projeto, caso seja solicitado pelo Gestor de negócio.	SQL Developer	- Consultar a base de dados do projeto	Equipe de banco de dados

Após a realização de todos os testes e suas correções, deverá ser gerado uma release no formato citados no item 2.2 para ser inserida no ambiente de treinamento se for o caso, e no ambiente de produção.

#### 4. Ambiente de Produção

Este é o ambiente no qual os sistemas da Enap estão operacionais e funcionais, em que o produto é efetivamente colocado em funcionamento. Os recursos deste ambiente deverão estar alinhados com os recursos da homologação, tendo como adendo o hardware e software complementar (firewall, licenças, etc).

SERVIDOR DE APLICAÇÃO – Ambiente de produção		
Endereço	Finalidade	Usuário
http://siglasistema.enap.gov.br Ex: <a href="http://suap.enap.gov.br">http://suap.enap.gov.br</a>	- Utilização e operacionalização do sistema.	Usuários do sistema

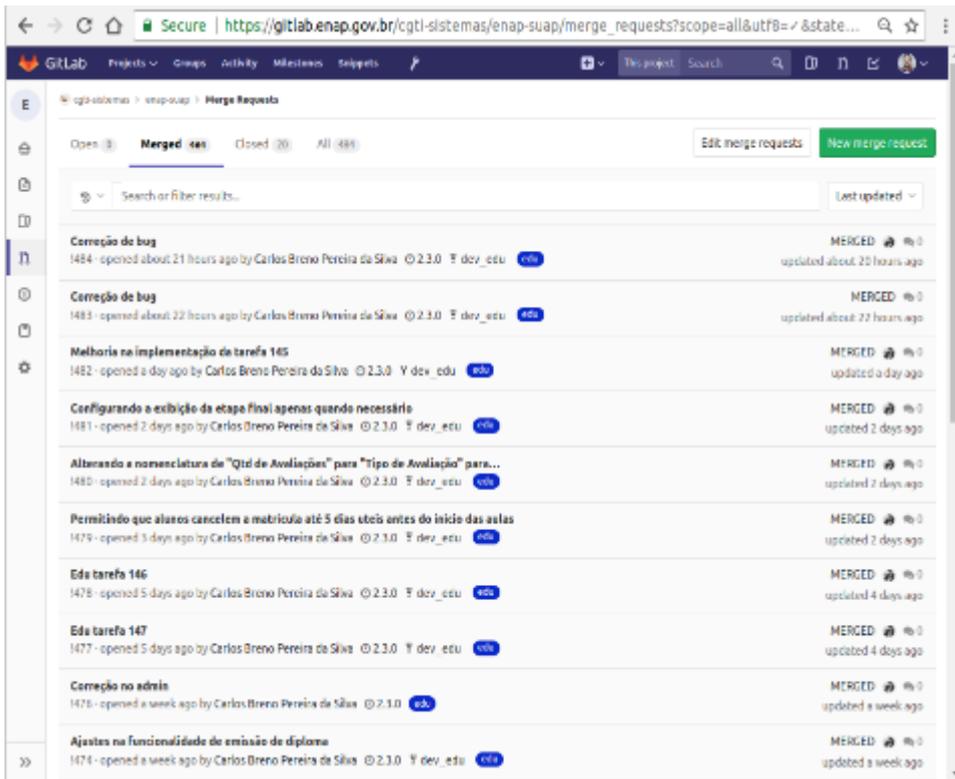
BASE DE DADOS – Ambiente de produção			
Base	Software	Finalidade	Usuário
Base produção	Postgres, MySQL SQL Server	- Disponibilizar a base de dados do projeto	Equipe de banco de dados

#### 5. Promoção entre os ambientes e o Versionamento

Promover uma aplicação de um ambiente de Desenvolvimento para um ambiente de Homologação significa instalar, no ambiente de Homologação uma versão da aplicação que atingiu com sucesso seus propósitos no ambiente de Desenvolvimento. O objetivo é verificar se essa mesma versão da aplicação consegue ser validada no novo ambiente de Homologação.

Nesta seção, deve ser definido em que momentos do projeto a aplicação será promovida entre os ambientes. No caso da Enap é utilizado o Git Lab e o Jenkins como repositório transiente para que a promoção entre ambientes seja realizada. Para as aplicações legadas, após a conclusão de sua manutenção, o desenvolvedor deve publicar as

alterações no Git para que sejam publicadas na Homologação para que o usuário final aceite as modificações. Somente após este “de acordo”, é que as alterações deverão ser publicadas no ambiente de Produção.



## 5.1. Utilização do Repositório de Versionamento GitLAB

### 5.1.1 Permissões

Os integrantes da Fábrica de Software contratada não poderão criar grupos nem projetos, ou ainda adicionar novos membros ao projeto, ficando estas ações a cargo dos Analistas em TI da CGTI-Enap.

### 5.1.2 Perfis

Tais integrantes terão somente o perfil de *Developer* (para mais informações sobre o perfil consultar: <https://goo.gl/Otw6hp>) no git. Para isso, o preposto da Fábrica de Software solicitará aos Analistas em TI da CGTI acesso a ferramenta de versionamento supracitada.

### 5.1.3. Utilização e Operação

#### 5.1.3.1 Branch

Para cada ambiente (Produção, Homologação e Desenvolvimento), existirá uma branch específica com o seu devido apontamento da seguinte forma:

Branch	Ambiente
Master	Ambiente de Produção (*.enap.gov.br)
Homologação	Ambiente de Homologação (*.hom.enap.gov.br)
Desenvolvimento	Ambiente de Desenvolvimento (*.dev.enap.gov.br)

A Branch neste contexto significa um fluxo entre passagens dos ambientes, para controle das requisições (merge request) entre eles.

#### 5.1.3.2 Master

Os commits realizados na *branch* (por padrão será usada a 'Master') deverão ser acompanhados de mensagem indicando sua natureza e seguirá o seguinte padrão: - número da OS/tipo da demanda ('evolutiva': em caso de novo produto ou novas funcionalidade, ou 'corretiva' quando da correção de incidentes). Já o número da Ordem de Serviço respeitará a natureza dos sistemas onde foram solicitados: CITSMART em caso de sustentação e READMINE em caso de projeto.

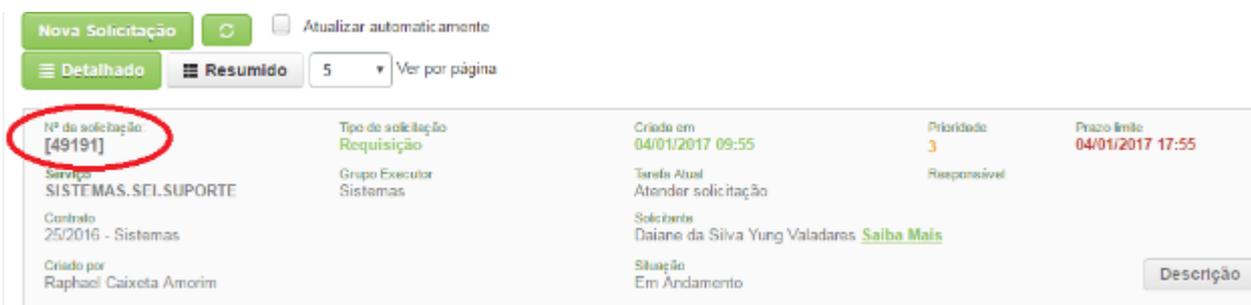


Figura 1 - Tela do CITISMART destacando o número da OS que será usado na mensagem do commit para o gitlab.



Figura 2 - Tela do REDMINE destacando o número da OS que será usado na mensagem do commit para o gitlab.

As *Branches* receberão como nome o Número da OS à qual está sendo atendida e a indicação de que seja uma demanda corretiva ou evolutiva. O Número da OS será o mesmo número criado pelo CITISMART ou READMINE, de acordo com a natureza da solicitação, sustentação ou projeto, respectivamente. Além disso, o pacote de entrega da OS deverá seguir a mesma lógica de acordo com o exemplo abaixo

Exemplo: 7855-Corretiva, 3496-Evolutiva;

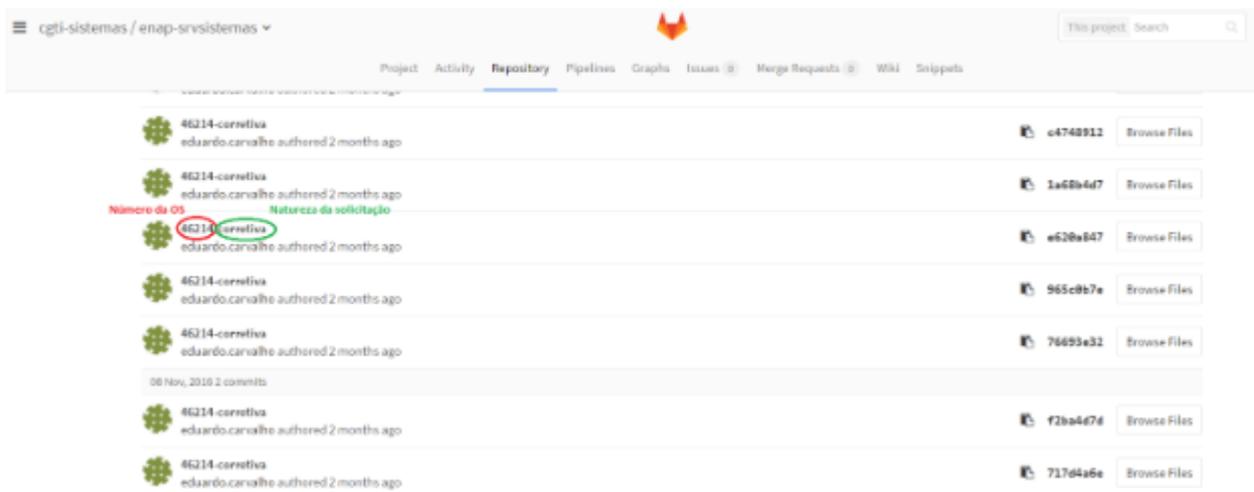


Figura 3 - Exemplo de mensagem do commit de natureza corretiva

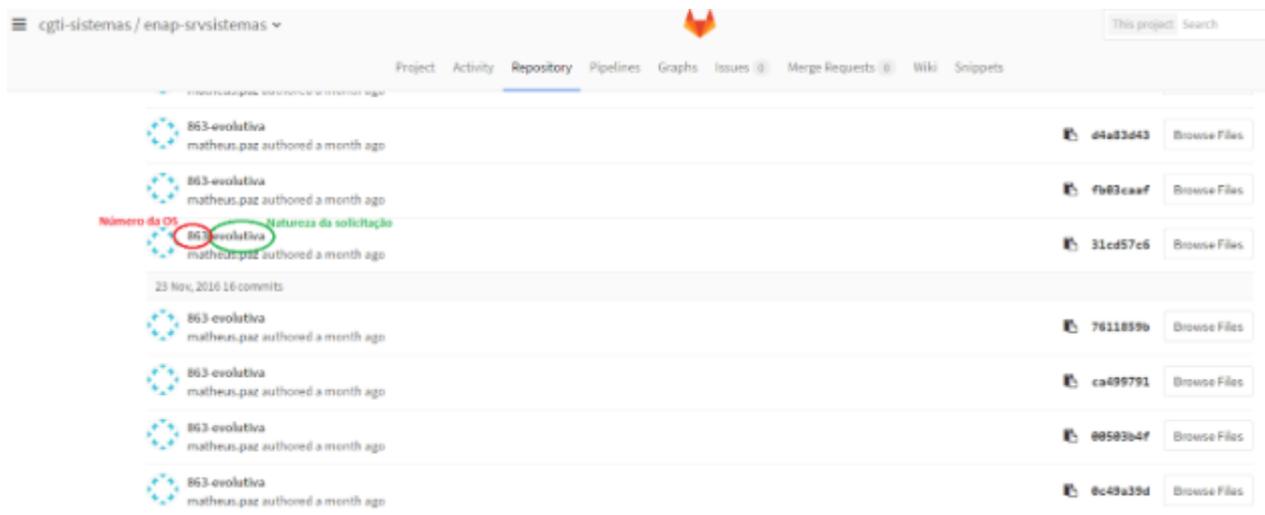


Figura 4 - Exemplo de mensagem do commit de natureza evolutiva

A homologação será realizada com base no último *commit* feito na *branch* criada pelo desenvolvedor. Após este procedimento o setor de infraestrutura publicará no ambiente de homologação, onde será validado pelo setor que gerou a demanda juntamente com representantes da fábrica.

Finalizado este procedimento, caso haja validação por parte do cliente, o referido *commit* passará por um *merge request* para a *branch* master, onde passará pelo processo de tagueamento (segundo as orientações deste documento).

### Versionamento Semântico Aplicado

#### 2.0 Regras de Numeração de Versão (*Tags*)

O esquema de numeração de versões adotado é baseado no esquema adotado pela organização Apache Foundation, definindo que uma versão é composta por quatro números inteiros:

v X.Y.Z

sendo que a alteração desses números segue o critério:

‘v’ : para indicar que se trata de uma versão do projeto.

X - MAIOR: Existe modificação na estrutura de dados ou na arquitetura do sistema, incremental;

Y - MENOR: Existe modificação na inclusão de um ou mais conjuntos de novas funcionalidades. Inicia com zero (zero) e deve ser reiniciado sempre que houver a troca da versão maior;

Z - MICRO: Existe correção de erros ou correção de comportamentos esperados no sistema. Inicia com zero (zero) e deve ser reiniciado sempre que houver a troca da versão maior ou da versão menor.

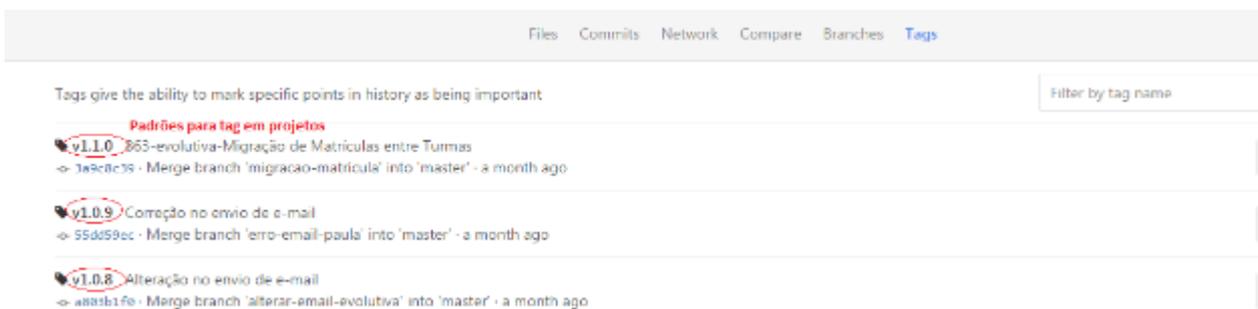


Figura 5 - Exemplo de tag para especificar projetos para ambiente de produção