

**UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS QUIXADÁ**

**PLANO DE GERÊNCIA DE CONFIGURAÇÃO DO SISTEMA DE CONTROLE DE
FLUXO DE GALINHAS (SCFG)**

Equipe:

Carlos Alberto, Francisca Beatriz, Gabriel Uchôa, Johnny Marcos, José Robertty

Professor: Jéssyka Vilela

**Abril
2018**

SUMÁRIO

1	Introdução	3
1.1	Objetivos	4
1.1.1	Organização do Documento	4
2	Papéis e Responsabilidades	4
3	Plano de Configuração	5
3.1	Controle de Configuração	5
3.1.1	Estrutura do Repositório de Gerência de Configuração	5
3.2	Política de Segurança	8
4	Métodos de Identificação	9
4.1	Documentos	9
4.2	Configuração Base	10
4.3	Documentos de Análise / Projeto de Software	10
5	Controle de Mudanças	10
6	Ambiente e Ferramentas	12
6.1	Plano de Software	12
7	Versão de Entrega do Produto	13
	Relatório da Equipe	14

1 Introdução

A Gerência de Configuração é um conjunto de práticas de assistência que permitem controlar as alterações que acontecem ao decorrer do desenvolvimento do projeto, mantendo o projeto estável durante sua evolução. Para ajudar a cumprir esta função, a Gerência de Configuração conta com algumas práticas: Controle de Versão, Controle de Mudanças e Integração Contínua.

O Controle de Versão é responsável por identificar, armazenar e gerenciar os itens de configuração e suas versões durante o período de vida do software; já o Controle de Mudanças assegura que todas as alterações aconteçam de forma controlada e otimizada; e a Integração Contínua é responsável por certificar que as alterações no projeto são feitas, testadas e relatadas assim que possível logo após serem inseridas.

Algumas das principais ferramentas utilizadas para uma boa execução da Gerência de Configuração são Concurrent Versions System (CVS), Subversion (SVN) e Git.

O CVS é um sistema open source, de estrutura cliente/servidor, de controle de versão que possibilita lidar com inúmeras versões de arquivo dispostas em um diretório, preservando suas versões anteriores e os logs dos responsáveis pela manipulação do arquivo. O SVN também é um controle de sistema de versão open source que gere arquivos e diretórios monitorando as modificações realizadas durante o desenvolvimento do projeto. Um de seus pontos fortes é a utilização em rede, que proporciona o trabalho de forma colaborativa entre os usuários. Permite a recuperação de arquivos antigos ou acessar o histórico de alterações. Já o Git se assemelha ao propósito geral das ferramentas anteriormente citadas, com o acréscimo de proporcionar um mecanismo de gerenciamento de código fonte. Ele conta com repositórios que disponibilizam o histórico completo e possuem a habilidade de acompanhamento total das revisões.

Os sistemas embarcados também contam com a necessidade da implementação de Gerência de Configuração e, apesar de existir uma carência de ferramentas que contemplem, com o mesmo alto grau de importância, a evolução de projetos de sistemas embarcados, existem práticas e ferramentas que desempenham papel importante na Gerência de Configuração.

No que diz respeito às práticas, temos o planejamento correto e o acompanhamento das tarefas desenvolvidas ao longo do ciclo de vida do projeto. Já em relação às ferramentas, temos o Project Management Body of Knowledge (PMBOK) e o Rational Unified Process (RUP).

O PMBoK, é um conjunto de técnicas na gestão de processo, oferecendo padronização de tarefas ao gerenciar um projeto, aperfeiçoamento no fluxo de comunicação entre os envolvidos e controle sobre a direção do projeto. O RUP oferta uma abordagem constituída em disciplinas para atribuir atividades e responsabilidades em uma organização de desenvolvimento. Sua principal meta é garantir a produção de alta qualidade do software.

Este documento, cujo modelo foi obtido através de um template pronto disponível em: <https://liliansimao.wikispaces.com/>, relata o Plano de Gerência de Configuração para o projeto em desenvolvimento do Sistema de Controle de Fluxo de Galinhas — SCFG, onde o GitHub foi utilizado como ferramenta.

1.1 Objetivos

O presente documento tem por objetivo apresentar a organização, nomenclatura e regras de versionamento para a gerência de configuração do projeto de desenvolvimento do SCFG.

Este plano é destinado a todos os integrantes da equipe responsáveis pelo desenvolvimento do sistema.

1.1.1 Organização do Documento

As seções subseqüentes deste documento estão assim organizadas:

- Seção 2, é descrito os papéis e responsabilidades da gerência de configuração;
- Seção 3 é apresenta o plano de configuração onde é definido a estrutura do armazenamento, as configurações bases do projeto, o controle de configuração e as políticas de segurança e acesso aos itens de configuração;
- Seção 4 define como este plano será evoluído;
- Seção 5 define os métodos de controle de mudanças;
- Seção 6 apresenta os ambientes de desenvolvimento do projeto;
- Seção 7 exhibe os itens que estarão dispostos na versão de entrega do projeto.

2 Papéis e Responsabilidades

Papel	Membro	Responsabilidade
Gerente de Desenvolvimento (GD)	Beatriz Precebes	Juntamente com a CCM receber, analisar e aprovar os PFM.
Líder de Projeto (LP)	Gabriel Uchôa	Planejar as atividades de GC juntamente com o Responsável pela Configuração, designar executante, finalizar SM, autorizar a criação das configurações bases conforme descrito na seção Plano de Configuração.
Responsável pela Configuração (RC)	Robertty Freitas	Criar e manter infra-estrutura corporativa (servidores) de GC; Implementar as políticas de Controle de Acesso ao ambiente de GC, Realizar os backups dos repositório de configuração dos projetos.

Tabela 1 - Papéis e responsabilidades

3 Plano de Configuração

A estrutura de pastas utilizadas para o controle de versionamento é descrita como:

Trunk – pasta que contém a estrutura definida no item. Sua finalidade é receber todos os artefatos. A equipe armazena nesta pasta todas as versões de trabalho dos documentos ou códigos.

Branches – pasta que armazena os documentos de uma versão que está sofrendo uma mudança diferente da linha normal de desenvolvimento.

Tags – pasta que armazena as configurações bases do projeto. Estes itens de configuração representam versões-base para entrega e não sofrerão mais mudanças.

3.1 Controle de Configuração

Para o armazenamento dos artefatos de projeto a equipe utiliza um repositório denominado SCFG do GitHub e está disponível no seguinte link: <https://github.com/RoberttyFreitas/SCFG>.

O controle de versionamento dos itens de configuração ocorrerá através da própria ferramenta. As baselines ou configuração base quando aprovadas receberá uma versão manual definido de acordo com o item 4.2 deste documento.

As atualizações nos itens de configuração no repositório ocorre através da execução do comando **git commit -m “[mensagem descritiva]”** da ferramenta sempre que o autor julgar conveniente a atualização de um nova versão do item. Caso o integrante da equipe necessite trabalhar deverá executar o comando **git clone [URL]** para obter a versão mais recente do repositório, ou **git branch [nome do branch]** para experimentar e realizar edições no repositório (ou trabalhar em diferentes versões do projeto) antes da realização do commit.

As versões aprovadas para fazer parte de uma baseline deverão ser bloqueadas através da configuração de cada branch que será bloqueado, impedindo-o de ser excluído por terceiros ou sobrescrito sem antes uma análise de **Pull Request**. O desbloqueio ocorrerá somente após aprovação das mudanças submetidas ao processo formal de mudanças definido no item 5 deste documento. Somente o Responsável pela Configuração pode realizar o desbloqueio e uma nova versão de trabalho deve ser gerada.

Os itens de configuração gerenciados serão:


- Código fonte principal do projeto
- Código da biblioteca SPI
- Plano de projeto da Antena RFID




3.1.1 Estrutura do Repositório de Gerência de Configuração

A seguir será apresenta a estrutura definida para armazenamento dos artefatos do projeto no repositório.

SCFG
+-branches
+-tags




- +trunk
 - | +-apps
 - | +-docs
 - | +-ambiente
 - | +-ata
 - | | +-configuração
 - | | +-e-mail
 - | +-especificação
 - | | +-arquitetura
 - | | +-protótipos
 - | | +-requisitos
 - | +-gerenciamento
 - | | +-aceites
 - | | +-planilha_fpa
 - | +-implantação
 - | +-validação_verificação
 - | | +-artefatos_de_teste
 - | | +-checklist_revisão_por_pares_casos_de_teste
 - | | +-checklist_revisão_por_pares_especificação
 - | | +-checklist_revisão_por_pares_implementação


[RoberttyFreitas / SCFG](#)


 Watch
 0
 Star
 0
 Fork
 0

[Code](#)
[Issues 0](#)
[Pull requests 0](#)
[Projects 0](#)
[Wiki](#)
[Insights](#)




Repositório irá armazenar o controle de versão do Sistema de Controle de Fluxo de Versão (SCFG)

 9 commits
  2 branches
  1 release
  1 contributor

Branch: [master](#)
[New pull request](#)
[Create new file](#)
[Upload files](#)
[Find file](#)
[Clone or download](#)


[ferreiracaf](#)
 Organização do repositório

Latest commit a1fedce 27 minutes ago

 app	Estruturação do repositório	42 minutes ago
 docs	Estruturação do repositório	42 minutes ago
 README.md	Estruturação do repositório	42 minutes ago

RoberttyFreitas / SCFG

Watch

0

Star

0

Fork

0

<> Code

Issues 0

Pull requests 0

Projects 0

Wiki

Insights

Branch: master

SCFG / docs /

Create new file

Upload files

Find file

History

ferreiracaf Estruturação do repositório

Latest commit 1ab29de 44 minutes ago

..

ambiente

Estruturação do repositório

44 minutes ago

ata

Estruturação do repositório

44 minutes ago

especificacao

Estruturação do repositório

44 minutes ago

gerenciamento

Estruturação do repositório

44 minutes ago

implantacao

Estruturação do repositório

44 minutes ago

validacao_verificacao

Estruturação do repositório

44 minutes ago

RoberttyFreitas / SCFG

Watch

0

Star

0

Fork

0

<> Code

Issues 0

Pull requests 0

Projects 0

Wiki

Insights

Branch: master

SCFG / docs / ata /

Create new file

Upload files

Find file

History

ferreiracaf Estruturação do repositório

Latest commit 1ab29de an hour ago

..

configuracao

Estruturação do repositório

an hour ago

email

Estruturação do repositório

an hour ago

RoberttyFreitas / SCFG

Watch

0

Star

0

Fork

0

<> Code

Issues 0

Pull requests 0

Projects 0

Wiki

Insights

Branch: master

SCFG / docs / especificacao /

Create new file

Upload files

Find file

History

ferreiracaf Estruturação do repositório

Latest commit 1ab29de an hour ago

..

arquitetura

Estruturação do repositório

an hour ago

prototipo

Estruturação do repositório

an hour ago

requisitos

Estruturação do repositório

an hour ago

RoberttyFreitas / SCFG

Watch

0

Star

0

Fork

0

<> Code

Issues 0

Pull requests 0

Projects 0

Wiki

Insights

Branch: master

SCFG / docs / gerenciamento /

Create new file

Upload files

Find file

History

ferreiracaf Estruturação do repositório

Latest commit 1ab29de an hour ago

..

aceites

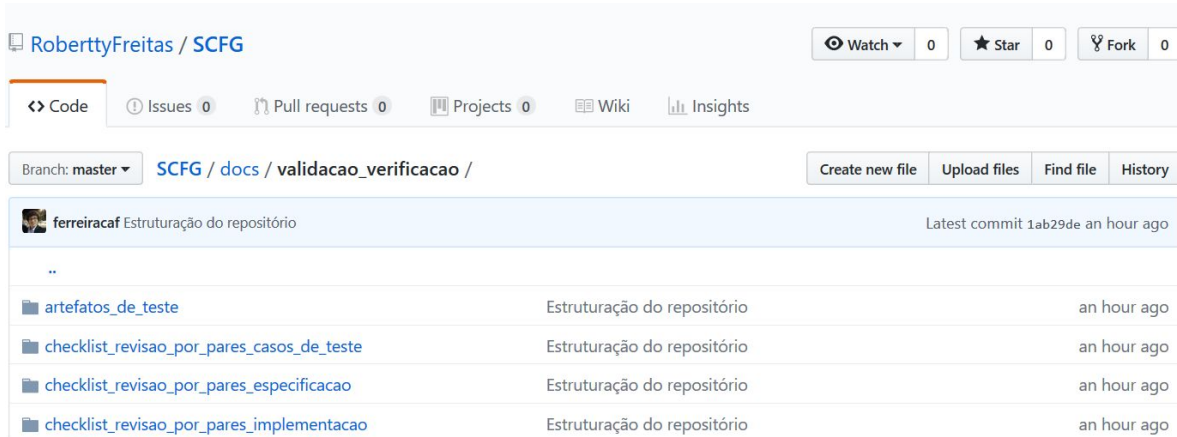
Estruturação do repositório

an hour ago

planilhas_fpa

Estruturação do repositório

an hour ago



3.2 Política de Segurança

As políticas de segurança definida para as pastas do repositório serão definidas conforme tabela a seguir:

Branches – GitHub		
Nome/Localização da Pasta	Membro da Equipe	Permissão
Todas	Gerente de Desenvolvimento (GD)	L, S, E, X
	Líder de Projeto (LP)	L, S, E, X
	Responsável pela Configuração (RC)	L, S, E, X
	Outros Membros da Equipe	L, S, E

Tabela 2 - Permissões diretório Branches GitHub

Tags – GitHub		
Nome/Localização da Pasta	Membro da Equipe	Permissão
Todas	Gerente de Desenvolvimento (GD)	L, S, E, X
	Líder de Projeto (LP)	L, S, E, X
	Responsável pela Configuração (RC)	L, S, E, X
	Outros Membros da Equipe	L, S, E

Tabela 3 - Permissões diretório Tags GitHub

Trunk – GitHub		
Nome/Localização da Pasta	Membro da Equipe	Permissão
Todas	Gerente de Desenvolvimento (GD)	L, S, E, X
	Líder de Projeto (LP)	L, S, E, X
	Responsável pela Configuração (RC)	L, S, E, X
	Outros Membros da Equipe	L, S, E

Tabela 4 - Permissões diretório Trunk GitHub

Legenda: L – Ler
 S – Salvar
 E – Editar
 X – Excluir

4 Métodos de Identificação

4.1 Documentos

Todos os documentos disponibilizados no repositório devem ser identificados baseados na seguinte nomenclatura:

<ID_REQUISITO_FUNCIONAL> - <DESCRIÇÃO>

Onde:

- <ID_REQUISITO_FUNCIONAL> é a sigla de identificação do requisito funcional implementado. Todos os tipos tipos de siglas estão descritas na tabela 6.
- <DESCRIÇÃO> é uma pequena descrição do que foi feito / alterado.

ID_REQUISITO_FUNCIONAL	DESCRIÇÃO DO REQUISITO
RF01	Realização de cadastro de uma nova galinha.
RF02	Realização do descadastro de uma galinha.
RF03	Realização do controle de quantidade de galinhas.
RF04	Realização de consulta sobre dados gerais do galinheiro no painel físico acoplado a porta do galinheiro.
RF05	Realização de consulta sobre dados gerais do galinheiro via um aplicativo móvel.
RF06	Realização de consulta sobre dados de uma galinha específica no painel físico acoplado a porta do galinheiro.
RF07	Realização de consulta sobre dados de uma galinha específica via um aplicativo móvel.

Tabela 6 - Identificadores dos requisitos funcionais do sistema

4.2 Configuração Base

As configurações bases definidas ao longo do projeto deverá ser utilizada a seguinte regra para a nomenclatura:

<VERSAO_MANUAL> - < DD-MM-AAAA >

Onde:

- <DD-MM -AAAA > é a data de criação da configuração base.
- <VERSAO_MANUAL> é o número da versão realizada conforme o padrão XX.YY.ZZ. Em que X é um número decimal que representa alterações de grande porte, como mudanças na arquitetura, grande mudanças na interface, mudanças na estratégia de marketing ou quando se é introduzida alterações incompatível na API. O Y indica adições que complementam funções já existentes. O Z indica o número atual da correção para a versão já entregue.

4.3 Documentos de Análise / Projeto de Software

Para cada caso de uso do sistema será definida um Documento de Análise de Software que será nomeado conforme regra a seguir:

<ID_CASO_USO> - <NOME_CASO_USO>

Onde:

- <ID_CASO_USO> é a identificação do caso de uso conforme definido no EOR – Especificação de Objetivos e Requisitos.
- < NOME_CASO_USO > é o nome caso de uso conforme definido no EOR – Especificação de Objetivos e Requisitos.

5 Controle de Mudanças

São definidos dois processos de controle de mudanças:

- a) **Procedimento Formal:** este processo deve ser implementado quando a mudança proposta afeta a última versão configuração já aprovada e estabelecida. A **Figura 1** ilustra o processo formal de mudança.

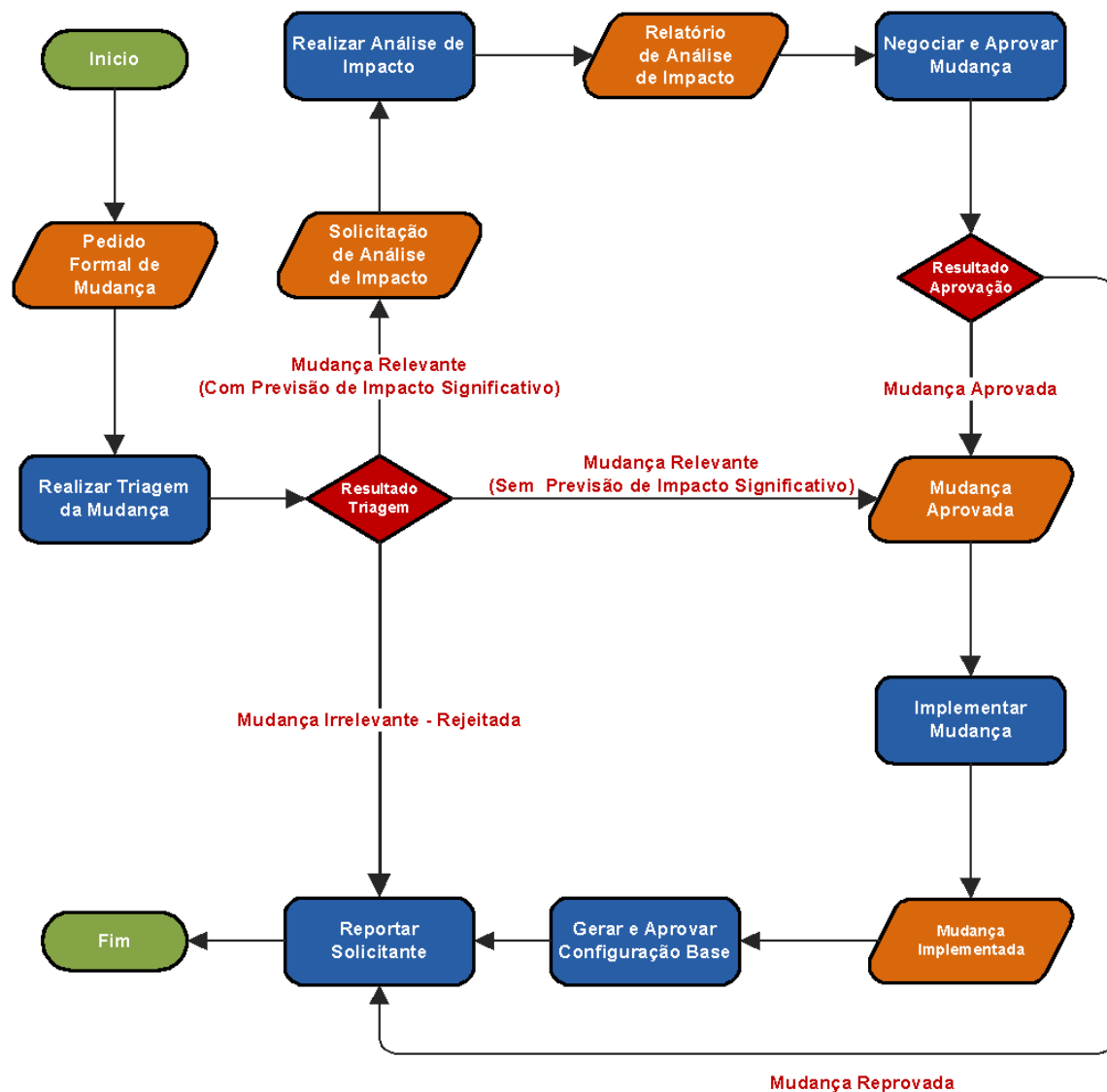


Figura 1 - Processo Formal de Mudança de Itens de Configuração

1. **Pedido Formal de Mudança:** O solicitante envia o Pedido de Mudança para o Gerente de Desenvolvimento para ser analisado. Em cada Pedido de Mudança devem estar destacadas as mudanças a serem avaliadas.
2. **Realizar Triagem de Mudança:** O GD com apoio do Líder de Projeto realiza a triagem definindo a relevância e possível impacto da mudança. O resultado da triagem pode gerar uma das seguintes decisões: Relevante Com Previsão de Impacto Significativo, Relevante Sem Previsão de Impacto Significativo e Mudança Irrelevante - Rejeitada.
3. **Solicitação de Análise Detalhada de Impacto:** As mudanças que foram classificadas como Relevante com previsão de impacto significativo devem passar por uma nova análise agora em conjunto com o Responsável pela Configuração.

4. **Realizar Análise de Impacto:** Estudo e análise de quais itens de configuração a mudança impactará.
5. **Relatório de Análise de Impacto:** Relatório simples contendo itens de configuração e o impacto sofrido pelo mesmo com a implementação da mudança. Além disso apresenta a estimativa de esforço, de tempo e, se aplicável, de custo.
6. **Negociar e Aprovar Mudanças:** O GD juntamente com o cliente (ou solicitante) negocia e aprova ou reprovava a mudança a implementação da mudança.
7. **Mudança Aprovada:** Relatório simples contendo as alterações aprovadas que serão implementadas.
8. **Implementar Mudança:** O GD envia a mudança aprovada para o RC, que providenciará a implementação da mudança (distribuindo as tarefas entre os componentes da equipe).
9. **Mudança Implementada:** O RC, com o resultado da mudança implementada, gera uma nova versão da configuração.
10. **Gerar e Aprovar Configuração Base:** Gera uma nova versão com as mudança totalmente implementadas.
11. **Reportar Solicitante:** A GD envia ao solicitante se mudança foi reprovada ou considerada irrelevante.

b) Procedimento Informal: O processo informal deve ser implementado nos casos em que a(s) mudança(s) proposta(s) não afetarem os itens de configuração na última versão da configuração, não afetando o desempenho geral da configuração. Neste caso, será feita apenas uma análise simples pelo GD e o LP para aprovação ou não da mudança.

6 Ambiente e Ferramentas

6.1 Plano de Software

Software	Propósito	Ambiente	Release/Versão
GitHub	Controle de Repositório	Desenvolvimento	-
OpenProj	Gerenciamento de Projeto	Gerência de Projeto	1.4
Atmel Studio	IDE - Atmega 328P	Desenvolvimento	7
App Inventor	Aplicações Android	Desenvolvimento	2
Autodesk Eagle	Design de PCB	Desenvolvimento	8.7.1
Proteus	Simulação	Desenvolvimento	8
ShareLatex	Documentação	Desenvolvimento	-

Tabela 7 - Plano de Software

7 Versão de Entrega do Produto

Os itens contidos na versão de entrega do produto serão:

- Antena de identificação RFID;
- Aplicativo móvel;
- Código fonte;
- Documentos de casos de uso;
- Elicitação de Requisitos;
- Painel físico;
- Planejamento realizado na ferramenta OpenProj;
- Plano de Gerência de Configuração.

RELATÓRIO DA EQUIPE

Nesta última seção, segue a porcentagem de esforço de cada membro da equipe.

Tabela 1. Porcentagem de esforço dos membros da equipe.

Nome	Esforço da Equipe
Carlos Alberto	20%
Francisca Beatriz	20%
Gabriel Uchôa	20%
Johnny Marcos	20%
José Robertty	20%