

Arduino Weather Station	Versão: 1.0
Plano de Gerenciamento de Configuração	Data: 19/08/13
AWS_SCPM_201308	

**Arduino Weather Station**  
**Plano de Gerenciamento de Configuração**  
**Versão 1.0**

Arduino Weather Station	Versão: 1.0
Plano de Gerenciamento de Configuração	Data: 19/08/13
AWS_SCPM_201308	

## Histórico da Revisão

Data	Versão	Descrição	Autor
19/08/13	1.0	Criação inicial	Jefferson Ribeiro Uchôas

Arduino Weather Station	Versão: 1.0
Plano de Gerenciamento de Configuração	Data: 19/08/13
AWS_SCPM_201308	

## Índice Analítico

1. Introdução .....	4
2. Descrição do Projeto.....	4
3. Processo.....	4
4. Finalidade .....	4
5. Escopo .....	4
6. Definições, Acrônimos e Abreviações.....	5
7. Referências .....	5
8. Itens de configuração .....	5
8.1. Configuração do software – Ferramentas do ambiente de desenvolvimento.....	5
8.2. Configuração das máquinas dos ambientes .....	6
8.3. Papéis .....	7
9. Controle de Configuração e Mudança .....	8
9.1. Processamento e Aprovação de Solicitações de Mudança .....	8
9.1.1. <b>Status do Projeto</b> .....	8
9.1.2. <b>Diagrama de Status das Tarefas do Projeto</b> .....	8
9.2. <b>Software de Gerenciamento de Mudança</b> .....	9
9.2.1. <b>Estrutura de pastas</b> .....	9
9.3. <b>Processo de Armazenamento e Liberação do Projeto</b> .....	10
9.4. <b>Liberação de release</b> .....	10
9.5. <b>Gerenciamento de responsabilidades das tarefas (Issues)</b> .....	10
9.5.1. <b>Ferramenta Issue Track</b> .....	11
9.6. <b>Commit</b> .....	11
10. <b>Testes</b> .....	12
11. <b>Baselines do Projeto</b> .....	12

Arduino Weather Station	Versão: 1.0
Plano de Gerenciamento de Configuração	Data: 19/08/13
AWS_SCPM_201308	

## Plano de Gerenciamento de Configuração

### 1. Introdução

Este Plano de Gerenciamento de Configuração estabelece e mantém a integridade de códigos-fonte e demais produtos do projeto Arduino Weather Station, permitindo o acompanhamento destes itens durante todo o ciclo de vida do projeto, preservando o histórico de evolução do sistema, auxiliando a gerenciar o estado dos itens de configuração dos sistemas, controlar as mudanças em itens de configuração e rastrear modificações nos itens de configuração ao longo do tempo.

### 2. Descrição do Projeto

O projeto Arduino Weather Station irá utilizar a Internet das Coisas para o desenvolvimento de um sistema automático e de baixo custo para medição de índices meteorológicos. [Inserir link para o documento de visão.](#)

### 3. Processo

Para o desenvolvimento do projeto Arduino Weather Station será adotado o processo SCRUM.

### 4. Finalidade

Este documento descreve a organização, nomenclatura e regras para o versionamento do projeto Arduino Weather Station, com o intuito de garantir o maior controle do produto no decorrer do projeto. Serão detalhados os recursos necessários (equipes, ferramentas e computadores), as responsabilidades atribuídas e o cronograma de atividades.

### 5. Escopo

Este Plano de Gerenciamento de Configuração é destinado para os integrantes da equipe responsável pelo desenvolvimento do projeto Arduino Weather Station e abrange todo o controle e gerenciamento da configuração do projeto.

Arduino Weather Station	Versão: 1.0
Plano de Gerenciamento de Configuração	Data: 19/08/13
AWS_SCPM_201308	

## 6. Definições, Acrônimos e Abreviações

Termo	Significado
SCRUM	É um processo ágil que permite manter o foco na entrega do maior valor de negócio, no menor tempo possível.
GC	Gerência de Configuração

## 7. Referências

## 8. Itens de configuração

### 8.1. Configuração do software – Ferramentas do ambiente de desenvolvimento

Tipo	Ferramenta	Versão
Sistema Operacional	Windows 8	Pro
Planilha	Microsoft Office Excel	2010
Editor de Texto	Microsoft Office Word	2010
Antivírus	Windows Defender	
Controle de Versão (Plug-in Eclipse)	Jboss Tools - Git	4.0.0
Plataforma de Desenvolvimento Arduino	Ferramenta: Arduino PDE	1.0.2
	Linguagem: Processing	1.0
Plataforma de Desenvolvimento Web	Ferramenta: Eclipse	Juno
	Linguagem: Java	
	JDK	1.7.0_17
	Java EE - Jboss Javaee	6.0
Banco de Dados	MongoDB	2.4.6
Driver MongoDB	Mongo Java Driver	2.10.1
Plataforma de hospedagem – PaaS	Openshift	
Gerenciamento de repositórios	Maven	2.1.1
API para o Web Service	JBoss jaxrs	2.1.0
Plug-in Jersey	Jersey Server	1.9

Arduino Weather Station	Versão: 1.0
Plano de Gerenciamento de Configuração	Data: 19/08/13
AWS_SCPM_201308	

## 8.2. Configuração das máquinas dos ambientes

QTD	Ambientes	Configuração Hardware	Configuração Software
1	Desenvolvedor	Processador: 2.4 GHz Memória RAM: 8GB Hard Disk: 320 GB	Windows 8 Professional
			Arduino PDE 1.0.2
			Eclipse Juno
			JDK 1.7.0_17
			Plugin Git
			Plugin Maven 2.1.1
			API JBoss jaxrs 2.1.0
			Mongo Java Driver 2.10.1
			Java EE - Jboss Javaee 6.0
			Jersey Server 1.9
			Windows Defender
1	Integração	Processador: 2.4 GHz Memória RAM: 8GB Hard Disk: 320 GB	Office 2010
			Windows 8 Professional
			Arduino PDE 1.0.2
			Eclipse Juno
			JDK 1.7.0_17
			Plugin Git
			Plugin Maven 2.1.1
			API JBoss jaxrs 2.1.0
			Mongo Java Driver 2.10.1
			Java EE - Jboss Javaee 6.0
			Jersey Server 1.9
			Windows Defender
			Office 2010

Arduino Weather Station	Versão: 1.0
Plano de Gerenciamento de Configuração	Data: 19/08/13
AWS_SCPM_201308	

### 8.3. Papéis

Papéis	Ferramenta	Responsabilidade	Colaborador
Product Owner	Issue Track	Estabelecer Políticas de GC Escrever Plano de GC Configurar Ambiente de GC Criar Espaços de Trabalho de Integração	Jefferson Uchôas
Scrum Master		Estabelecer Processo de Controle de Mudanças Revisar Solicitação de Mudança	Jefferson Uchôas
Desenvolvedor Web	Eclipse JDK 1.7.0_17 Plugin Git Plugin Maven API JBoss jaxrs Java EE Jersey Server	Construir a interface web e o web service	Jefferson Uchôas
Desenvolvedor Arduino	Arduino PDE Processing	Construir o software de controle do hardware	Jefferson Uchôas
Construtor do Hardware	Eletrônica Sensores	Construir o hardware do projeto	Jefferson Uchôas
Analista de banco de dados	MongoDB Eclipse JDK 1.7.0_17	Criar a lógica de inserção e consultas ao banco de dados	Jefferson Uchôas

Arduino Weather Station	Versão: 1.0
Plano de Gerenciamento de Configuração	Data: 19/08/13
AWS_SCPM_201308	

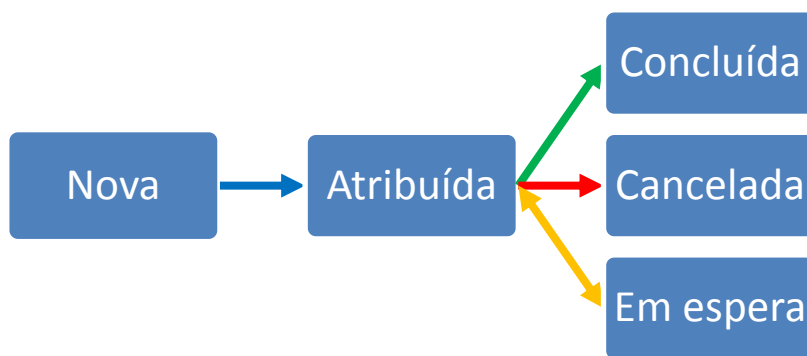
## 9. Controle de Configuração e Mudança

### 9.1. Processamento e Aprovação de Solicitações de Mudança

#### 9.1.1. Status do Projeto

Status	Descrição	Tarefa	Responsabilidade
Nova	Solicitação aguardando atribuição	Desenvolvimento	
		Teste	
		Documentação	
Atribuída	Solicitação sendo desenvolvida	Desenvolvimento	
		Teste	
		Documentação	
Concluída	Solicitação finalizada	Desenvolvimento	
		Teste	
		Documentação	
Cancelada	Solicitação inviável	Desenvolvimento	
		Teste	
		Documentação	
Em espera	Dependência imprevista	Desenvolvimento	
		Teste	
		Documentação	

#### 9.1.2. Diagrama de Status das Tarefas do Projeto



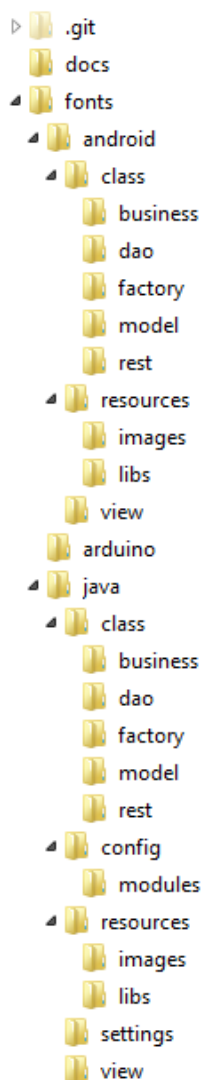


Arduino Weather Station	Versão: 1.0
Plano de Gerenciamento de Configuração	Data: 19/08/13
AWS_SCPM_201308	

## 9.2. Software de Gerenciamento de Mudança

As mudanças serão controladas através do GIT, as quis serão armazenadas no seguinte repositório: <https://github.com/jefuchoas/AWS>.

### 9.2.1. Estrutura de pastas



Arduino Weather Station	Versão: 1.0
Plano de Gerenciamento de Configuração	Data: 19/08/13
AWS_SCPM_201308	

Diretório	Descrição
fonts	Contém diretórios para o código fonte, bibliotecas utilizadas e scripts de compilação.
Android	
Class	Classes do projeto
Resources	Bibliotecas externas utilizadas pelo projeto, imagens
View	Telas e layouts
Java	
Class	Classes do projeto
Config	Arquivos de configuração do projeto
Resources	Bibliotecas externas utilizadas pelo projeto, imagens
Settings	Arquivos de configuração do ambiente
View	Telas e layouts
docs	A documentação para o projeto está nas sub-pastas de docs
test	Documentos de teste para o projeto, incluindo documentos de procedimentos de testes, planos de teste, etc.

### 9.3. Processo de Armazenamento e Liberação do Projeto

O backup do repositório deverá ser feito toda semana pelo Scrum Master. Os artefatos alterados durante a semana de trabalho será armazenado em mídia de CD, sendo um para cada mês.

Os CDs deverão conter a seguinte descrição: a data e hora e a periodicidade.

### 9.4. Liberação de release

Basicamente os projetos irão ser desenvolvidos e testados na linha principal (main-line). Para gerar o release, a versão em questão tem que estar devidamente testada, livre de erro e aprovado pelo analista responsável.

### 9.5. Gerenciamento de responsabilidades das tarefas (Issues)

O gerenciamento da responsabilidade das tarefas será realizado com a ferramenta do próprio GIT (Issue Track). Esta função será exercida por todos os integrantes do time, os

Arduino Weather Station	Versão: 1.0
Plano de Gerenciamento de Configuração	Data: 19/08/13
AWS_SCPM_201308	

quais serão responsáveis pela divisão e cadastro de uma nova tarefa, pela atribuição a um membro da equipe e, caso seja necessário, pela sua reabertura.

Para toda nova tarefa cadastrada e posteriormente atribuída o sistema dispara uma mensagem de e-mail para o responsável pela tarefa. As labels do Issue Track corresponderão aos Status do Projeto.

Ao ser aberta uma nova tarefa, sua descrição deverá conter no mínimo o nome da Classe, nome do método a ser criado ou alterado e uma descrição do problema a ser resolvido, de modo que todo integrante da equipe possa entender perfeitamente o que deve ser feito durante a execução da tarefa.

### 9.5.1. Ferramenta Issue Track

The screenshot displays the Hitechdv Issue Track interface. At the top, there are tabs for 'Browse Issues' and 'Milestones', and a 'New Issue' button. Below the tabs, there are filters for 'Everyone's Issues' (1), 'Assigned to you' (1), 'Created by you' (1), and 'Mentioning you' (0). A summary bar shows '1 Open' and '3 Closed' issues, with a 'Sort: Newest' dropdown. The main list shows an issue titled 'Alteração da classe WeatherStationActivity para a plataforma Android' with a purple 'Atribuida' label and status '#4'. The issue was opened by 'jefuchoas' 9 minutes ago. Below the issue list, there is a 'Labels' section with a list of labels and their counts: Atribuida (1), Cancelada (0), Concluida (0), Em espera (0), and Nova (0).

### 9.6. Commit

As tarefas apenas poderão ser enviadas (commit) para o repositório quando devidamente desenvolvidas e encerradas. Entende-se por desenvolvida a tarefa devidamente funcional, testada e livre de erro. **Pensar em monitoramento dos commits.**

Arduino Weather Station	Versão: 1.0
Plano de Gerenciamento de Configuração	Data: 19/08/13
AWS_SCPM_201308	

## 10. Testes

Os testes aplicados serão do tipo Teste de Unidade e para tal será utilizada a ferramenta JUnit. O responsável pelo teste será o próprio desenvolvedor da tarefa, o qual deverá entregá-la livre de erros.

As classes de teste serão mantidas em pacotes idênticos aos pacotes das classes de Release.

## 11. Baselines do Projeto

Baseline é o conjunto de artefatos que recebe uma aprovação de estabilidade. Uma baseline é usada como uma base no desenvolvimento das próximas fases dos artefatos e tem suas modificações controladas por um processo.

As baselines serão definidas em três fases, a saber:

Fases	Itens de Configuração da Baseline
Planejamento	Documentação (Artefatos do projeto)
Arquitetura do projeto	Código fonte contendo apenas a arquitetura do Projeto. <ul style="list-style-type: none"><li>- Camadas</li><li>- Garantia transacional</li></ul>
Release	Fontes do sistema pronto

Os Artefatos entrarão em baseline ao final de um Sprint.

## 12. Nomenclatura de versões

## 13. Política de Merge e Branch