

## **Plano de Gerência de Configuração e Mudanças**



## **Sumário**

<b>1. Objetivo do Plano de Configuração</b>	<b>4</b>
<b>2. Disposições Gerais</b>	<b>4</b>
<b>3. Processo de Gerência de Configuração</b>	<b>4</b>
<b>4. Definições, Acrônimos e Abreviações</b>	<b>5</b>
<b>5. Ferramentas</b>	<b>5</b>
<b>6. Gerenciamento da configuração</b>	<b>6</b>
6.1. Armazenamento dos Itens de Configuração	6
6.2. Artefatos de Trabalho e Itens de Configuração.	6
6.3. Baselines de Projeto	6
6.3.1. Itens de Configuração da Baseline	7
6.4. Branches	7
<b>6. Padrões</b>	<b>7</b>
6.1. Identificadores	7
6.2. Controle de Versão	7
<b>7. Controle da configuração</b>	<b>9</b>
7.1. Solicitação de mudança (CR)	9
7.2. Ciclo de vida da CR	10

## 1. Objetivo do Plano de Configuração

O objetivo deste documento é identificar e descrever as políticas e métodos para as atividades de Gerência de Configuração (GC), as quais serão utilizadas no ciclo de vida dos sistemas desenvolvidos pela Athena Key Solutions. Este documento será atualizado de acordo com o andamento e as necessidades dos projetos.

## 2. Disposições Gerais

O objetivo deste documento é registrar as decisões pertinentes à gerência de configuração. Deve-se então identificar, manter a integridade e gerenciar os itens de configuração dos projetos desenvolvidos pela Athena Key Solutions, como também de expor os princípios e preceitos a serem seguidas pelos seus colaboradores, tais como ferramentas, regras de colaboração e produção, infraestrutura e nomenclatura. Só é permitida a visualização deste documento pelas pessoas que sejam colaboradoras da Athena Key Solutions.

## 3. Definições, Acrônimos e Abreviações

Segue abaixo tabela com as abreviações, acrônimos e definições que serão utilizadas durante todo este documento:

IC - Item de Configuração	Qualquer artefato produzido durante o ciclo de vida do projeto cujas mudanças serão controladas através de um processo formal.
Baseline	Marco de referência utilizado para o desenvolvimento de software, pode ser utilizada para teste ou até mesmo para identificar uma entrega.
Iteração	Utilizada para identificar uma mudança ou melhoria em um software.
Release	É uma entrega planejada de uma parte de um projeto, essa parte deve ser funcional.
Git	É um sistema para controle de versão, com um conjunto de comandos que provém operação de alto nível.
Branch	É uma nomenclatura utilizada para identificar um ramo em um repositório git
Sprint	Nomenclatura utilizada para identificar ciclos de trabalho na metodologia ágil adotada pela Athena Key Solution Tecnologia, Scrum.

#### 4. Ferramentas

A tabela abaixo lista as ferramentas, tecnologias e outros recursos necessários para correta execução das atividades da Athena Key Solutions:

Ferramenta	Versão	Objetivo
<a href="#">Google Drive</a>	Atual	Repositório para documentos utilizados por projetos, departamento administrativo-financeiro, gestão de pessoas, comercial, marketing, presidência e para gerenciamento de conhecimento.
<a href="#">Google Docs</a>	Atual	Elaboração de documentos.
<a href="#">Google Slides</a>	Atual	Elaboração de apresentações ou qualquer artefato que precise de slides.
Draw.io	Atual	Modelagem UML.
Git	Estável	Ferramenta de controle de versão.
<a href="#">Github</a>	Atual	Ferramenta remota que utiliza o serviço de controle de versão.
Discord	Atual	Ferramenta para comunicação de todos os membros da empresa.
EERCase	Estável	Ferramenta utilizada para a modelagem do modelo entidade-relacionamento.
Visual Studio Code	Atual	IDE utilizada para desenvolvimento de aplicações web.
Angular(*)	Estável	Tecnologia utilizada para desenvolvimento de sistemas Web.
React(*)	Estável	Tecnologia utilizada para desenvolvimento de sistemas Web.
React Native(*)	Estável	Tecnologia utilizada para desenvolvimento de aplicativos móveis híbridos.
Node.js(*)	Estável	Tecnologia utilizada para criação de backend para sistemas web/mobile.
MySQL(*)	Estável	SGBD utilizado em todas as aplicações que necessitam de um sistema de banco de dados relacional.

(\*) A Escolha dessa tecnologia será realizada baseada nos requisitos levantados pela equipe de desenvolvimento.

## 5. Gerenciamento da configuração

### 5.1. Armazenamento dos Itens de Configuração

O repositório para controle dos itens de configuração gerados no desenvolvimento de qualquer um dos projetos da Athena Key Solutions, deve utilizar a ferramenta para controle de versão padrão (definida na seção 5 deste documento) e seguir a estrutura de diretórios abaixo:

Diretório	Descrição
Código	Diretórios contendo os códigos de sistemas desenvolvidos pela Athena Key Solutions
Documentação	Diretórios contendo a documentação relativa aos projetos e todos os outros procedimentos da Athena Key Solutions
Produto	Documentos relativos ao sistema a ser desenvolvido.
Projeto	Documentos relativos ao projeto.
Artefatos	Documentos de suporte para o desenvolvimento do Projeto.
Templates	Templates utilizados.

### 5.2. Artefatos de Trabalho e Itens de Configuração.

Os artefatos que estão sobre controle da gerência de configuração, bem como sua localização no repositório estão relacionados na tabela a seguir <sup>1</sup>:

Artefato	Localização	Responsável
Código	<a href="https://github.com/athenakeysolutions">https://github.com/athenakeysolutions</a> Para todo novo projeto da Athena Key Solutions deve ser criado um novo repositório na organização acima.	Gerente de Projetos
Documentos de Definição de Projetos	Para todo novo projeto deverá ser criada uma pasta no repositório de artefatos da Athena Key Solutions, deve ser utilizado como nomenclatura da pasta o nome do cliente. Essa pasta deve ser criada dentro da pasta "PROJETOS".	Gerente de projetos

Vale ressaltar que não haverá controle de acesso nesses artefatos visando uma maior produtividade da equipe.

---

1

### 5.3. Baselines de Projeto

As *baselines* geradas para os projetos seguirão o padrão de nomenclatura descrito a seguir:

Baseline	Descrição	Padrão
Iteração	Criada assim que finalizada a fase de iteração do projeto.	<ITERACAO>-<VERSAO>
Finalização	Criada assim que finalizada a fase de finalização do projeto.	<FINALIZACAO>-<VERSAO>
Release	Criada para identificar o marco de uma entrega	<SPRINT>-REL

<VERSAO> - Número da versão que será atribuída a tal conteúdo, o padrão para código de versionamento será descrito na secção 6.2;

<SPRINT> - Código sequencial que identifica a sprint de entrega.

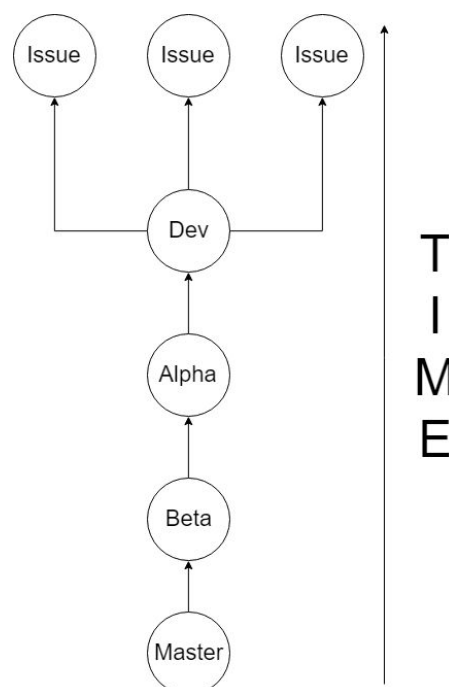
#### 5.3.1. Itens de Configuração da Baseline

Um artefato apenas poderá integrar a *baseline* depois da autorização do PO.

### 5.4. Branches

O projeto possuirá uma *Master*, na qual será salva a versão final após análise do gerente de projetos. Nenhum desenvolvedor que não seja o gerente de projetos deve enviar artefatos para esta branch.

Cada feature possuirá a sua Branch, seguindo o padrão "branch-<ID\_ISSUE>", cada desenvolvedor, durante o desenvolvimento, poderá apenas enviar o código de sua issue para a respectiva branch. Após finalizadas as modificações, o desenvolvedor deverá enviar as modificações realizadas para a branch utilizada para desenvolvimento "branch-dev".



## 5.5. Organização e Responsabilidades

Funções	Responsabilidades
Gerente de Projetos	Responsável por planejar e controlar a execução e o fluxo de projetos.
Engenheiro de Software	Responsável por aplicar as tecnologias e práticas de padrões de projetos, visando organização, produtividade e qualidade.
Analista de Qualidade	Responsável por aplicação de testes, análise e registro dos resultados.
Design	Responsável pela criação e edição em conteúdos relacionados a parte de interface gráfica.

## 6. Padrões

Esta seção descreve os padrões de identificação dos artefatos adotados para o desenvolvimento de qualquer projeto que esteja sendo desenvolvido pela Athena Key Solutions.

### 6.1. Identificadores

Com exceção dos arquivos de código do sistema, a nomenclatura de todos os itens de configuração deve possuir o seguinte formato: [AKS]-NOME\_DO\_ARTEFATO. A nomenclatura deve ser totalmente escrita em letras maiúsculas e se o nome do artefato possuir mais de uma palavra, elas devem ser separadas por um *underscore* (\_). Entre a sigla e nome do artefato, utiliza-se apenas um hífen, sem espaços.

### 6.2. Controle de Versão

As versões dos artefatos sob a gerência de configuração devem ser representadas no padrão "XX.YY[-D<ZZ>]", no qual:

- <XX> indica alterações significativas no artefato, ou seja, este campo deve ser incrementado a cada nova aprovação de documento. Inicia-se em "01";
- <YY> indica pequenas alterações no artefato. Inicia-se esse campo com "00" e quando <XX> for incrementado, este número deverá voltar para "00";
- [-D<ZZ>] Indica uma versão rascunho, ou seja, uma versão que ainda está sendo trabalhada e não deve ser considerada, pois sua estabilidade ainda não foi comprovadamente atestada. Este campo só deve ser utilizado quando uma nova versão do documento é produzida e não passou por revisão, ou seja, a versão está imprópria para o uso. Este campo inicia-se com o valor "-D01" e volta a este valor quando <XX> ou <YY> são modificados, ou seja, quando se

altera pela primeira vez um documento aprovado (em baseline). Fica a critério do Gerente de Projetos definir se este item será utilizado ou não.

As versões anteriores à 1.0 são consideradas rascunhos de um artefato, portanto, não serão controladas pela gerência de configuração.

## 7. Controle da configuração

### 7.1. Solicitação de mudança (CR)

As solicitações de mudança em requisitos dos projetos devem seguir os procedimentos apresentados no Plano de gerência de configuração e mudanças.

O controle das mudanças realizadas nos artefatos deve ser realizado através da seção “Histórico de Revisões”, apresentada em cada artefato. Esse histórico deve conter o nome do autor, descrição e data de realização da mudança.

Todas as alterações realizadas na *baseline* devem ser registradas através da [ferramenta de issues](#) do Github, e, apenas, por membros da fábrica. Cada CR (*issue*) contém os seguintes campos<sup>2</sup>:

Nome	Descrição
Template	Deve ser escolhida a opção “Solicitação de Mudança”.
Title	Breve descrição do problema.
Content	Descrição detalhada do problema, respondendo as perguntas: “Passos para reprodução”, “Qual saída esperada e qual é a saída real?” E qualquer outra informação que o solicitante achar importante.
Status	Detalhado na seção a seguir: “Ciclo de vida da CR”.
Assignee	Responsável pela resolução da CR.
CC	Co-responsável pela CR.
Labels	Esse campo refere-se ao tipo de CR e a sua prioridade. Os tipos de CR são: Defeito - CR de defeito; Melhoria - CR de melhoria. E as prioridades, de acordo com as descrições apresentadas na seção Atributos deste documento, são: Essencial; Importante; Desejável.



## 7.2. Ciclo de vida da CR

Durante o ciclo de vida da solicitação, a CR recebe os seguintes status:

Nome	Descrição
Nova	CR criada e ainda não atribuída.
Atribuída	CR com um responsável já designado.
Iniciada	CR após atribuída teve seu desenvolvimento iniciado.
Resolvida	Uma CR já definida como resolvida.
Verificada	Mudanças validadas pelo responsável pela abertura da CR.
Concluída	Merge realizado com a branch de desenvolvimento

O fluxo entre as fases pode ser visto na figura a seguir:



**Figura 2: Fluxo do Status da Solicitação.**