



Diretrizes de Gerência de Configuração

1. Introdução

Este documento registra as diretrizes de Gerência de Configuração do projeto do Sistema de Análise de Notícias.

2. Modelo de Ramificação

O modelo de ramificação a ser adotado para os repositórios de código será o *Git Flow*, devido ao nível de maturidade da equipe e diversidade de *branches* que oferece.

3. Gerenciamento de versões

3.1. Branches

O padrão de nomenclatura de *branches* no modelo será: *<tag da branch/nome-da-branch>*. O nome da *branch* deve seguir o *snake-case*. As *tags* padrões de *branch* para o projeto são:

<i>feature</i>	Ramificação para desenvolvimento de uma funcionalidade nova do sistema.
<i>bugfix</i>	Ramificação para resolução de bugs do sistema.
<i>release</i>	Ramificação que contém as versões estáveis do sistema.
<i>develop</i>	Ramificação que contém as versões em desenvolvimento do sistema.
<i>master</i>	Ramificação que contém as versões estáveis e implantáveis do sistema.

3.2. Commits

Estrutura do *commit*: *<*tag* primária>*: *<Mensagem do *commit*>*

As *issues* podem ser referenciadas dentro da mensagem de *commit*, de modo que se possa até fechá-las com a própria mensagem.

Exemplo: *fix*: Padronização de *unicodes close #29* e Apagamento dos *logs close #30*

Palavras chaves (fecham, editam *issues*): *close, closes, closed, fixes, fixed, resolve, resolves, resolved*.

3.3. Pull requests

O código será mergeado se, e somente se, aprovado por ao menos dois integrantes, preferencialmente que não estejam envolvidos diretamente com o desenvolvimento e produção.

4. Gerenciamento de mudanças

4.1. Issues

O título da *issue* deve ser sucinto e explicativo. Criar uma *issue* para cada questão/problema, de forma a modularizar adequadamente o trabalho, com uma descrição clara e precisa. Caso seja uma *issue* de *bug*, é obrigatório uma demonstração do erro na descrição da *issue*, seja por foto ou vídeo de execução.

É imprescindível a manutenção do *status* da *issue* atualizado.

nome da <i>tag</i>	descrição da <i>tag</i>	cor	nome da <i>tag</i>	descrição da <i>tag</i>	cor
<i>bug</i>	Algo não está funcionando.	vermelha	<i>help wanted</i>	Atenção extra é requerida.	verde
<i>documentation</i>	Melhorias e/ou adições na documentação.	azul	<i>invalid</i>	Não parece correto.	amarelo
<i>duplicate</i>	Essa <i>issue</i> ou <i>pull request</i> já existe.	cinza	<i>question</i>	É necessário informação adicional/complementar.	rosa
<i>enhancement</i>	Nova funcionalidade ou pedido de funcionalidade.	verde claro			

5. Plataforma de integração contínua e entrega contínua (CI/CD)

5.1. GitHub Actions

A ferramenta *GitHub Actions* será utilizada para a Integração e Entrega Contínua por ser completamente integrada ao *GitHub*, eliminando a necessidade de criar e configurar uma infraestrutura externa com outras ferramentas de CI e CD.

A ferramenta de automação também possui uma comunidade muito ativa que oferece milhares de ações pré-construídas, além de empresas como a *Microsoft* e o próprio *GitHub* que contribuem com seus repositórios. As *actions* são *open-source* e funcionam em qualquer plataforma, linguagem e nuvem. Além disso, também rodam no *Linux*, *macOS*, *Windows* e contêineres.

Fluxos de trabalho

São processos automatizados configuráveis que executam um ou mais trabalhos. São definidos por um arquivo YAML que deve estar no repositório no diretório `.github/workflows` e que serão executados quando acionados por um evento, manualmente ou de acordo com um cronograma previamente definido.

Um repositório pode ter fluxos de trabalho diferentes para:

- Testar *pull requests*.
- Implantar o aplicativo toda vez que uma nova versão for criada.
- Adicionar uma etiqueta toda vez que alguém abre uma nova *issue*.

Arquivo de fluxo de trabalho

nome(opcional)	Nome do fluxo de trabalho a ser exibido na aba “Ações” do repositório.
run-name	Nome das execuções do fluxo de trabalho.
on	Especifica o evento/gatilho para o fluxo de trabalho.
jobs	Agrupa todos os trabalhos executados no fluxo de trabalho.
runs on	Determina o executor para o trabalho.
steps	Agrupa todas as tarefas que são executadas no trabalho.
uses	Especifica qual ação será executada durante a etapa.
run	Instrui o trabalho a executar um comando no executor (pode ser mais de um).

Para incluir ações em um fluxo de trabalho basta adicioná-las como novas etapas dentro de *jobs*. Atentar-se às entradas que as ações podem exigir, que por sua vez devem ser definidas no fluxo de trabalho.

Para adicionar uma ação de um repositório diferente é necessário incluir a referência `{owner}/{repo}@{ref}` em *uses*, especificando também a versão da ação (pode ser uma tag, um SHA ou *branches*).

6. Nomenclatura de Código

Na presente seção do documento, definem-se diretrizes de nomenclatura de código a serem seguidas durante o desenvolvimento do projeto.

7. Histórico de versões do documento

A presente seção apresenta o histórico de versões desse documento.

Versão	Publicação	Autor(es)	Ações realizadas
1.0	26/04/2023	Lucas Henrique Alves Borth, Júlia Alves Corazza, Murilo Matias e Pedro Rodrigues Arantes	- Versão inicial das diretrizes de Gerência de Configuração do projeto.