

# **Documentação Técnica: Guia de CI/CD**

## 1. Resumo Executivo

Este guia descreve o processo de **CI/CD** (Integração e Entrega Contínua) do projeto feito para a Sauter University 2025 realizado na UFAPE, utilizando o **GitHub Actions**. O objetivo é automatizar a validação, testes e *deploy* de aplicações na nuvem (Google Cloud), garantindo que o processo seja seguro, eficiente e fácil de ser reproduzido por qualquer membro da equipe. A documentação cobre a arquitetura, o fluxo de trabalho, boas práticas e um guia de solução de problemas.

## 2. Prioridades

- **Validação de Código** : Garantir que todo código passe por checagens automáticas (tipagem, testes) antes de ser mesclado à *branch* principal.
- **Deploy Automatizado**: Publicar artefatos de código (Docker no artifacts) ou infraestrutura (Terraform) em ambientes de destino (Cloud) de forma automática após a aprovação.
- **Segurança**: Proteger credenciais e acessos, utilizando métodos como **WIF** (Workload Identity Federation) e **Segredos do GitHub Actions**.

## 3. Responsabilidades e Histórico

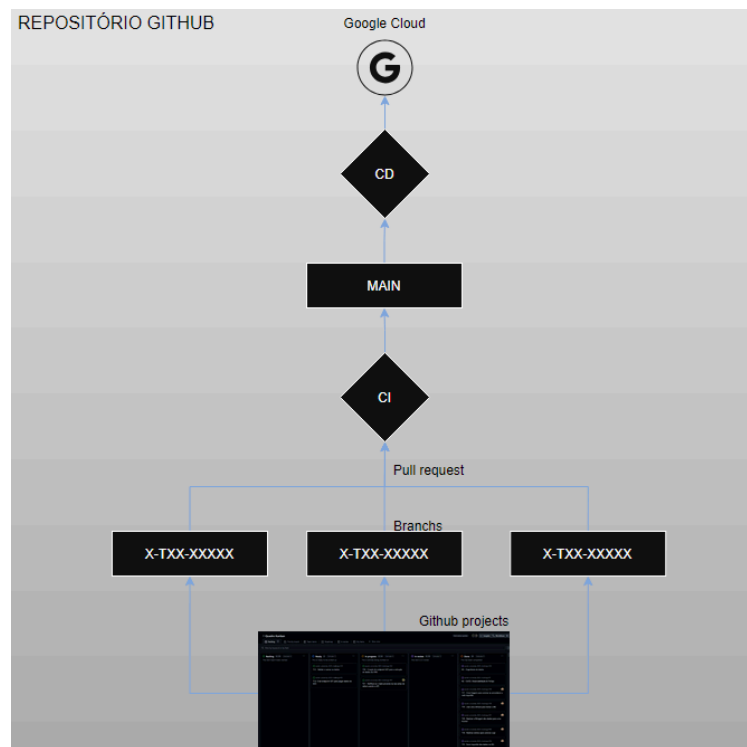
- **Responsável**: Clauderson Branco Xavier
- **Ações**: Criação do fluxo de CI/CD, incluindo os arquivos de *workflow* (*ci.yml*, *cd.yml*, *iac-dev.yml*) e a documentação inicial.
- **Data de Atualização**: 24/09/2025

## 4. Diagramas de Arquitetura e Fluxo de Dados

- **Fluxo de CI/CD**:
  1. O desenvolvedor abre um **Pull Request (PR)** para uma *branch* de *feature*.
  2. O *workflow* de **CI** (*ci.yml* e *iac-dev.yml*) é acionado automaticamente.
  3. O CI executa testes de código (mypy) e testes automatizados.
  4. Após a aprovação do PR e o *merge* para a *branch* *main*.

5. O terraform utiliza o plan para verificar se a estrutura está correta.
6. O *workflow* de **CD** (**cd.yml** e **iac-dev.yml**) é acionado.
7. O CD autêntico no ambiente de destino (usando WIF/Segredos).
8. O *deploy* do código e da infraestrutura é executado.

- **Diagrama de CI/CD:**



## 5. Guia de Implementação Passo a Passo

### 5.1 Pré-requisitos

- Conhecimento básico de **GitHub Actions** e **YAML**.
- Acesso ao repositório do projeto.
- Variáveis de ambiente (Segredos) e configurações de **WIF** devem estar configuradas no repositório para acesso a ambientes externos.

### 5.2 Configuração Local

- Clone o repositório: **git clone [URL\_DO\_REPOSITÓRIO]**
- Nenhuma outra configuração local é necessária para a execução dos *workflows*.

### 5.3 Como Rodar/Testar

- **CI:** Abra um Pull Request com seu código. O *workflow* `ci.yml` e `iac-dev.yml` será acionado automaticamente.
- **CD:** Faça o *merge* do seu PR para a *branch* `main`. O *workflow* `cd.yml` e `iac-dev.yml` será acionado e fará o *deploy*.

### 5.4 Validação em Produção

- Confira os logs do *workflow* de *deploy* no **GitHub Actions** para verificar se a execução foi bem-sucedida.
- Valide o comportamento da aplicação ou da infraestrutura no ambiente de produção.

### 5.5 Como Reverter

- Em caso de problema, utilize o histórico de *commits* para reverter a alteração: `git revert [HASH_DO_COMMIT]`. Isso acionará um novo *workflow* de CD para fazer o *rollback*.

## 6. Observabilidade e Logs

- **Onde ver logs:** Todos os logs de execução dos *workflows* estão disponíveis na aba "**Actions**" do repositório no GitHub.
- **Runbook para Incidentes Comuns:**
  - **Falha de Autenticação:** Verifique se as permissões da **Service Account** estão corretas e se o **WIF** foi configurado adequadamente.
  - **Erro de Sintaxe/Código:** O *workflow* de CI apontará o erro. Corrija o código e abra um novo PR.
  - **Deploy Parcial:** Revise o *workflow* de *deploy* (`cd.yml`) para garantir que todos os artefatos necessários estão sendo incluídos.

## 7. Segurança

- **Segredos:** Nunca *commite* credenciais ou informações sensíveis diretamente no código. Use os **Segredos do GitHub Actions** para armazená-los.

- **Permissões:** As permissões mínimas necessárias para o *deploy* devem ser configuradas nas **Service Accounts** na plataforma *cloud* de destino.

## 8. Registro de Decisões Arquiteturais

- **Decisão:** Utilizar GitHub Actions para CI/CD.
- **Justificativa:** GitHub Actions foi escolhido pela sua integração nativa com o GitHub, facilidade de uso via YAML e por ser uma solução robusta e escalável, ideal para equipes que já utilizam o ecossistema GitHub.

## 9. Changelog

- **v1.0.0 (24/09/2025):** Implementação inicial dos *workflows* de CI/CD ([ci.yml](#), [cd.yml](#), [iac-dev.yml](#)) e documentação.

## 10. Branch trabalhada

Abaixo segue alguns Pull requests de branches trabalhadas trabalhadas.

### [8-t6---github-actions-ci](#)



### [9-t7---cd-para-cloud-run](#)

