

---

**Integração e Entrega Contínua – DSM – Professora Lucineide – 1ºSemestre/2026****Aula 3 – Automação com GitHub Actions**

---

**Exercício 1 – Ativar GitHub Actions**

**Explicação:** Precisamos habilitar o “robô” que fará a integração contínua.

**Passo a passo:**

1. No GitHub, abra o repositório.
  2. Clique em **Actions**.
  3. Escolha **set up a workflow yourself**.
  4. Nomeie o arquivo como **ci.yml**.
- 

**Exercício 2 – Criar workflow inicial**

**Explicação:** Esse workflow básico garante que o Actions está funcionando.

**Passo a passo:**

1. No editor do GitHub, cole o seguinte código:

*# Nome do workflow (aparecerá na aba Actions)*

*name: CI Simples*

*# Eventos que disparam o workflow:*

*# - push (quando algo é commitado)*

*# - pull\_request (quando há PR)*

*on:*

*push:*

*branches: [ dev ]    # dispara apenas quando o commit é na branch dev*

*pull\_request:*

*branches: [ dev ]    # dispara quando há PR para dev*

*jobs:*

*# Definindo um job chamado "build"*

*build:*

*runs-on: ubuntu-latest    # ambiente que o GitHub usa (máquina Linux)*

*steps:*

---

Integração e Entrega Contínua – DSM – Professora Lucineide – 1ºSemestre/2026

# 1. Baixar o código do repositório

- name: Checkout do repositório

uses: actions/checkout@v4

# 2. Executar um comando simples (teste inicial)

- name: Teste de funcionamento

run: echo "CI funcionando no projeto INPE!"

2. Clique em **Commit changes** para salvar na branch **dev**.
3. Vá em **Actions** para confirmar que o workflow foi criado.

---

Exercício 3 – Testar o workflow

**Explicação:** Vamos ver se o workflow dispara de fato.

**Passo a passo:**

1. Edite o README.md.
2. Adicione no final:

*CI/CD configurado na Aula 3*

Faça commit na branch dev.

Vá em **Actions** -> clique no workflow em execução -> veja os **logs**.

No final, deve aparecer:

*CI funcionando no projeto INPE!*

---

Exercício 4 – Personalizar o workflow

**Explicação:** Agora vamos alterar a mensagem para ver que podemos customizar.

**Passo a passo:**

1. Abra .github/workflows/ci.yml.
2. Localize este trecho:

- name: Teste de funcionamento

run: echo "CI funcionando no projeto INPE!"

Substitua por:

- name: Teste de funcionamento

run: echo "Pipeline do projeto INPE executado com sucesso!"

Faça commit.

---

**Integração e Entrega Contínua – DSM – Professora Lucineide – 1ºSemestre/2026**

Veja em **Actions** a mensagem atualizada.

---

**Exercício 5 – Criar trigger para PR**

**Explicação:** Assim, o pipeline roda também em Pull Requests, antes do merge.

**Passo a passo:**

1. Já está configurado no YAML:

*pull\_request:*

*branches:* [ dev ]

1. Para testar:
    - Crie a branch feature/test-ci a partir de **dev**.
    - Altere qualquer arquivo (ex.: adicione um comentário no index.js).
    - Faça commit e push.
    - Abra PR feature/test-ci -> dev.
  2. Vá em **Actions** e confirme que o workflow disparou com o PR.
- 

**Entrega da Atividade**

1. Exporte/salve o arquivo de acordo com o padrão: **IEC-Atividade-Aula03.pdf**
2. Enviar a Atividade em PDF no GitHub:
  1. Acesse seu repositório da disciplina: **FATEC-JCR-2DSM-IEC-2026-1-seunome**
  2. Se ainda não existir, você deverá criar a pasta da atividade: **P1-Conteudos/Atividades/Atividade-Aula03/README.md**
  3. Faça o **commit** do pdf da atividade feita na pasta.
  4. Em seguida copie o **link** do pdf.
  5. Acesse o **Kanban-IEC-2026-1** da disciplina que se encontra no repositório da professora.
  6. Clique no card: **Atividade-Aula03**.
  7. No comentário, cole o **link** do **pdf** da atividade que você fez.
  8. Volte ao seu repositório e acesse o seu **Kanban-IEC-2026-1**.
  9. Mova o card **Atividade-Aula03** para a coluna **Entregue**.
3. Ou, no caso de configurações feitas no repositório do GitHub, enviar o link.