**Web Academy CI/CD**

**Laboratório - Aula 01**

**Objetivo do Laboratório**: Primeiro contato com o GitHub Actions através de exemplos de workflows simples

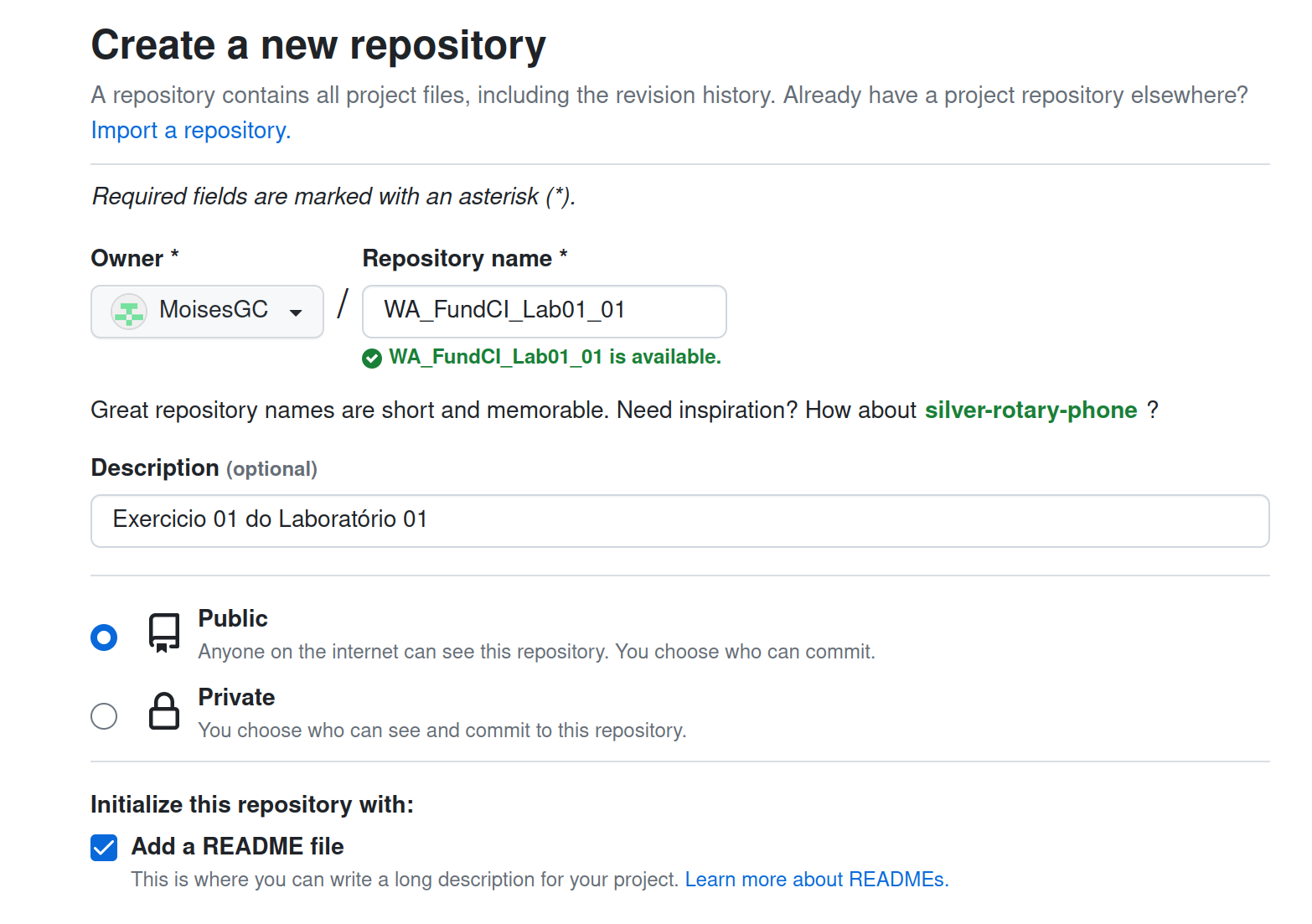
**Pré-Requisitos:**

* Ter uma conta no GitHub
* Máquina com Linux / Windows instalado e Git já configurado com a conta do GitHub
* Arquivos disponíveis no Colabweb para o Laboratório (caso necessário)

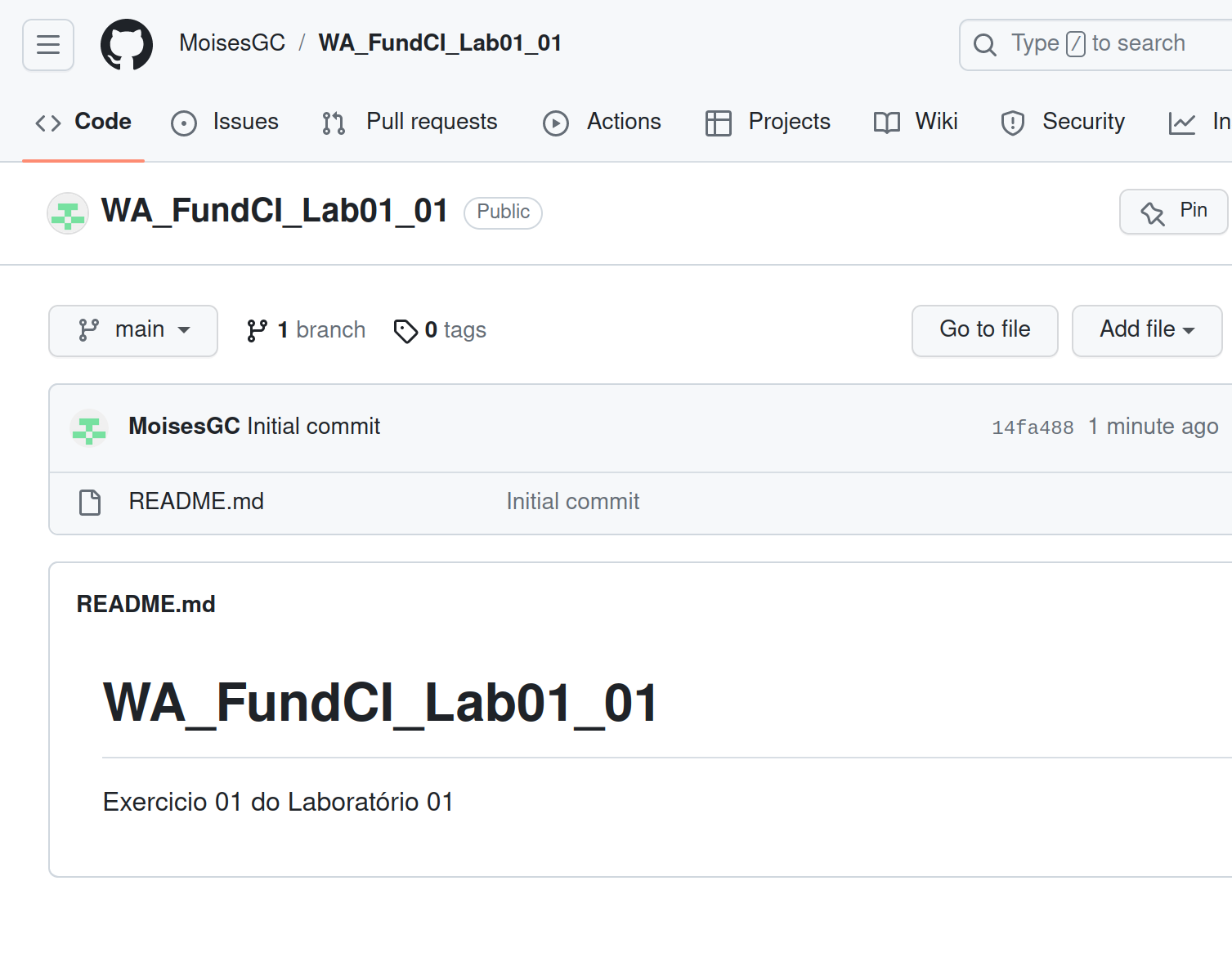
**Exercício 01 - Criando um Action de ativação manual**

1 - Crie um repositório (publico) na sua conta do GitHub

Durante a criação, permita a criação do arquivo README

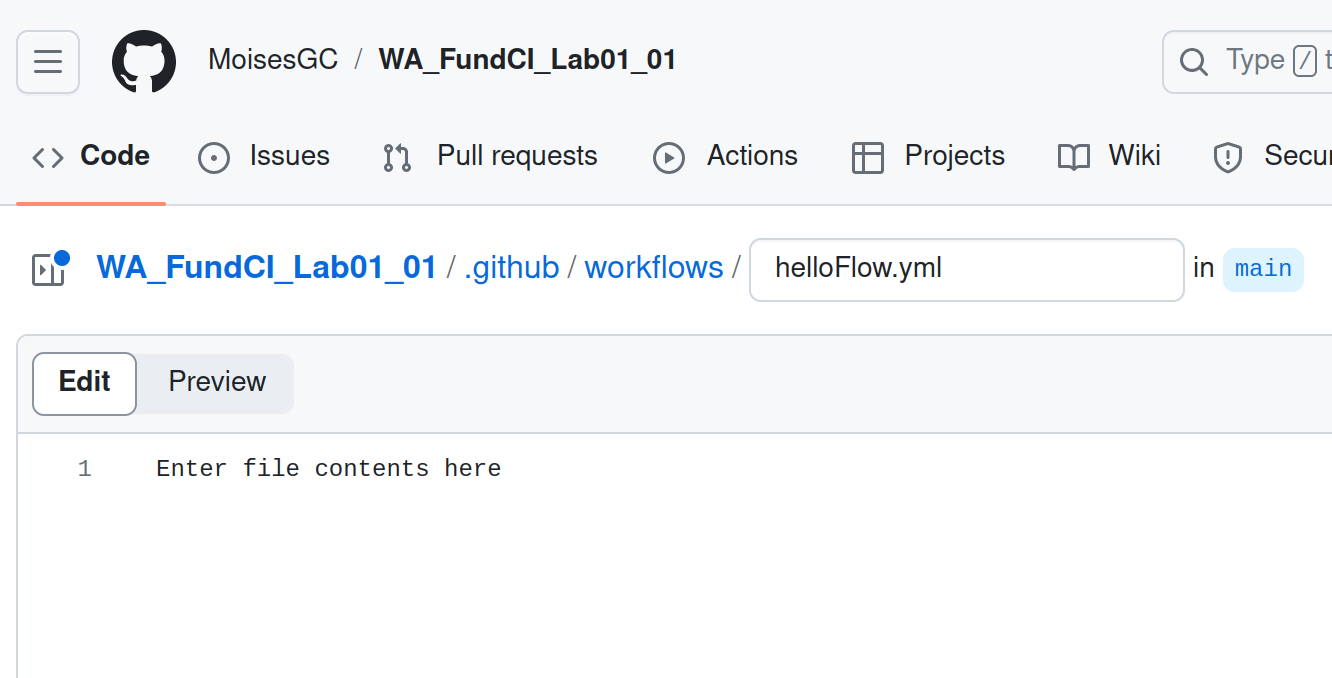


2- Depois de criado, temos a landing page do repositório



3 - Vá na tab Actions e clique no “Simple Workflow”

4 - Renomeie o seu primeiro workflow para “helloFlow.yml” e apague todo o conteúdo que foi gerado automaticamente pelo template.



5 - Copie o seguinte texto para o seu arquivo .yml e depois clique em “Commit Changes”

name: Hello Flow

on: workflow\_dispatch

jobs:

first-job:

runs-on: ubuntu-latest

steps:

- name: Diga oi

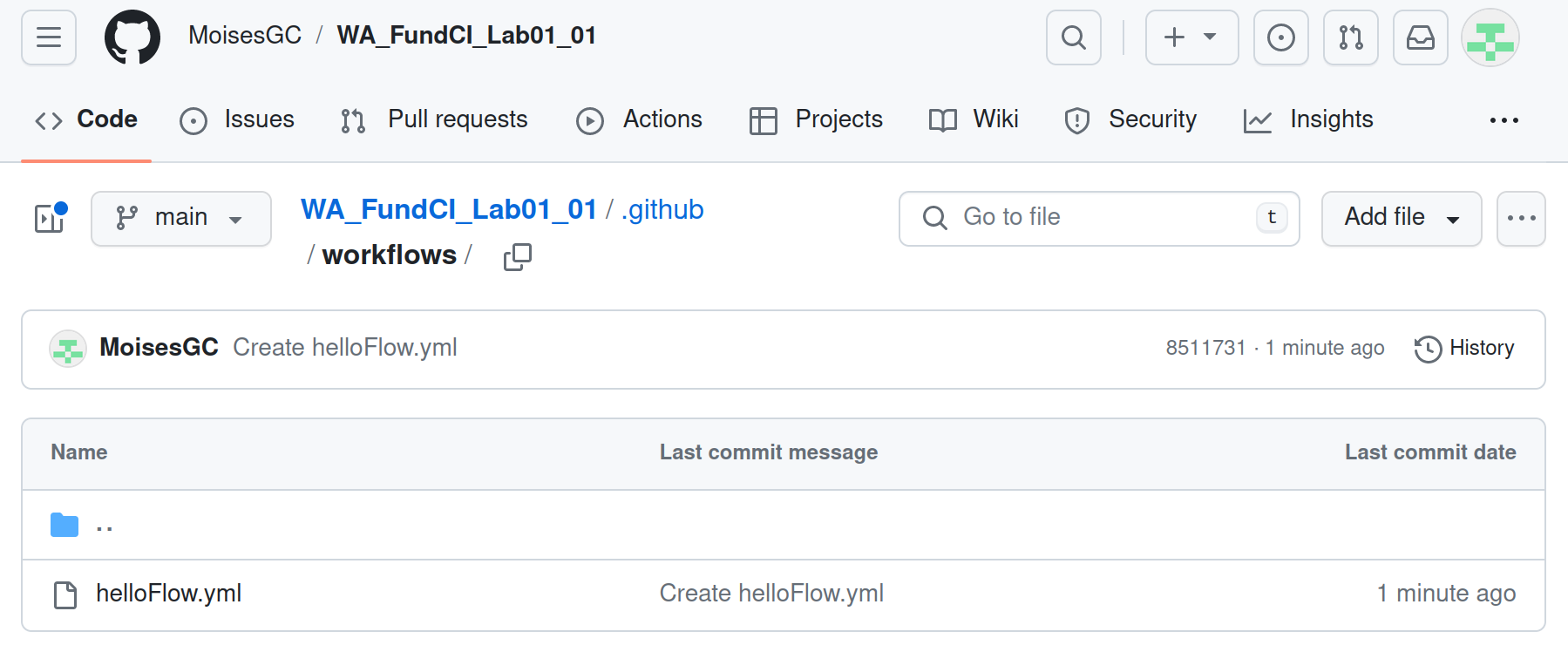
run: echo "Hello World!"

- name: Diga valeu

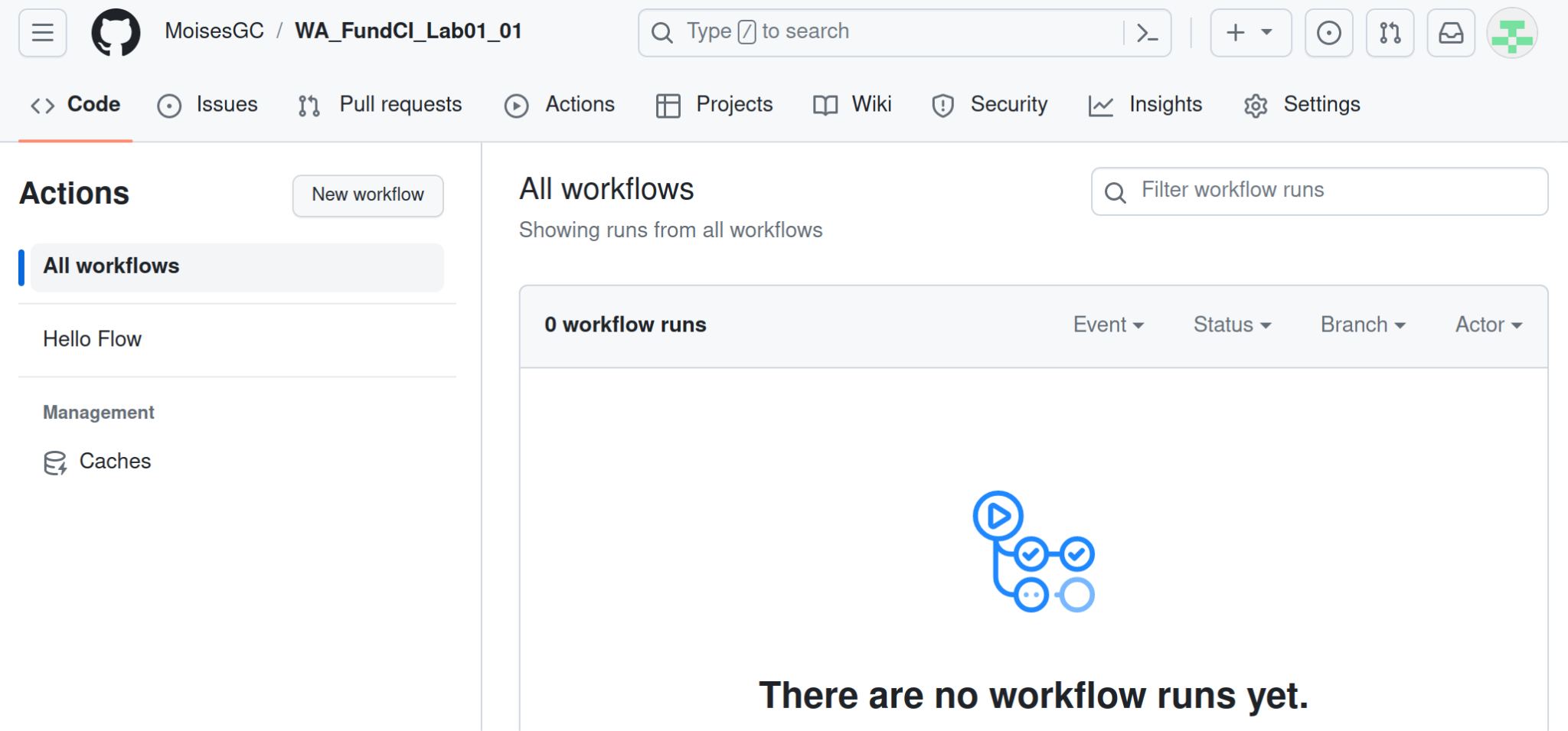
run: echo "Valeu - bye!"



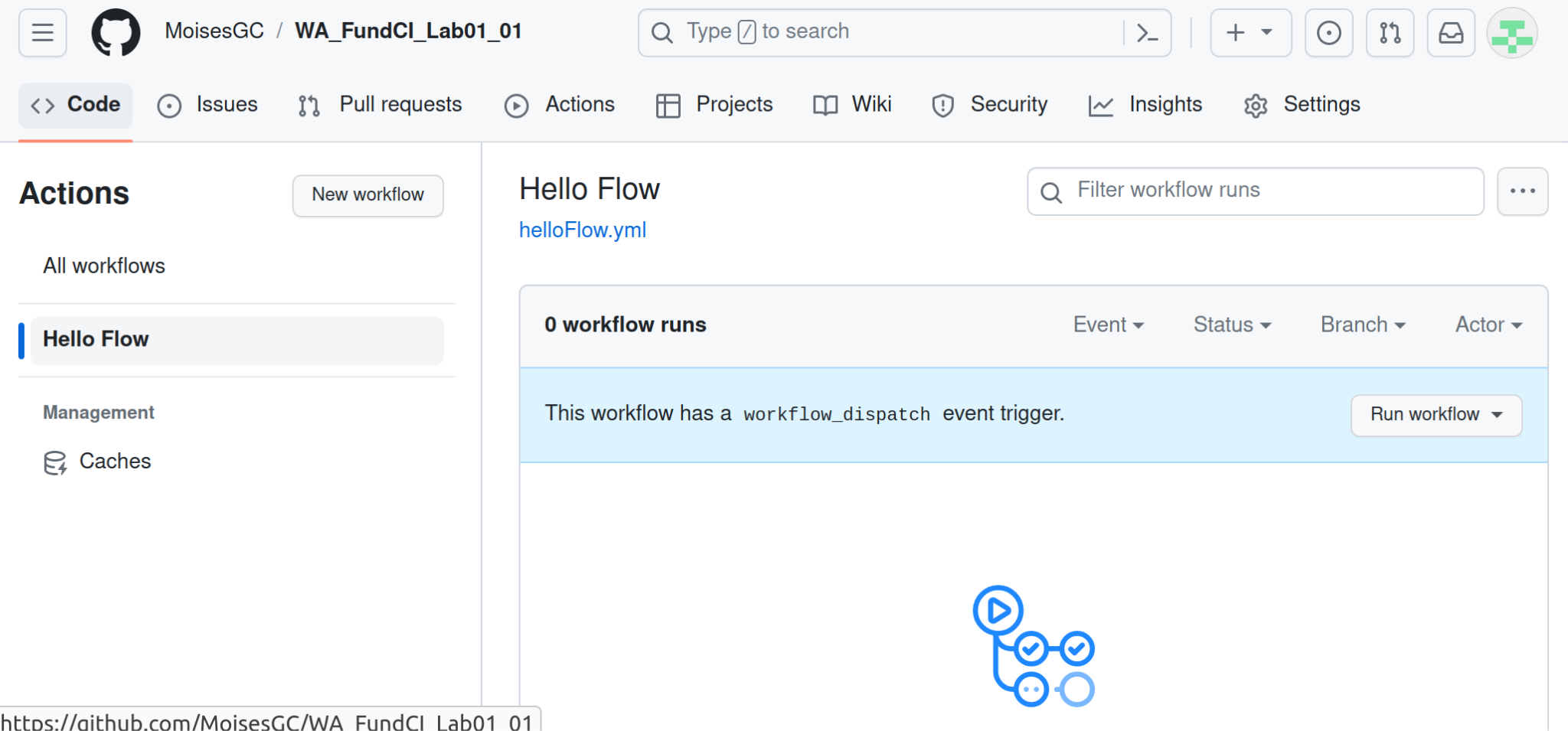
e o resultado é



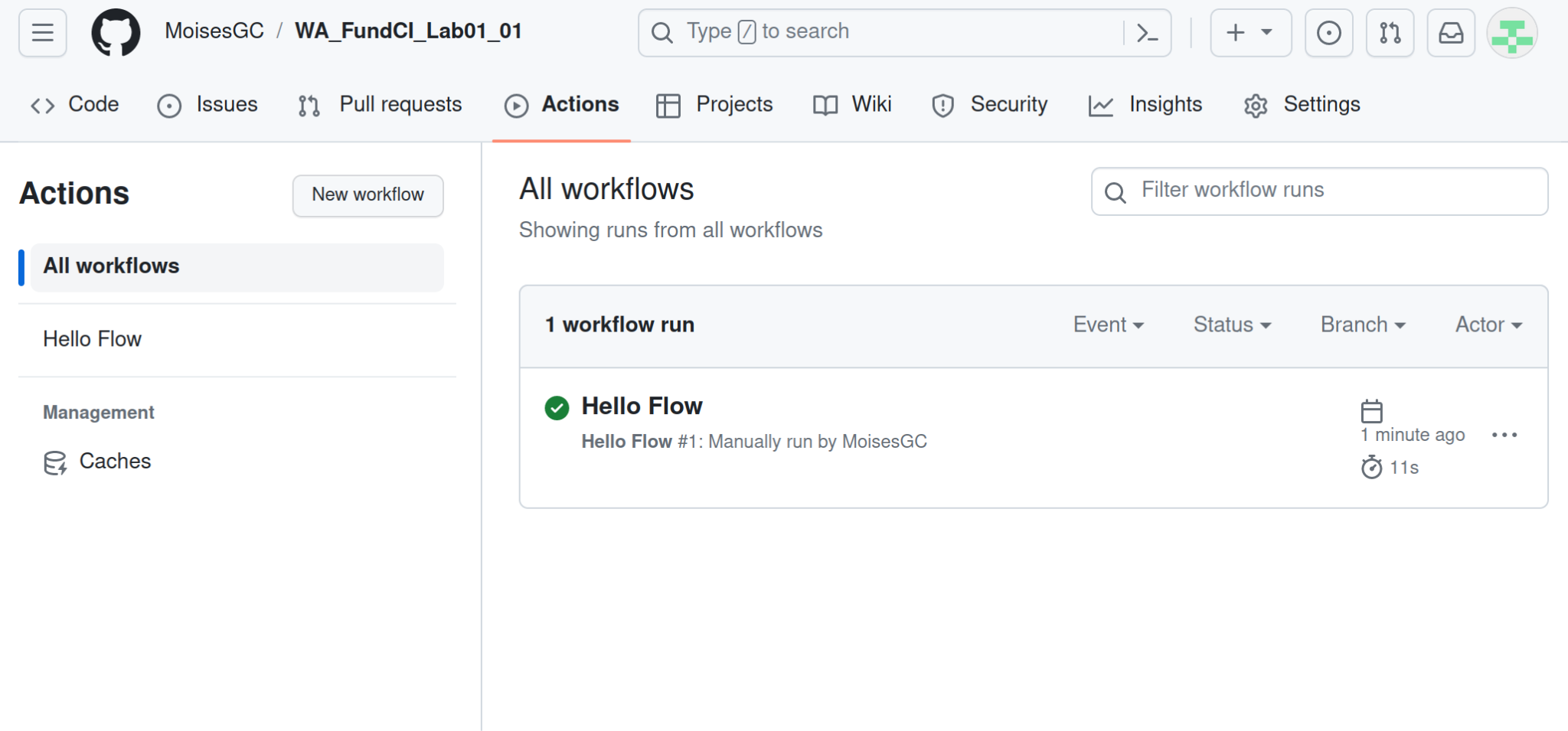
6 - Volte na tab “Actions” e note que ela está refletindo a atual configuração de workflows disponíveis.



7 - Clique no workflow “Hello Flow” e note que ele disponibiliza a opção de rodar manualmente

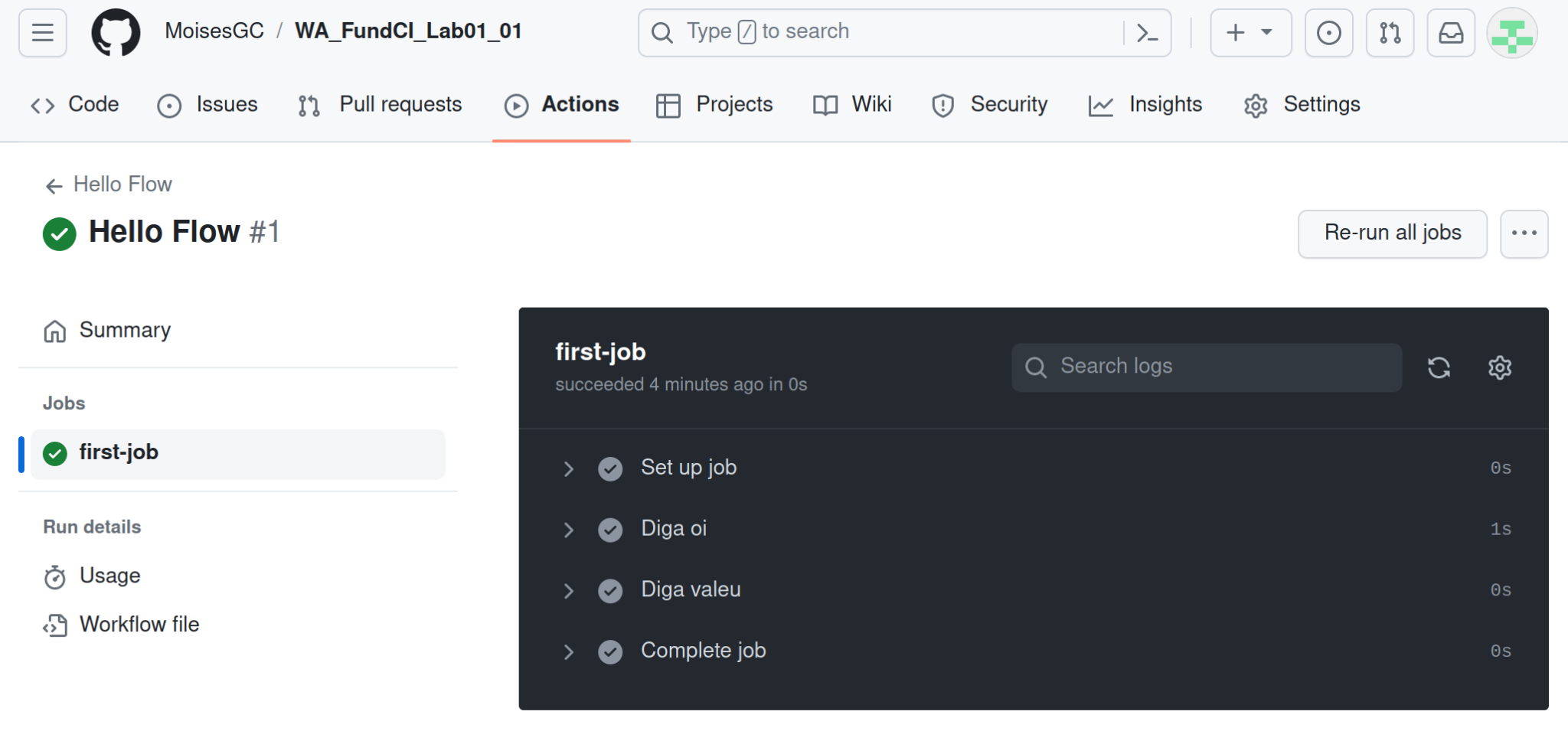


8 - Clique em “Run workflow” e aguarde ele ser executado (atualize a tela se ele nao atualizar).

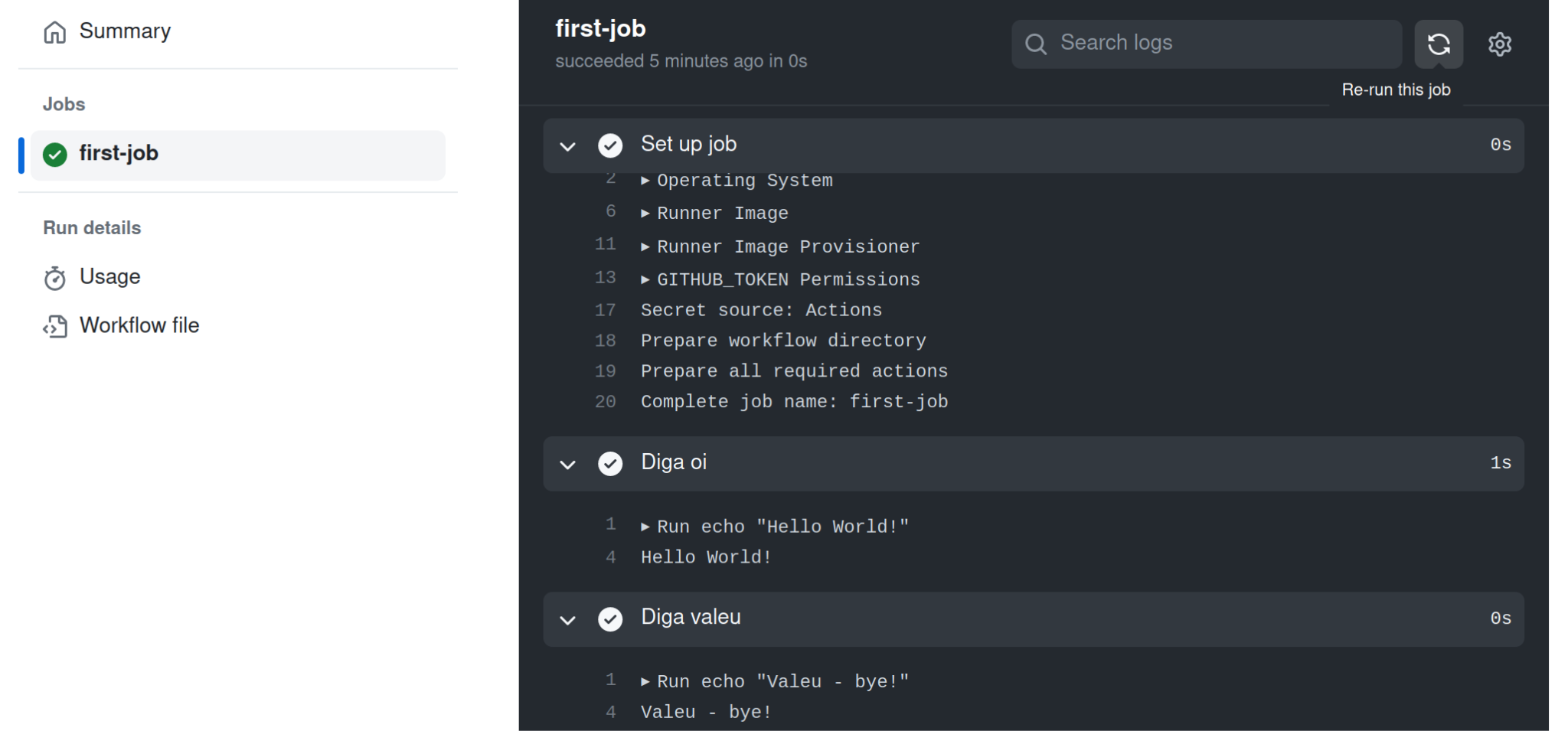


**Parabéns**! **Você executou o seu primeiro workflow!**

9 - Clique no nome do workflow e depois no job “first-job” que criamos em nosso arquivo .yml. e veja os passo que foram executados!



10 - Expanda as etapas para avaliar sua execução!



Parabéns!! Fim do exercício!

**Exercício 02 - Criando um Action de ativação programada**

Vamos agora acionar o sistema de agendamento de execução das actions!

1 - Volte no arquivo “helloFlow.yml” (acessando a tab actions) e edite-o.

2 - Altere ele para o seguinte conteúdo (vamos rodar a cada 5 mins) (dica forte: https://crontab-generator.org/)

name: Hello Flow

on:

schedule:

- cron: '\*/5 \* \* \* \*'

jobs:

first-job:

runs-on: ubuntu-latest

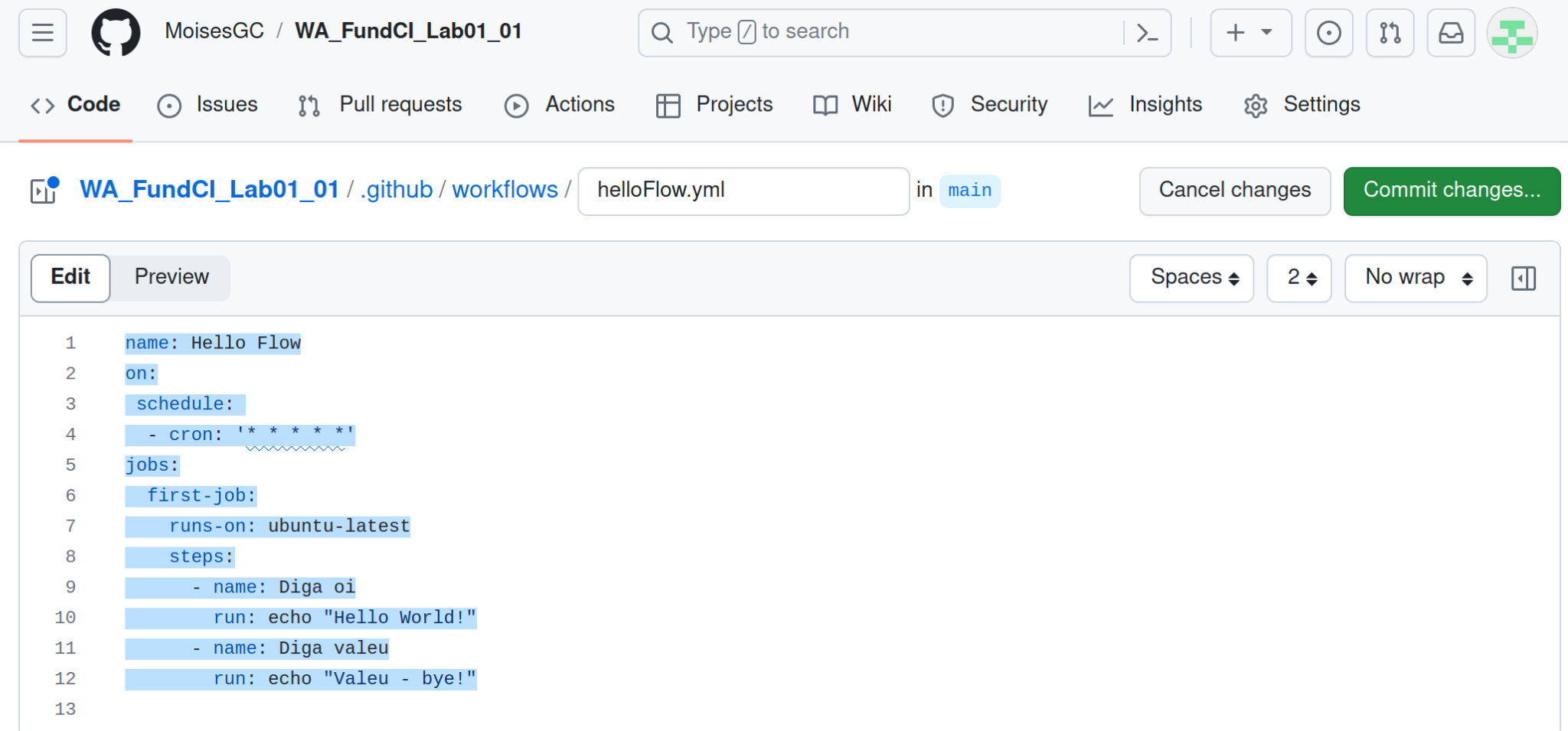
steps:

- name: Diga oi

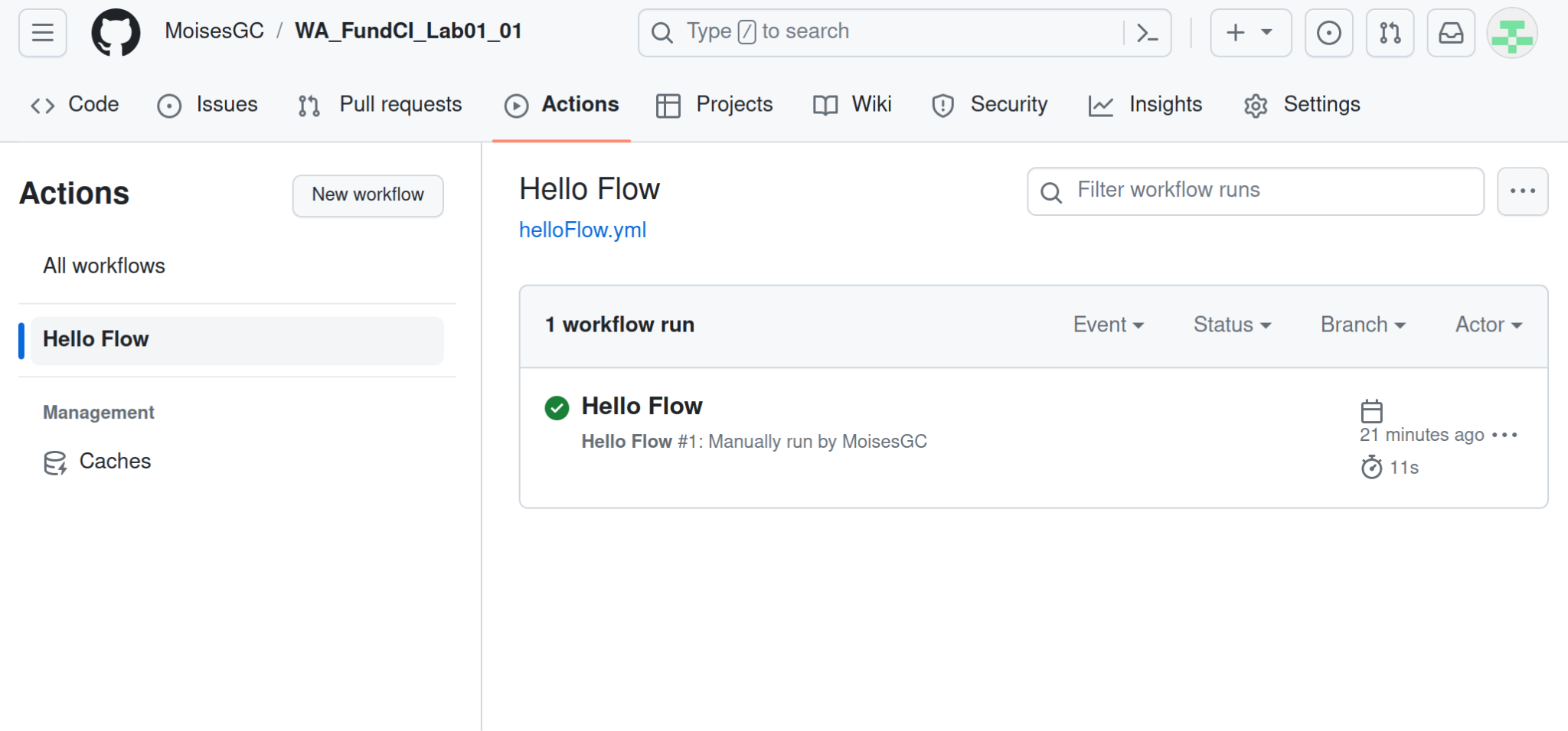
run: echo "Hello World!"

- name: Diga valeu

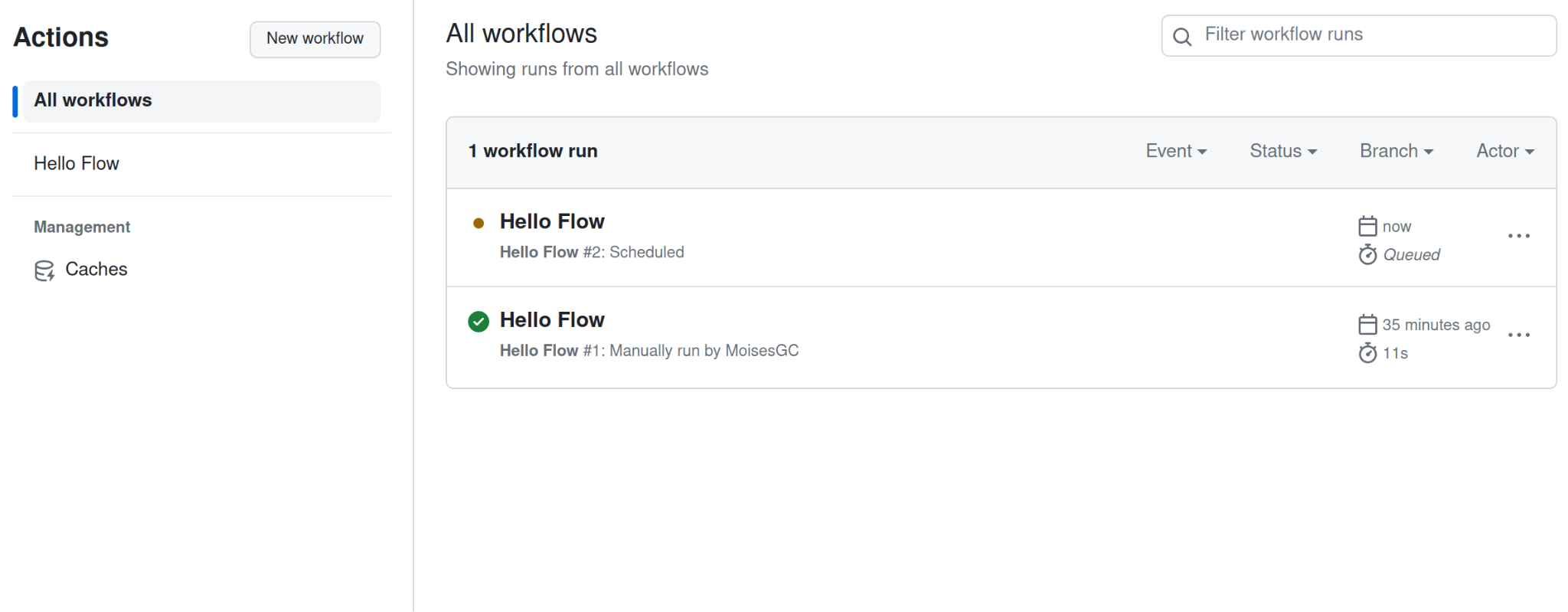
run: echo "Valeu - bye!"

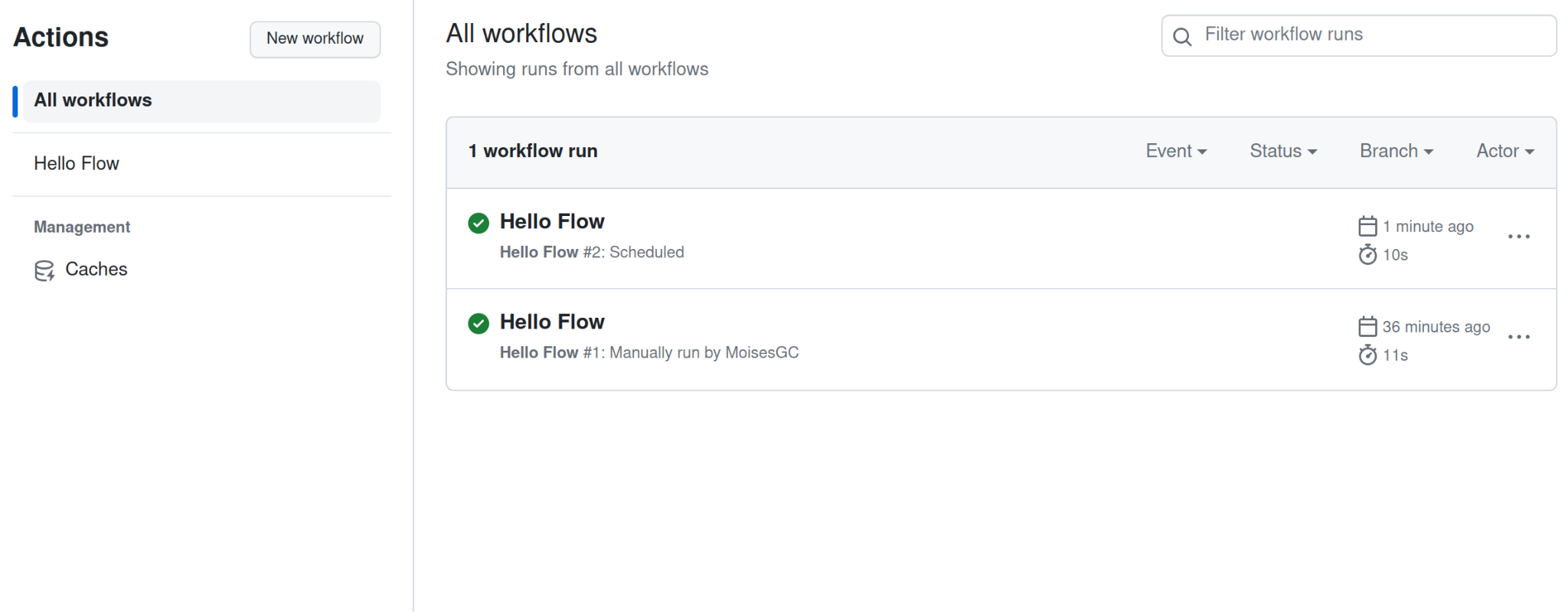


3 - Realize o commit desta modificação, ao voltar a tab Actions, note que não existe mais a opção de executar manualmente o workflow - agora ele funciona pelo controle de agendamento.



Quando o agendamento inicia, outro workflow aparece na lista de execução

e ao finalizar, temos



**4 - Parabéns, você ativou um workflow para funcionar via agendamento!**

Ref - https://docs.github.com/en/actions/using-workflows/events-that-trigger-workflows#schedule

**Exercício 03 - Criando um action de ativação por evento (um push)**

Para fechar, vamos agora considerar um push para ativar o workflow!

1 - Edite o seu .yml e adicione o seguinte conteúdo

name: Hello Flow

on:

push

jobs:

first-job:

runs-on: ubuntu-latest

steps:

- name: Diga oi

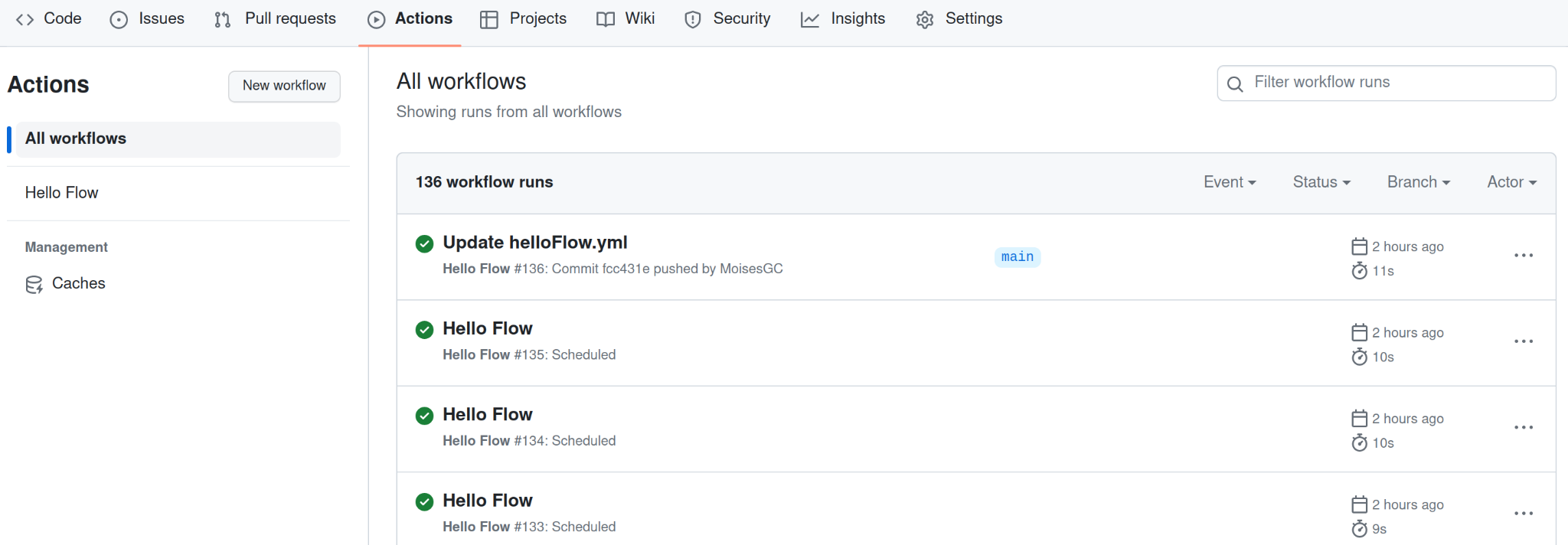
run: echo "Hello World!"

- name: Diga valeu

run: echo "Valeu - bye!"

Note que agora no “on”, colocamos a condição de push para ativar o flow - qualquer push neste repositório vai ativar essa tarefa. Até mesmo a propria edição do arquivo .yml! (ele faz parte do repositório).

2 - Ao terminar a edição, faça o commit da alteração e volte na tab do Actions.



3 - Perceba que a atualização do repositório já causou a execução do workflow!

**4 - Parabéns, você ativou um workflow para funcionar para qualquer push que for realizado no seu repositório.**

Nos próximos laboratórios, vamos utilizar uma abordagem mais realista adotando um repositório local que será sincronizado com o seu repositório no GitHub, utilizando um pequeno projeto web para a prática!!