

# Integração contínua e teste de cobertura

## CircleCi

Tutorial de uso do CircleCI, ferramenta para Integração contínua de dados.

### 1. Criando um Repositório

Primeiramente deve-se criar um repositório no GitHub (<https://github.com/new>.)  
Selecionar checkbox para Inicializar novo repositório com README

#### Create a new repository

A repository contains all the files for your project, including the revision history.

Owner



Repository name



Great repository names are short and memorable. Need inspiration? How about **bookish-meme**.

Description (optional)



Public

Anyone can see this repository. You choose who can commit.



Private

You choose who can see and commit to this repository.



**Initialize this repository with a README**

This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: **None**

Add a license: **None**



Create repository

### 2. Adicionando um arquivo YML

O CircleCI usa um arquivo YAML para identificar como você deseja que seu ambiente de teste seja configurado e quais testes você deseja executar. Esse arquivo deve ser chamado `config.yml` deve ser uma pasta oculta com o nome `.circleci`

Após isso, criar uma pasta no github com o botão "create new file" e com o nome `.circleci/config.yml`

Para começar com um simples `config.yml` adicione o seguinte texto:

```
version: 2
```

```
jobs:
```

```
  build:
```

```
    docker:
```

```
      - image: circleci/ruby:2.4.1
```

```
    steps:
```

```
      - checkout
```

```
      - run: echo "A first hello"
```

### 3. Configurando sua compilação no CircleCI

Primeiramente crie uma conta no circleCi (<https://circleci.com/signup/>) faça login no CircleCi com seu GitHub, em seguida irá aparecer uma página de gerenciamento dos seus projetos.

Para adicionar um novo repositório acesse a aba “Add Projects” e selecione o projeto desejado com “Set up projects”.

Haverá algumas opções após a seleção do projeto, como, sistema operacional, linguagem, mas deixe como está e clique no botão “Start Building” para começar a compilação.

### 4. Compilando seu CircleCi

Após isso, poderá ser visto se seu código foi compilado com sucesso ou obteve algum erro, caso tenha passado poderá ver as seguintes partes da compilação :

**-Spin up environment**

**-Checkout code**

**-echo**

### 5. Visualizando fluxo de trabalho

Para melhor visualizar o fluxo de trabalho, edite seu arquivo

*.circleci/config.yml* e coloque:

```
version: 2
jobs:
  build:
    docker:
      - image: circleci/ruby:2.4.1
    steps:
      - checkout
      - run: echo "A first hello"
  build:
    docker:
      - image: circleci/ruby:2.4.1
    steps:
      - checkout
      - run: echo "A first hello"
```

Os fluxos de trabalho será mostrado após o uso do seguinte código

```
workflows:
  version: 2
  one_and_two:
    jobs:
      - one
      - two
```

Faça o commit dessas mudanças no git e poderá ver o fluxo nos painéis do CircleCI

Clicar nos links para poder ver os dois trabalhos em paralelo.