

EMBARCADERO CONFERENCE



embarcadero®

Automatização de build não é só para os grandes

Luiz Sfolia



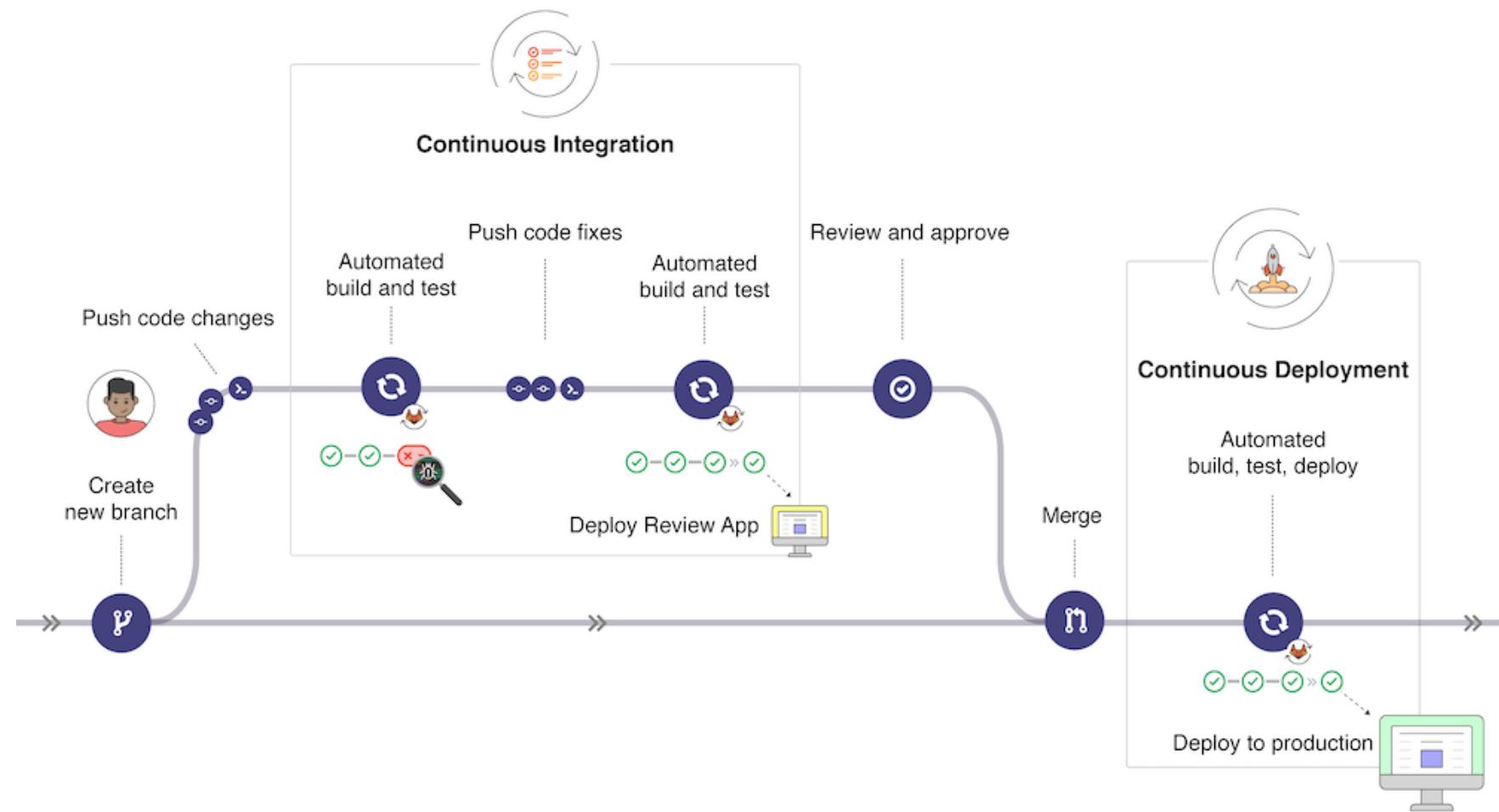
CONFERENCE
EMBARCADERO CONFERENCE 2022

Automação de Build não é só para os grandes

- 1-Demanda
- 2-Analise e cadastramento do Ticket
- 3-Priorização
- 4-Desenvolvimento – Nova branch
- 5-Testes (Unitários e Métricas)
- 6-Code Review
- 7-Reintegração da Branch/Merge
- 8-Teste de Recessão
- 9-Liberação Versão - Release



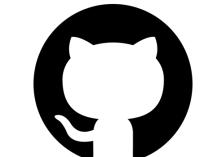
Pipeline



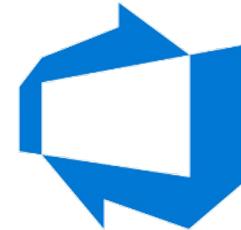
CI/CD Como implementar ?



GitLab



GitHub



Azure DevOps



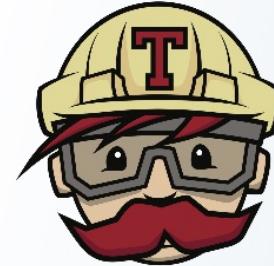
Jenkins



circleci



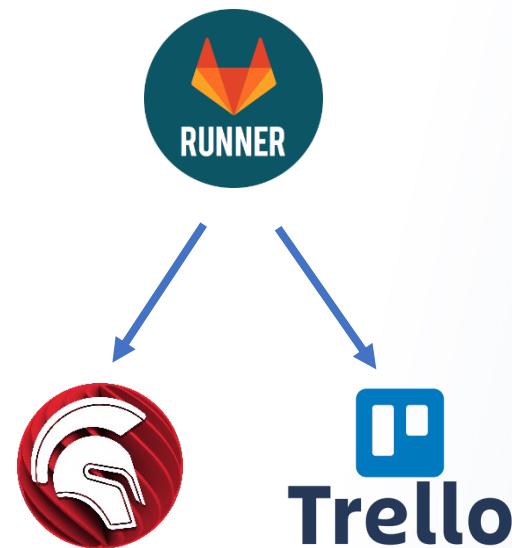
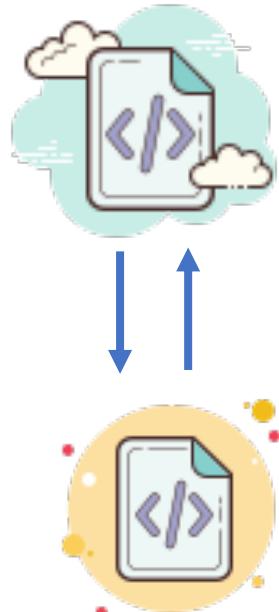
ATLASSIAN
Bitbucket



Travis CI



CI/CD Como implementar ?



CI/CD Como implementar ?

- [!\[\]\(313b3b3c8a0c38ad35f0f4cceb5f9abb_img.jpg\) Project information](#)
- [!\[\]\(01fb5058363dcb3bfe1ee1159e9c248e_img.jpg\) Repository](#)
- [!\[\]\(54f0ad8b6afbf069171bcb3f2d838cc1_img.jpg\) Merge requests 0](#)
- [!\[\]\(9e65eb946a0821820bf150eaecad484c_img.jpg\) CI/CD](#)
- [!\[\]\(39de8e92d71816f94aa94a39563f909f_img.jpg\) Security & Compliance](#)
- [!\[\]\(fac943d56447ae39baf4c6cb02da634d_img.jpg\) Deployments](#)
- [!\[\]\(c89ae5d41e0aa2f58d7ffc6342362147_img.jpg\) Packages and registries](#)
- [!\[\]\(5f2dfe8ea2fcdf131acae7d8bf5cb0f0_img.jpg\) Infrastructure](#)
- [!\[\]\(b2a125222f62a0e819eaaa961d2a1086_img.jpg\) Monitor](#)
- [!\[\]\(a0d47a0f54b7880ec05b83106e47b581_img.jpg\) Analytics](#)
- [!\[\]\(8485711a8e03ae764209496eab2b7a7c_img.jpg\) Settings](#)
 - [General](#)
 - [Integrations](#)
 - [Webhooks](#)
 - [Repository](#)
 - [Merge requests](#)
- [!\[\]\(2d9568d9f568be2271ef52e0b3b963fb_img.jpg\) CI/CD](#)
 - [Packages and registries](#)
 - [Pages](#)
 - [Monitor](#)
 - [Usage Quotas](#)

Search page

General pipelines

Customize your pipeline configuration.

[Expand](#)

Auto DevOps

Automate building, testing, and deploying your applications based on your continuous integration and delivery configuration. [How do I get started?](#)

[Expand](#)

Runners

Runners are processes that pick up and execute CI/CD jobs for GitLab. [What is GitLab Runner?](#)

[Expand](#)

Artifacts

A job artifact is an archive of files and directories saved by a job when it finishes.

[Expand](#)

Variables

Variables store information, like passwords and secret keys, that you can use in job scripts. [Learn more.](#)

Variables can be:

- **Protected:** Only exposed to protected branches or protected tags.
- **Masked:** Hidden in job logs. Must match masking requirements. [Learn more.](#)

[Expand](#)

Pipeline triggers

[Expand](#)



CI/CD Como implementar ?

Project information

Repository

Merge requests 0

CI/CD

Security & Compliance

Deployments

Packages and registries

Infrastructure

Monitor

Analytics

Settings

General

Integrations

Webhooks

Repository

Merge requests

CI/CD

Packages and registries

Pages

Monitor

Usage Quotas

Runners

Runners are processes that pick up and execute CI/CD jobs for GitLab. [What is GitLab Runner?](#)

Register as many runners as you want. You can register runners as separate users, on separate servers, and on your local machine.

Runners are either:

- **active** - Available to run jobs.
- **paused** - Not available to run jobs.

Specific runners

These runners are specific to this project.

Set up a specific runner for a project

1. [Install GitLab Runner and ensure it's running.](#)
2. Register the runner with this URL:
<https://gitlab.com/>

And this registration token:

[Reset registration token](#)

[Show runner installation instructions](#)

Available specific runners

#17402918 (in-y-qzT)

[Remove runner](#)

LUIZSFOLIA07D2

#17298046 (2qgvMeph)

[Remove runner](#)

Shared runners

These runners are shared across this GitLab instance.

Shared runners on GitLab.com run in [autoscale mode](#) and are powered by Google Cloud Platform. Autoscaling means reduced wait times to spin up builds, and isolated VMs for each project, thus maximizing security.

They're free to use for public open source projects and limited to 400 CI/CD minutes per month per group for private projects. Read about all [GitLab.com plans](#).

Enable shared runners for this project



Available shared runners: 30

#11728733 (ZhKvGs-d)

4-blue.shared-gitlab-org.runners-manager.gitlab.com/dind
gitlab-org-docker

#11574076 (8zCxmpPt)

2-green.shared-gitlab-org.runners-manager.gitlab.com
gitlab-org



TMR

CONSULTORIA &
DESENVOLVIMENTO

CI/CD Como instalar o runner

- [Project information](#)
- [Repository](#)
- [Merge requests 0](#)
- [CI/CD](#)
- [Security & Compliance](#)
- [Deployments](#)
- [Packages and registries](#)
- [Infrastructure](#)
- [Monitor](#)
- [Analytics](#)
- [Settings
 - General
 - Integrations
 - Webhooks
 - Repository
 - Merge requests
 - CI/CD**
 - Packages and registries
 - Pages
 - Monitor
 - Usage Quotas](#)

Install a runner

X

Environment

Linux macOS **Windows** Docker Kubernetes

Architecture

amd64 ▾

Download and install binary

[Download latest binary](#)

```
# Run PowerShell: https://docs.microsoft.com/en-us/powershell/scripting/windows-powershell/starting-windows-powershell?view=powershell-7#with-administrative-privileges-run-as-administrator
# Create a folder somewhere on your system, for example: C:\GitLab-Runner
New-Item -Path 'C:\GitLab-Runner' -ItemType Directory

# Change to the folder
cd 'C:\GitLab-Runner'

# Download binary
Invoke-WebRequest -Uri "https://gitlab-runner-downloads.s3.amazonaws.com/latest/binaries/gitlab-runner-windows-amd64.exe" -OutFile "gitlab-runner.exe"

# Register the runner (steps below), then run
.\gitlab-runner.exe install
.\gitlab-runner.exe start
```

Command to register runner

```
./gitlab-runner.exe register --url https://gitlab.com/ --registration-token $REGISTRATION_TOKEN
```

[Close](#)



Rodar o Delphi sem instalar

- Copiar para pasta -> C:\DelphiBuild_10_4
 - C:\Program Files (x86)\Embarcadero\Studio\21.0\bin
 - C:\Program Files (x86)\Embarcadero\Studio\21.0\bin64
 - C:\Program Files (x86)\Embarcadero\Studio\21.0\lib
 - %AppData%\Roaming\Embarcadero\BDS\21 – (EnvOptions.dproj)



Rodar o Delphi sem instalar

- Build.bat -> C:\ GitLab-Runner

```
@SET BDS=C:\DelphiBuild_10_4
@SET BDSINCLUDE=C:\DelphiBuild_10_4\include
@SET BDSCOMMONDIR=C:\DelphiBuild_10_4\Public
@SET FrameworkDir=C:\Windows\Microsoft.NET\Framework\v4.0.30319
@SET FrameworkVersion=v4.5
@SET FrameworkSDKDir=
@SET
PATH=%FrameworkDir%;%FrameworkSDKDir%;C:\DelphiBuild_10_4\bin;C:\DelphiBuild_10_4\bin64;C
:\DelphiBuild_10_4\cmake;%PATH%
@SET LANGDIR=EN
@SET PLATFORM=
@SET PlatformSDK=

C:\Windows\Microsoft.NET\Framework64\v4.0.30319\msbuild.exe %1.dproj /target:Build
/p:"Config=Release" /p:"Platform=Win32"
```



CI/CD Como implementar ?

- [!\[\]\(e9b39248487d5213c295d272578e4234_img.jpg\) Project information](#)
- [!\[\]\(3e48acfe78b7158655caf828b3a39d86_img.jpg\) Repository](#)
- [!\[\]\(1da7092a7e45cf36ed94217841535ae3_img.jpg\) Issues 0](#)
- [!\[\]\(cc07f8d3215e993d9867b89a99f56db9_img.jpg\) Merge requests 0](#)
- [!\[\]\(774e1f6119078522a572d391acae4f52_img.jpg\) CI/CD](#)
- [!\[\]\(d0ad233e20354b0d174994eaa6bf13c0_img.jpg\) Pipelines](#)
- [!\[\]\(f81e5fc84506a23bd8cd404b0abd6388_img.jpg\) Editor](#)
- [!\[\]\(9316b1b88a093265e51c70a69a489e9d_img.jpg\) Jobs](#)
- [!\[\]\(67289ac87915fb4b039a38e627ee6d35_img.jpg\) Schedules](#)
- [!\[\]\(ef8a0e3a3ab4bb2078c8640a4898c9b8_img.jpg\) Security & Compliance](#)
- [!\[\]\(91e82bc40f50422c3c3de7de7b6de7be_img.jpg\) Deployments](#)
- [!\[\]\(e4ca6f7c3ba3a213f84c64aca829f9c2_img.jpg\) Packages and registries](#)
- [!\[\]\(d2b1ec01aec40f75270fd92132d4392a_img.jpg\) Infrastructure](#)
- [!\[\]\(7ad9e215e1befbc5ba9cefc513155f5e_img.jpg\) Monitor](#)
- [!\[\]\(d4920a2dacde8bb2f4ccf976d2204de9_img.jpg\) Analytics](#)
- [!\[\]\(5ac43296f3a5b3573a5152e6f4de9ea6_img.jpg\) Wiki](#)
- [!\[\]\(f083761f2c3238a65af0b9dcb24ebe6c_img.jpg\) Snippets](#)
- [!\[\]\(c3d0c25dfc218277935035dceaa1ce59_img.jpg\) Settings](#)

Get started with GitLab CI/CD

-  Runners are available to run your jobs now

GitLab Runner is an application that works with GitLab CI/CD to run jobs in a pipeline. There are active runners available to run your jobs right now. If you prefer, you can [configure your runners](#) or [learn more](#) about runners.

Learn the basics of pipelines and .yml files

Use a sample `.gitlab-ci.yml` template file to explore how CI/CD works.



"Hello world" with GitLab CI

Get familiar with GitLab CI syntax by setting up a simple pipeline running a "Hello world" script to see how it runs, explore how CI/CD works.

[Try test template](#)

Ready to set up CI/CD for your project?

Use a template based on your project's language or framework to get started with GitLab CI/CD.

CONSULTORIA &
DESENVOLVIMENTO

CI/CD Configuration

Project information

Repository

Merge requests 0

CI/CD

Security & Compliance

Deployments

Packages and registries

Infrastructure

Monitor

Analytics

Settings

General

Integrations

Webhooks

Repository

Merge requests

CI/CD

Packages and registries

Pages

Monitor

Usage Quotas

main

✓ Pipeline syntax is correct. Learn more

Edit Visualize Validate NEW View merged YAML

Browse templates Help

```

1  # This file is a template, and might need editing before it works on your project.
2  # This is a sample GitLab CI/CD configuration file that should run without any modifications.
3  # It demonstrates a basic 3 stage CI/CD pipeline. Instead of real tests or scripts,
4  # it uses echo commands to simulate the pipeline execution.
5  #
6  # A pipeline is composed of independent jobs that run scripts, grouped into stages.
7  # Stages run in sequential order, but jobs within stages run in parallel.
8  #
9  # For more information, see: https://docs.gitlab.com/ee/ci/yaml/index.html#stages
10 #
11 # You can copy and paste this template into a new '.gitlab-ci.yml' file.
12 # You should not add this template to an existing '.gitlab-ci.yml' file by using the `include:` keyword.
13 #
14 # To contribute improvements to CI/CD templates, please follow the Development guide at:
15 # https://docs.gitlab.com/ee/development/cicd/templates.html
16 # This specific template is located at:
17 # https://gitlab.com/gitlab-org/gitlab/-/blob/master/lib/gitlab/ci/templates/Getting-Started.gitlab-ci.yml
18
19 stages:      # List of stages for jobs, and their order of execution
20   - build
21   - test
22   - deploy
23
24 build-job:  # This job runs in the build stage, which runs first.
25   stage: build
26   script:
27     - echo "Building"
28
29 test-job:   # This job runs in the test stage, which runs second.
30   stage: test
31   script:
32     - echo "Testing"
33
34 deploy-job: # This job runs in the deploy stage, which runs third.
35   stage: deploy
36   script:
37     - echo "Deploying"

```

Commit message Update .gitlab-ci.yml file

Branch main

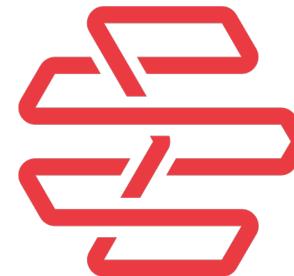
Commit changes Reset



Mão na massa



Obrigado



TMR

CONSULTORIA &
DESENVOLVIMENTO



Outsourcing de TI



Automação de Testes



Desenvolvimento de Software



Migração de Sistemas Legados



Consultoria em TI



Soluções IOT



Treinamentos



ACBr



DevOps



Licenças Delphi

- 
-  Luiz.sfolia@tmrti.com.br
 -  [@luizsfolia](https://twitter.com/luizsfolia)
 -  [luizsfolia](https://facebook.com/luizsfolia)
 -  [@sfolialuiz](https://instagram.com/sfolialuiz)
 -  [Linkedin.com/in/luizsfolia](https://linkedin.com/in/luizsfolia)