

Integração e Entrega Contínua - P2

Entrega até 27/06/2025.

Não haverá outra prova.

Atenção aos requisitos.

Para a nota completa, enviar os prints solicitados após a validação.

Cada requisito contabiliza o valor da nota final, itens não entregues/não funcionais não entrarão na soma. Peso maior para os itens da etapa CD.

Adicionar o meu usuário como contribuir no repositório (festmedeiros)

Eu devo conseguir rodar o pipeline para validar o fluxo.

Enviar um print com a tela do serviço criado no Render.

Enviar um print com os logs da aplicação no Better Stack.

Após a minha validação, enviar os prints dos logs gerados pelo meu teste.

Requisitos

Código

- O código não será avaliado e pode utilizar os exemplos das aulas anteriores
- Criar uma API Rest
- Endpoints devem ser expostos via Swagger
- API Deve logar todas as requisições (seja sucesso ou erro - try / catch)
- Deve efetuar um CRUD em um banco de dados relacional (MySQL / Postgre / etc)
- API deve possuir integração com o BetterStack para geração dos Logs

BetterStack - Logs

- Criar um stream para logs
- Integrar os logs da API com o BetterStack

Github

- Criar de um repositório backend
- Adicionar o meu usuário como contribuidor no repositório da P2 (festmedeiros)
- Utilizar de Gitflow
- Utilizar de padronização de commits
- Utilizar Secrets para todos os dados sensíveis (tokens de api, códigos de serviços, senhas de banco de dados, etc)
- Criar um pipeline CI/CD
- Etaps do CI
 - Checkout
 - Install
 - Build
 - Versionamento
 - Build Imagem
- Etapas do CD
 - Deploy da imagem no Docker Hub com a mesma versão da calculada no Github
 - Criação de uma TAG Latest no Docker Hub
 - Atualização de variáveis/secrets no Render (Log, Banco de dados, etc)
 - Deploy da API no Render com a imagem e versão gerada no build
- Envio de e-mail para os casos de erro em qualquer etapa do pipeline

Docker Hub

- Criar um repositório com o mesmo nome do repositório do Github
- Efetuar deploy da imagem Docker através do Github

Render

- Criar um service para hospedar a API no Render
- Garantir que os deploys no Render ocorram com sucesso
- Garantir que o deploy da imagem Docker acontece com a mesma versão de TAG do Github
- Enviar a rota pública da API hospedada no Render

Banco de dados online

- Utilizar um banco de dados relacional online
- Integrar a API ao banco de dados
- Cadastrar as credenciais do banco de dados no Github Secrets e injetar as mesmas no

momento de deploy no Render

Docker

- Utilizar uma imagem do docker hub para banco de dados no Docker compose
- Utilizar uma imagem local do backend
- Criar um Docker compose com as duas peças: Backend e Banco de dados
- O compose deve garantir que ocorra a integração entre as duas peças
- O CRUD deve funcionar rodando com os dois containers localmente

Prints pré-validação

- Tela do serviço criado no Render
- Logs da aplicação no Better Stack

Prints pós-validação (Vou pedir via Teams)

- Better Stack: Logs gerados durante a validação
- Render: Eventos de deploy gerados durante validação

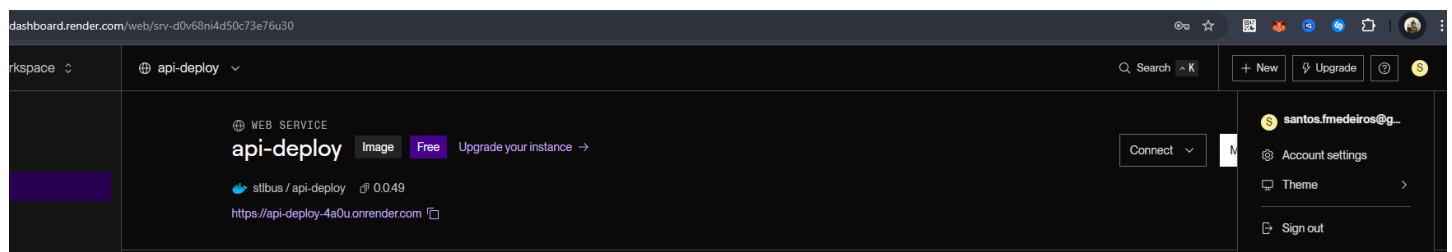
Requisitos adicionais:

Controle de TAGs geradas


- Garantir que exista um controle no Github Actions para vincular cada build com a TAG gerada
- Se tentar rodar um build antigo, garantir que o deploy no Render aconteça com a TAG específica daquele build

Exemplo de Prints antes da validação

Render



Eventos Render



 WEB SERVICE


api-deploy

Image

Free

Upgrade your instance →

 stlbus / api-deploy  0.0.49

<https://api-deploy-4a0u.onrender.com> 

 Your free instance will spin down with inactivity, which can delay requests by 50 seconds or more

June 6, 2025 at 5:20 PM

✓ Live

Deploy for 40f4055  0.0.49

All logs ▾

 Search

Jun 6 05:20:42 PM 

Jun 6 05:20:42 PM  > api-docker@0.0.49 start

Jun 6 05:20:42 PM  > node index.js

Jun 6 05:20:42 PM 

Jun 6 05:20:44 PM  LOGTAIL_URL14 <https://s1331889.eu-nbg-2.betterstackdata.com>

Jun 6 05:20:44 PM  LOGTAIL_TOKEN14 HNQYSvt4PeWN2mbH2FKAas9V

Jun 6 05:20:44 PM  Servidor rodando na porta 10000

Jun 6 05:20:44 PM  Erro ao conectar ao MySQL: Error: getaddrinfo ENOTFOUND mysql

Jun 6 05:20:44 PM  at GetAddrInfoReqWrap.onlookup [as oncomplete] (node:dns:107:26) {

Jun 6 05:20:44 PM  errno: -3008,

Jun 6 05:20:44 PM  code: 'ENOTFOUND',

Jun 6 05:20:44 PM  syscall: 'getaddrinfo',

Jun 6 05:20:44 PM  hostname: 'mysql',

Jun 6 05:20:44 PM  fatal: true

Jun 6 05:20:44 PM  }

Jun 6 05:20:53 PM  --> Your service is live 🎉

Better Stack

