

DESM2 - Desafio do Módulo 2

Entrega 14 fev em 21:00 **Pontos** 40 **Perguntas** 15
Disponível até 14 fev em 21:00 **Limite de tempo** Nenhum

Instruções

O Desafio do Módulo 2 está disponível!

1. Instruções para realizar o desafio

Consulte a data de entrega no teste e em seu calendário.

Reserve um tempo para realizar a atividade e leia as orientações e enunciados com atenção. Em caso de dúvidas, utilize o "Fórum de dúvidas do Desafio do Módulo 2".

Para iniciar o Desafio, clique em "Fazer teste". Você tem somente **uma** tentativa e não há limite de tempo definido para realizá-la. Caso você precise interromper a atividade, apenas deixe a página e, ao retornar, clique em "Retomar teste".

Lembre-se de conferir todas as questões antes de enviar a atividade, e clique em "Enviar teste" **somente** quando você tiver terminado.

Uma vez terminado o prazo de entrega, a plataforma enviará as tentativas não finalizadas automaticamente. Por isso, fique atento ao prazo final.

Novas tentativas só serão concedidas perante a apresentação de atestado médico.

O gabarito será disponibilizado a partir de sexta-feira, **17/02/2023**, às 23h59.

Bons estudos!

2. O arquivo abaixo contém o enunciado do desafio

[Enunciado do Desafio - Módulo 2 - Bootcamp Profissional DevOps.pdf](#)

Histórico de tentativas

	Tentativa	Tempo	Pontuação
MAIS RECENTE	<u>Tentativa 1</u>	1.332 minutos	34,66 de 40

⚠ As respostas corretas estarão disponíveis em 17 fev em 23:59.

Pontuação deste teste: **34,66** de 40

Enviado 7 fev em 18:51

Esta tentativa levou 1.332 minutos.

Incorreta

Pergunta 1

0 / 2,67 pts

No Forking Workflow cada desenvolvedor possui seu próprio repositório local e depois abre um Pull Request para o repositório principal. Qual a principal vantagem desta estratégia?



Cada desenvolvedor pode desfazer suas alterações a hora que quiser, sem afetar os outros desenvolvedores.



Os desenvolvedores integram o código produzido sem a necessidade de realizar as modificações direto no repositório central.



Todas as alternativas.



Os desenvolvedores não trabalham direto no repositório central, o que pode causar conflitos.

Pergunta 2

2,67 / 2,67 pts

No Gitflow Workflow existem dois tipos de ramificações: a Develop e a Master. Qual delas é responsável por levar o código para a produção?



Ambas.



Nenhuma delas, é criada uma outra branch para isso.

☐ A primeira, a Develop.

☒ A segunda, a Master.

Pergunta 3

2,67 / 2,67 pts

No Release Blue/Green, existem dois ambientes muito parecidos, onde colocamos a próxima versão do código que será promovida para produção. É nesta situação que migramos TODOS os usuários para a nova versão. Essa afirmação está CORRETA?

☒ Sim.

☐ Não sei dizer.

☐ Parcialmente correta.

☐ Não.

Pergunta 4

2,67 / 2,67 pts

O Release Canário é muito parecido com o Blue/Green, a diferença é que:

☐ Podemos reverter a versão que está ativa para a versão anterior caso algo dê errado.

☐ Os dois ambientes podem ser diferentes, mas o mais idênticos possível.

☐ Temos mais de 2 ambientes para alterar os ambientes de produção.



Aumentamos o número de usuários de forma gradual e não migramos os usuários todos de uma vez só.

Pergunta 5

2,67 / 2,67 pts

Qual destas características NÃO é de uma aplicação baseada em microsserviços?



Coleção de pequenos serviços desacoplados que se comunicam geralmente através de HTTP.



Possui camadas sobrepostas com responsabilidades distintas.



Ramificação do padrão de Arquitetura Orientada a Serviço (SOA).



Distribuída em diversas sub aplicações independentes e desacopladas.

Pergunta 6

2,67 / 2,67 pts

Qual destes é um requisito para escolher uma arquitetura monolítica?



Se você tem um time experiente e capaz de desenvolver em várias linguagens.



Se você está criando algo com nível alto de incerteza, tipo um MVP ou uma PoC para validar um negócio ou produto.

- ☐ Se escalabilidade é um requisito fundamental.
- ☐ Se uma parte da aplicação precisa ser extremamente eficiente.

Pergunta 7**2,67 / 2,67 pts**

Qual a diferença fundamental entre containers e Máquinas Virtuais?

- ☒ Containers não utilizam hypervisor como as Máquinas Virtuais, ou seja, Container não tem uma visão geral do ambiente físico fora de seu espaço, como a Máquina Virtual tem.
- ☐ Máquina Virtual utiliza a virtualização e Container não.
- ☐ No Container você pode instalar todos os programas de que precisa em uma mesma máquina.
- ☐ Não existe diferença alguma.

Pergunta 8**2,67 / 2,67 pts**

Quais destas plataformas não é uma plataforma de orquestração do Docker em produção?

- ☐ Apache Mesos.
- ☐ Kubernetes.
- ☐ Docker Swarm.
- ☒ Google Cloud.

Pergunta 9**2,67 / 2,67 pts**

Quais são os principais componentes do Kubernetes?

- ☒ Cluster, Plano de Controle, Kubelet e Pod.
- ☐ Cluster, Docker, Kubelet e Pod.
- ☐ Docker, Cluster, Pod, Console e Cloud Build.
- ☐ Cluster, Kubelet e Pod.

Pergunta 10**2,67 / 2,67 pts**

É a finalidade de orquestração de containers, EXCETO:

- ☐ Escala ou remoção de containers com base no balanceamento de cargas de trabalho na infraestrutura.
- ☐ Configuração da aplicação com base no container em que ela será executada.
- ☒ Gerenciamento dos testes automatizados.
- ☐ Balanceamento de carga e roteamento de tráfego.

Incorreta**Pergunta 11****0 / 2,67 pts**

Por que eu preciso criar ou obter uma imagem para subir um container?

☐ Todas as alternativas.

☒ Porque a imagem faz o download do container.

☐ Porque a imagem já está pronta e eu não preciso de mais nada.

☐

Por que uma imagem é um conjunto de linhas de comando que criam todo o ambiente que eu preciso para o container ser criado, desde bibliotecas até variáveis de ambiente.

Pergunta 12

2,67 / 2,67 pts

Qual o comando eu uso para ver todos os containers em execução, mesmo os que já foram encerrados/parados?

☐ docker stop

☒ docker ps -a

☐ docker rm

☐ docker run [nome_container]

Pergunta 13

2,67 / 2,67 pts

Dos comandos abaixo, qual deles NÃO corresponde ao comando para ser usado na criação de uma imagem?

☐ EXPOSE expõe uma porta qualquer.

☐ WORKDIR muda o diretório atual (similar ao comando cd em um terminal).

☒ MK DIR cria um diretório no sistema operacional.

☐ FROM especifica a imagem base que será usada.

Pergunta 14

2,67 / 2,67 pts

São os estágios básicos de todo pipeline:

☐ Lançamento: estágio em que a aplicação é enviada ao repositório.

☒ Todas as alternativas anteriores estão CORRETAS.

☐ Teste: estágio em que o código é testado. O uso da automação neste estágio poupa tempo e esforços.

☐ Compilação: estágio onde é feito o build da aplicação.

Pergunta 15

2,62 / 2,62 pts

São algumas ferramentas de automação de pipelines:

☒ Jenkins, Azure DevOps, Circle CI e Github Actions.

☐ Jenkins, TravisCI, Azure DevOps, Github Actions, Circle CI e Amazon S3.



Jenkins, Github Actions, OWASP ZAP, Gitlab, Circle CI e Google Cloud Build.



Jenkins, Github Actions, OWASP ZAP, Gitlab, Circle CI e Amazon S3.

Pontuação do teste: **34,66** de 40