

UNIOPET

**DESENVOLVIMENTO DE COMPONENTES DISTRIBUÍDOS –
TDS NOITE**

ARQUITETURA REST

Aluna: Marcela Arlete Rosa
André Schwab
Rafael Koinski
Mateus Menegacio

Curitiba
2021

Questionário - Parte Teórica:

1. O que é integração contínua em um projeto de software?

A integração contínua é a integração do código ao projeto principal a cada alteração ou implementação de funcionalidades novas, esse termo teve origem na metodologia ágil XP e os principais objetivos da integração contínua é checar se as alterações/novas funcionalidades não deram origem a defeitos novos no projeto que já existia, encontrar e investigar bugs mais rápido, melhorar a qualidade do software e reduzir o tempo de validação e lançamento de novas atualizações de software.

2. Quais ferramentas podem ser usadas para verificar débito técnico?

- SonarQube – Ferramenta de inspeção e revisão contínua, onde a cada build é possível fazer uma inspeção do código e verificar se o desempenho está bom ou ruim.
- Matriz de Débitos Técnicos – Ferramenta que auxilia a avaliar cada débito, ajuda na identificação e priorização de quais débitos podem ser absorvidos e aqueles cujo risco será dividido com o Product Owner, fazer a estratégia de medição das formas mais usadas pelo time para adquirir débitos.
- OutSystems – Oferece um ambiente baseado em IA, que pode agilizar a construção de códigos e em eventuais mudanças futuras, onde o código poderá ser reescrito, sem ser quebrado, quando houver nova alteração pelo desenvolvedor.

3. Quais ferramentas podem ser usadas para realizar a integração contínua?

- Jenkins - Uma das principais ferramentas para integração continua, cuja vantagem é o código aberto e modular ao mesmo tempo, o Jenkins consegue ser instalado em qualquer tipo de sistema operacional se este já tenha o Java executando.
- Travis - Ferramenta de código aberto e pode ser usada em projetos que estiverem no GitHub, utiliza os arquivos travis.yml que precisam ter informações, uma vez que a ferramenta consegue suportar uma grande variedade linguagens.
- GitlabCI - Uma ferramenta com grande eficiência, que provê suporte para a série continuous (integration, deployment e delivery), o principal objetivo é em relação ao commit onde é necessário que haja a

integração com o repositório centra, para cada commit, o GitlabCI roda um pipeline de integração passando por estágios que varia de acordo com o projeto.

- Hudson - Uma das principais vantagens é que permite realizar testes durante a integração contínua. É conhecido como um servidor de testes e pode receber vários projetos, tanto de servidor interno quanto externo, que vão fazer build de todos esses projetos que estiverem sendo realizados, também é possível que vários desenvolvedores utilizem o mesmo servidor, que se trabalhe em tarefas distintas, fazer agendamento de construções e tarefas, que poder ser automatizadas
- CruiseControl - Uma das ferramentas mais antigas lançada como um tipo de quadro cuja ideia era conseguir controlar builds. Tem toda a sua base em Java e ainda existe a possibilidade de implementações em .net e rubi.
- Bamboo - Uma ferramenta eficiente com uma parte é gratuita e outra onde as organizações comerciais são cobradas de acordo com o número de builds necessários. Faz a utilização de vários plug-ins para personalização de uso
- Team City - A principal função é permitir a execução de testes automáticos. Se existirem falhas, será mais fácil perceber o os erros e realizar a correção pelo desenvolvedor. É possível antecipar a visão daquilo que dará certo ou não. Se houver erros, a integração é cancelada e se estiver certo é liberada para uso, o que poupa o retrabalho e melhora os resultados.

4. O que é DevOps?

É um termo definido como um grupo de práticas para integração entre determinadas equipes onde adota – se processos automatizados para produção segura e rápida de aplicações e serviços, que propõe a valorização da diversidade de atividades, profissionais envolvidos e atitudes colaborativas, que possibilite o desenvolvimento ágil de apps num modelo de gestão de infraestrutura definido sob regras estabelecidas.

5. Qual o relacionamento da integração contínua com o conceito de DevOps?

A integração contínua faz parte de um conjunto de recursos que aumenta a entrega de software e o desempenho organizacional e esses recursos foram descobertos pelo programa de pesquisa DORA State of DevOps uma investigação acadêmica independente com critérios altos a respeito de práticas e recursos que podem impulsionar um maior desempenho. Com a integração contínua, novas alterações no código de uma aplicação são criadas, testadas e mescladas em um repositório compartilhado, ideal para evitar conflitos de ramificações onde várias aplicações são desenvolvidas simultaneamente e a

cultura DevOps tem como um de seus pilares a integração Contínua com transferência fácil de informação e experiências entre áreas de desenvolvimento, apoio e operações, implantação contínua com liberação rápida e contínua de novas versões de software/serviços e também do feedback contínuo das equipes envolvidas em todas as fases do ciclo de vida do software/serviço.

Parte Prática:

Como atividade da semana do nosso projeto da disciplina, cada grupo deve criar uma conta no GITHUB.

1. Nesta conta deve ser criado um projeto com o nome DesenvCompDist.
2. Dentro deste projeto deve ser criada uma pasta chamada Trabalhos.
3. Dentro da pasta Trabalhos devem ser subidos todas as entregas até agora do nosso projeto da disciplina. Ou seja, as entregas feitas na aula da semana 4 e a entrega de hoje (parte teórica)

Link GitHub:

<https://github.com/schwabandre/DesenvCompDist>