

Blockchain e Smart Contracts

Engenharia de Software

Jesiel Viana

jesiel@ifpi.edu.br

Descrição:

O armazenamento de dados de forma segura é uma preocupação em diversas áreas da sociedade. Os sistemas de armazenamento de dados que utilizam o padrão Blockchain surgem como alternativas para garantir nativamente proteção contra alteração de informações, proporcionando a centralização das informações, descentralização do armazenamento, auditabilidade e transparência no processo de armazenamento.

Segundo CROSBY (2016,), blockchain é essencialmente um banco de dados distribuído de registros (cadeia de blocos) de todas as transações ou eventos digitais que foram executados e compartilhados entre as partes participantes. Cada transação registrado na rede é verificada por consenso da maioria dos participantes no sistema. Uma vez inserida, a informação nunca pode ser apagada. O blockchain contém um registro certo e verificável de cada transação já feita. Tornou-se popular graças

ao Bitcoin [NAKAMOTO 2008]. No Blockchain os blocos gerados são incluídos de forma linear, e cronológica, mantendo uma cópia completa dos dados em cada nó na sua cadeia, que é atualizada automaticamente quando um novo nó é inserido no sistema [NAKAMOTO 2008].

Contratos inteligentes (Smarts Contracts em Inglês) são basicamente programas de computador que podem executar automaticamente os termos de um contrato. Quando uma condição pré-configurada em um contrato inteligente entre as entidades participantes é atendida, as partes envolvidas em um acordo contratual podem automaticamente efetuar pagamentos de acordo com o contrato de maneira transparente.

Os contratos inteligentes Ethereum são predominantemente escritos em uma linguagem chamada Solidity. Um contrato inteligente é implementado na blockchain da Ethereum na forma de uma transação por um remetente, onde um endereço é atribuído ao contrato. Cada contrato inteligente contém um estado e código executável. Uma vez implantado, um contrato inteligente é imutável e nenhuma modificação pode ser feita no contrato. No entanto, pode ser morto se uma instrução de auto destruição no contrato for executada [TANN, A. ET AL, 2018].

Dentro deste contexto as sugestões para projetos/pesquisas são:

- 1. Utilizar uma blockchain que suporta contratos inteligentes ou desenvolver uma blockchain própria para ligar smartphones aos seus respectivos donos. Através dessa blockchain seria possível validar a**

propriedade de um smartphone, possibilitando ao dono rastreá-lo e transferir para outra pessoa.

2. Utilizar uma blockchain que suporta contratos inteligentes ou desenvolver uma blockchain própria para gestão de contratos de aluguéis de imóveis;
3. Qualquer tema que tenha como foco/utilização Blockchain.

Referências:

- CROSBY, Michael et al. Blockchain technology: Beyond bitcoin. **Applied Innovation**, v. 2, n. 6-10, p. 71, 2016.
- NAKAMOTO, Satoshi. Bitcoin: a peer-to-peer electronic cash system (2008). 2008.
- TANN, A. et al. Towards Safer Smart Contracts: A Sequence Learning Approach to Detecting Vulnerabilities. **arXiv preprint arXiv:1811.06632**, 2018.
- <https://www.udemy.com/blockchain-introducao/> (Curso Gratuito de Introdução à Blockchain)
- <https://www.udemy.com/contratos-inteligentes/> (Curso Gratuito de Introdução à Smart Contract Ethereum)

DevOps
Engenharia de Software
Jesiel Viana
jesiel@ifpi.edu.br

Descrição:

O DevOps é um conjunto de práticas que automatizam os processos entre equipes de desenvolvimento de software e de TI para que possam criar, testar e liberar softwares de maneira mais rápida e confiável. O movimento DevOps pretende melhorar a comunicação, a colaboração e a integração entre desenvolvedores de software (Dev) e profissionais de operações de TI (Ops). A automação é a chave para o sucesso do DevOps: criação automatizada de sistemas a partir de repositórios de gerenciamento de versão; execução automatizada de testes unitários, integração e testes de sistema; implantação automatizada em ambientes de teste e produção; incluindo benchmarks de desempenho.

O DevOps obteve sucesso significativo no setor industrial, mas ainda recebeu pouca atenção no ensino superior.

Christensen (2016) descreveu seus desafios para o ensino de DevOps e propôs técnicas de ensino voltadas para o DevOps. Ensinar o DevOps pode ser desafiador pelos seguintes motivos:

- Experiência limitada em Suporte de Operações do corpo docente que ensina DevOps;
- A maioria dos cursos universitários está focada em desenvolvimento ou operações, mas não em ambos os tópicos;
- O DevOps é focado na competência de habilidades práticas, como codificação, banco de dados configuração, etc.
- Criar um ambiente de produção realista para testar aplicativos pode ser uma tarefa difícil por causa de restrições financeiras em muitas universidades
- A avaliação do trabalho do aluno pode ser difícil sem testes reais em um ambiente de produção, especialmente ajuste de desempenho para transações de alto volume

Com base nesses desafios, Christensen (2016) desenvolveu uma Aprendizagem Cognitiva e Abordagem Histórica para ensinar DevOps dentro de um curso de computação em nuvem que é ministrado na Universidade de Aarhus, na Dinamarca.

- Os alunos projetaram e codificaram um sistema distribuído simples chamado “SkyCave”
- Os resultados do aluno incluíram a documentação e o código para o aplicativo;
- As notas foram baseadas na quantidade e qualidade do trabalho submetido;

- Não foram cumpridos prazos específicos, exceto para o prazo final do curso;
- O feedback para os exercícios de aula foi completado com 24 horas do aluno submissão.

Dentro deste contexto as sugestões para projetos/pesquisas são:

1. **Avaliar a prática de DevOps na Fábrica Escola de Software Mambee dentro do processo de aprendizagem dos alunos de ADS.**
2. **Impacto do DevOps na produtividade de equipes iniciantes em desenvolvimento de software;**
3. **Implantação de DevOps em equipes com pouca experiência em desenvolvimento de software;**
4. **Qualquer tema que tenha como foco/utilização DevOps.**

Referências

1. BOBROV, Evgeny et al. Teaching DevOps in academia and industry: reflections and vision. **arXiv preprint arXiv:1903.07468**, 2019.
2. MASON, Robert; MASTERS, William; STARK, Alan. Teaching Agile Development with DevOps in a Software Engineering and Database Technologies Practicum. In: Proceedings of the 3rd International Conference on Higher Education Advances. Editorial Universitat Politècnica de València, 2017. p. 1353-1362.
3. MOSSER, Sébastien et al. Teaching DevOps at the Graduate Level: A report from Polytech Nice Sophia. 2018.

Outras referências:

1. <https://aws.amazon.com/pt/devops/what-is-devops/>
2. <https://www.redhat.com/pt-br/topics/devops>
3. <https://br.atlassian.com/devops>

Descrição:

Microserviços é a última tendência em design, desenvolvimento e entrega de serviços de software [1]. Microserviços são utilizados para desenvolver uma aplicação como um conjunto de pequenos serviços autônomos que trabalham juntos, cada um funcionando em seu próprio processo [2]. Esses serviços são construídos através de pequenas responsabilidades e publicados em produção de maneira independente, através de processos de implantação automatizados.

Martin Fowler define Microserviço como uma abordagem para desenvolver uma única aplicação como uma suíte de serviços, cada um rodando em seu próprio processo e se comunicando através de mecanismos leves, geralmente através de uma API HTTP. Estes serviços são construídos através de pequenas responsabilidades e publicados em produção de maneira independente através de processos de deploys automatizados. Existe um gerenciamento centralizado mínimo destes serviços, que podem ser escritos em diferentes linguagens e usarem diferentes tecnologias para armazenamento de dados.[3]

No entanto Microserviços possuem alguns desafios e problemas, tais como aumento da complexidade de implantação e manutenção de vários serviços, além do monitoramento constante e elevação dos custos operacionais.

Dentro deste contexto as sugestões para projetos/pesquisas é investigar alguns algum dos tópicos:

1. **Decomposição dos microserviços (Ver Self-Contained Systems [4]);**
2. **Acoplamento entre os serviços e consistência dos dados;**
3. **Qualquer tema que tenha como foco/utilização de microserviço.**

Referências:

1. O. Zimmermann, "Microservices Tenets: Agile Approach to Service Development and Deployment," Computer Science—Research and Development, vol. 32, nos. 3–4, 2017, pp. 301–310.
2. S. Newman, Building microservices: designing fine-grained systems. O'Reilly Media, Inc., 2015.
3. <https://martinfowler.com/articles/microservices.html>
4. <https://scs-architecture.org/>

Outras referências:

1. JAMSHIDI, Pooyan et al. Microservices: The journey so far and challenges ahead. **IEEE Software**, v. 35, n. 3, p. 24-35, 2018.