Scripts de Instalação de uma Rede *Blockchain* como Recurso Didático para Metodologias Ativas de Ensino de Computação

Flávio Fernandes de Melo
Universidade Federal do Tocantins
Palmas, Tocantins, Brasil
meloflavio@uft.edu.br

Carlos Eduardo Alves Cavalcante
Universidade Federal do Tocantins
Palmas, Tocantins, Brasil
carlosalves@uft.edu.br

Patrick Letouze Moreira
Universidade Federal do Tocantins
Palmas, Tocantins, Brasil
letouze@uft.edu.br

RESUMO

Neste trabalho é proposto *Scripts* de instalação de uma rede *Blockchain* como recurso didático para o uso de metodologias ativas de aprendizagem com práticas *hands-on* no ensino de computação. O problema inicial proposto consiste na criação e instalação de uma rede *blockchain* privada. A intenção é disponibilizar um recurso didático que apoie as práticas no ensino de computação em relação a *blockchain* e as disciplinas que utilizem conceitos relacionados a essa tecnologia. Neste intuito, elaborou-se um *script* para a criação e configuração de uma rede *blockchain*, que juntamente com um roteiro de orientação compõem o material didático utilizado para introduzir os conceitos e fundamentos da tecnologia *blockchain* ao mesmo tempo que pode ser utilizado para demonstrar a criação e instalação real de uma rede. Este material pode ser facilmente adaptado para estimular os estudos além da tecnologia *Blockchain*, por exemplo, pode ser adaptado para as disciplinas de Introdução à Computação, Introdução à Programação, Algoritmos e Programação, Sistemas Operacionais, Banco de Dados, Redes de Computadores, Segurança em Tecnologia da Informação entre outras.

PALAVRAS-CHAVE

Blockchain, Metodologias Ativas de Aprendizagem, *Script*

1 INTRODUÇÃO

A tecnologia *blockchain* foi apresentada inicialmente por Nakamoto [15] ao descrever uma moeda inteiramente digital, o que permitiria enviar pagamentos online diretamente de uma pessoa para outra, sem a necessidade de passar por uma instituição financeira. A tecnologia rapidamente se popularizou com a criação da criptomoeda Bitcoin, que teve seu bloco inicial criado no início de 2009 e, desde então, expandiu-se em uma escala sem precedentes.

Apesar de seu foco inicial na criação de criptomoedas, Abdellatif [1] afirma que cada vez mais setores, como governos, finanças, saúde, indústrias em geral e entre outros buscam novas possibilidades de uso para esta promissora tecnologia. Grande parte do interesse sobre *blockchain* baseia-se em suas propriedades básicas, que prometem alta segurança, confiabilidade e disponibilidade dos dados contidos em sua rede, além de promover a descentralização no controle de suas transações.

Com o alto interesse na tecnologia, não demorou muito para que surgissem diversos projetos que a explorasse para além das criptomoedas, por exemplo, Cheng et al. [6], descreve um sistema para o reconhecimento de diplomas de graduação utilizando *blockchain*. Notheisen at al. [17] por sua vez, demonstra a utilização da tecnologia em um sistema para o gerenciamento de ativos do mundo real, como casas e carros. Brave (2019) desenvolveu um navegador com a possibilidade de recompensar os usuários e criadores de conteúdo utilizando uma *blockchain*. Em outros exemplos Souza Junior et al. [25] descreve a utilização de *blockchain* para um sistema internacional de acreditação de profissionais de saúde e Letouze et al.[13] descreve um sistema baseado em *blockchain* para a negociação de precatórios no Brasil.

As possibilidades de uso para a tecnologia *blockchain* são as mais diversas e a perspectiva de evolução e impacto da tecnologia são muito grandes. No entanto, Oliveira e Freitas [18] consideram insuficientes a quantidade de estudos realizados na área até então, o que segundo os autores dificulta a identificação de como ela poderá realmente afetar a sociedade de uma forma mais abrangente, assim necessitando de um maior número de pesquisas sobre o assunto.

Uma das grandes dificuldades na disseminação e utilização de *blockchain* segundo Bornelus, Chi e Shahriar [5] é a considerável curva de aprendizado da tecnologia, uma vez que os fundamentos científicos e computacionais por trás da tecnologia envolvem conhecimentos de múltiplas disciplinas, o que dificulta sua compreensão por pessoas que não estão familiarizados com estes fundamentos. Diante dessas dificuldades, existem trabalhos que ajudam a difundir o conhecimento dessa tecnologia, como o material de apoio produzido pelo Tribunal de Contas da União [26], elaborado em forma de sumário executivo para auxiliar gestores públicos a avaliar a pertinência do projeto *blockchain* de suas organizações, apresentando a experiência de outras organizações no Brasil e no mundo.

Fomentar o estudo de *blockchain* em sala de aula em cursos de tecnologia é uma alternativa para incentivar novos projetos na área, além de promover a utilização dos conceitos de diversas disciplinas da computação. Porém vale ressaltar que a simples apresentação de conceitos em aulas teóricas pode não ser suficiente, pois como mencionado por Gavaza, Salvador e Do Santos [11], uma disciplina que trata de tópicos que possuem um alto nível de abstração exige bastante esforço dos alunos para sua compreensão.

Para facilitar o aprendizado de assuntos com um alto grau de abstração, como a tecnologia *blockchain*, é preciso buscar alternativas ao ensino tradicional baseada apenas na exposição teórica de seus conceitos. Pinto et al. [8] afirma que para isso é necessário

EduComp'21, Abril 27–30, 2021, Jataí, Goiás, Brasil (On-line)

lançar mão de metodologias que busquem envolver mais o aluno no processo de aprendizagem, assim permitindo uma maior relação dos conhecimentos aprendidos em aula com sua utilização prática no mundo real. Neste contexto identifica-se a hipótese de utilização de metodologias ativas de aprendizagem, que contribuem para maior interação dos professores e alunos, permitindo a construção ativa e colaborativa dos conhecimentos.

Entre as metodologias ativas de aprendizado pode-se destacar o Aprendizado Baseado em Problemas (ABP) como metodologia para o ensino de disciplinas complexas. Nessa metodologia um problema é proposto para os estudantes, a solução prática é construída colaborativamente pelos alunos com a supervisão do professor. Neste âmbito, temos os exemplos de Silva et al. [24] que descreve a utilização de ABP para o ensino de urgência e emergência na enfermagem, um estudo realizado na Universidade Federal do Pará. Um outro exemplo de utilização de ABP é o trabalho de Rodrigues e Araújo [9] que relata a utilização da metodologia no ensino das disciplinas de contabilidade de uma universidade particular.

Outras abordagens de aprendizagem ativa também podem ser utilizadas, como o trabalho de Du [10] o qual descreve a utilização de exercícios laboratoriais práticos para o ensino de segurança na computação. Rao e Dave [20] por sua vez, apresenta a utilização de exercícios práticos para o ensino de novas tecnologias como Internet das Coisas (em inglês: Internet of Things, IoT) e *blockhcain*. A abordagem descrita por estes autores é conhecida como aprendizado *hands-on*, onde os estudantes são apresentados aos conceitos teóricos e logo em seguida são levados a aplicar os conhecimentos em exercícios práticos. A abordagem *hands-on* de aprendizado pode ser facilmente adaptada e integrada ao ABP, utilizando os exercícios práticos para auxiliar na resolução de um determinado problema, ao passo que constrói gradualmente os conhecimentos necessários.

Neste trabalho propomos a utilização de um *script* - arquivo com conjunto de comandos executados por um interpretador (COSTA, 2010) [7], como um recurso didático *hands-on* no ensino da computação, produto de uma abordagem ABP. O *script* utilizado foi desenvolvido para a criação e configuração automática de uma rede *blockchain* privada, o que permite introduzir os conceitos necessários enquanto realiza-se a demonstração prática da tecnologia. O mesmo *script* pode ser utilizado para o ensino de diversas disciplinas da grade curricular em um curso de ciência da computação apenas alterando o foco da apresentação dos conceitos, uma vez que os fundamentos da tecnologia *blockchain* são compostos de conceitos básicos de várias destas disciplinas, como Introdução à Computação, Introdução à Programação, Algoritmos e Programação, Sistemas Operacionais, Banco de Dados, Redes de Computadores, Segurança em Tecnologia da Informação entre outras.

2 FUNDAMENTOS

Nesta seção são descritos alguns dos fundamentos utilizados no trabalho, como *Blockchain*, Metodologias Ativas de Aprendizagem e Aprendizagem Baseada em Problemas.

2.1 *Blockchain*

Segundo Nakamoto [15], a tecnologia *Blockchain* funciona como um tipo de livro razão distribuído, com recurso de imutabilidade entre os nós em uma rede *peer-to-peer* baseado em um protocolo de consenso. Cada nó pode manter a mesma razão sem uma autoridade centralizada utilizando *hashes* criptográficos e assinaturas digitais garantindo a integridade das transações em cada bloco.

Quanto a estrutura do *Blockchain*, esta é construída por blocos ligados por uma lista encadeada, de forma que cada bloco contenha a referência do seu antecessor, garantindo assim que a modificação de informações gravadas em cada bloco exija um grande poder computacional, tornando essa ação computacionalmente impraticável em grandes redes.

Antonopoulos [2] descreve um bloco sendo composto por um identificador (*block hash*), definido pela dupla aplicação do algoritmo SHA-256 em seu cabeçalho, o *block hash* do bloco anterior, o conjunto de todas as transações, juntamente com um conjunto de informações que compõem seu cabeçalho.

A estrutura do cabeçalho pode ser dividida em três conjuntos de dados de acordo com o seu propósito. O primeiro chamado de *Previous Block Hash*, composto com o *hash* do bloco anterior, garante a conexão entre todos os blocos da *Blockchain*. O segundo é campo *Merkle Root*, usado para resumir de maneira eficiente o conjunto de transações do bloco. Por fim, o conjunto dos campos *timestamp*, *difficulty target*, e *nonce* são referentes ao processo de mineração, representando respectivamente, hora aproximada da criação do bloco, dificuldade alvo do algoritmo utilizada no bloco e o contador utilizado pelo algoritmo.

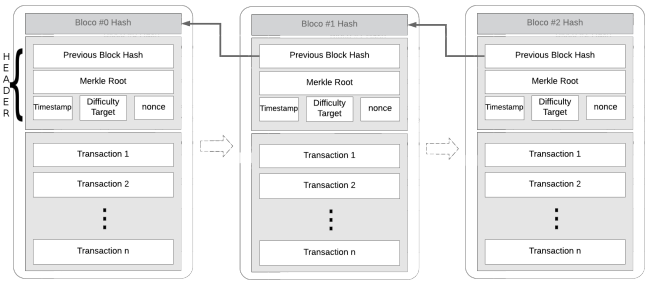

Figura 1: Blocos encadeados. Fonte: Adaptado de Antonopoulos [2] .

A Figura 1 mostra um exemplo da estrutura dos três conjuntos de dados do bloco e do encadeamento entre eles, sendo comum para a identificação do bloco, além do *hash* duplo criado pela criptografia, o número da posição em que ele se encontra na *Blockchain*. Sobre as propriedades inerentes aos conceitos confiabilidade e segurança da tecnologia de *blockchain*, Iansiti e Lakhani [12] descrevem cinco princípios básicos, os quais seguem listados abaixo:

- **Banco de dados distribuído:** cada parte em um *blockchain* tem acesso a toda base dados e ao seu completo histórico de transações sem a necessidade de intermediários, no entanto, ninguém pode alterar seus registros individualmente.
- **Transmissão ponto-a-ponto:** a comunicação ocorre diretamente entre os pontos, em vez de serem realizadas de forma centralizada, cada ponto armazena e encaminha as informações aos demais participantes.
- **Transparência e pseudoanonimato:** cada transação e os valores associados são disponibilizadas a qualquer usuário com acesso ao sistema. No entanto, cada nó, ou usuário, em

um *blockchain* tem um "endereço" alfanumérico único que o identifica. Um usuário pode escolher se manter anônimo ou compartilhar provas de identidade com os outros. As transações ocorrem entre estes "endereços" no *blockchain*.
- **Irreversibilidade de registros:** uma vez realizada uma transação e esta transação adicionada ao *blockchain*, os registros não podem ser alterados, uma vez que as propriedades do *blockchain* garantem que cada registro esteja relacionado a todos os registros adicionados antes dele.
- **Lógica computacional:** a natureza digital dos registros significa que as transações de *blockchain* podem ser vinculadas à uma lógica computacional e, em essência, programadas. Assim, os usuários podem determinar algoritmos e regras que vinculam automaticamente transações entre nós.

A introdução de *Smart Contracts*, que funcionam como "um contrato digital que é escrito em código-fonte e executado por computadores, que integra o mecanismo à prova de adulteração de *Blockchain*" (LIN, 2017) [14], propiciou maiores níveis de programabilidade para a tecnologia. A utilização de redes *blockchain* que dispõem desses recursos são ideais para aplicações em novas áreas que diferem de seu foco original das criptomoedas.

Visando estes conceitos e utilizando a linguagem de criação de códigos para automatização de tarefas Shell *Script*, foram desenvolvidos arquivos (*scripts*) contendo instruções que ao serem executadas criam e configuram uma rede *blockchain* privada. Vale ressaltar que estes *scripts* foram escritos para que possam ser executados em máquinas com os sistemas operacionais Windows, Linux e MacOS. Utilizamos para os testes as seguintes versões:

- Windows 7 e 10
- Ubuntu 18.04, Ubuntu 19.04 e Debian 9
- MacOS 10.15

2.2 Metodologias Ativas de Aprendizagem

As Metodologias ativas de aprendizagem colocam o aluno como centro do processo de ensino. Conforme Barbosa e Moura em [3], nessas metodologias a aprendizagem ocorre quando o aluno interage com o assunto em estudo das mais diversas formas, como falando, ouvindo, discutindo ou fazendo. O aluno deixa de ser um receptor passivo das informações limitado a memorizar o conteúdo para ele apresentado e torna-se um colaborador ativo do processo de aprendizado, despertando seu pensamento crítico. Neste sentido, Rocha e Lemos em [21] afirmam que nestas metodologias o conhecimento é construído pela interação dos alunos, professores e o ambiente, reforçando a participação do aluno como fundamental para a construção dos conhecimentos.

2.2.1 Aprendizado Baseado em Problemas. De acordo com Savery [22], a Aprendizagem Baseada em Problemas (ABP) é uma abordagem instrucional e curricular centrada no aluno que permite que eles conduzam pesquisas, integrem teoria e prática e apliquem conhecimentos e habilidades para desenvolver uma solução viável para um problema pré definido. Orey [19] afirma que em cursos acadêmicos, a ABP é usada como uma ferramenta para ajudar os alunos a compreender a utilidade de um determinado conceito ou estudo.

Segundo Silva et al. [24] nessa metodologia para solucionar o problema apresentado, os alunos devem recorrer aos sete passos do ABP, que são:

- Esclarecer termos e conceitos desconhecidos;
- Definir o problema;
- Analisar o problema baseado em conhecimentos prévios;
- Resumir as conclusões;
- Formular metas de estudo;
- Auto-aprendizado;
- Dividir conhecimentos com o grupo;

O aprendizado nessa abordagem não se limita apenas aos conhecimentos adquiridos, mas também no processo que foi empregado. Dessa forma, o aluno não só aprende resolver o problema proposto, mas como lidar com novas dificuldades que a ele serão apresentadas. Neste sentido Orey [19] afirma que a metodologia ABP é frequentemente abordada em um ambiente de equipe com ênfase na construção de habilidades relacionadas à tomada de decisão consensual, diálogo e discussão, manutenção da equipe, gestão de conflitos e liderança de equipe.

Senna e Lopes [23] ressaltam que a expressão Aprendizagem Baseada em Projeto surge, às vezes, como sinônimo de Aprendizagem Baseada em Problema, por aparecerem na língua inglesa como Project Based Learning e Problem Based Learning utilizando a mesma sigla – PBL, ou as vezes PjBL para o primeiro e PBL para o segundo, e mesmo que o desenvolvimento de um projeto possa ocorrer com a resolução de problemas, uma prática tem como foco o problema, e a outra, o projeto.

De acordo com Bender em [4], a Aprendizagem Baseada em Projetos é uma metodologia de ensino baseada no fato de os alunos confrontarem questões e problemas do mundo real que eles consideram significativos, determinar como abordá-los e, então, agir de forma colaborativa para criar soluções de problemas.

Neste trabalho foi proposto inicialmente um problema, a criação automatizada de uma rede privada de *blockchain*, que servirá como base para um projeto de mestrado e dado a proximidade das duas abordagens de aprendizado a metodologia de Aprendizagem Baseada em Projeto também foi utilizada.

Bender em [4] apresenta como base ou essencial para uma abordagem de Aprendizagem Baseada em Projeto as seguintes palavras ou conceitos:

- **Âncora:** a base para fazer a pergunta que serve para fundamentar a instrução em um cenário do mundo real.
- **Artefatos:** os itens que representam soluções possíveis para o problema ou aspectos da solução do problema, cenários de dramatização são incluídos.
- **Realização autêntica:** representa a ênfase, o tipo de coisas que os profissionais podem esperar fazer na vida real.
- **Debate:** este é um processo pelo qual os alunos passam para formular um plano para as tarefas do projeto.
- **Pergunta de direcionamento:** a pergunta principal que fornece o objetivo geral do projeto.
- **Voz e escolha do aluno:** representa que os alunos devem ter uma palavra a dizer na seleção do projeto e na formulação da questão essencial.

Levando em consideração esses conceitos e que este trabalho representando a primeira fase de um projeto apresenta-se o cenário mostrado na Tabela 1.

Tabela 1: Cenário ABP para fase 1 do projeto

Cenário ABP para Fase 1: Automatização da criação da rede *Blockchain*	
Âncora	O problema apresentado aos alunos deve ser a necessidade da automatização do processo de criação e preparação de uma rede *blockchain* privada que possa ser integrada a um sistema web já existente escrito na linguagem JAVA.
Artefatos	Um script que possa ser executado em diferentes sistemas operacionais e um roteiro de como utilizá-lo.
Realização autêntica	A rede *blockchain* em funcionamento.
Debate	Esse processo deve ser realizado em reuniões periódicas de gerenciamento de projetos.
Pergunta de direcionamento	A automatização do processo de criação e preparação de uma rede *blockchain* é um ativo valioso para integração de um sistema web em JAVA?
Voz e escolha do aluno	Os alunos devem ajudar a escolher as ferramentas e técnicas para o desenvolvimento do sistema.

3 TRABALHOS RELACIONADOS

Alguns trabalhos e abordagens para a introdução e ensino de tecnologia *blockchain* e computação podem ser encontrados na literatura, Rao e Dave [20] utilizaram uma abordagem de aprendizado baseado em exercícios de laboratório (*hands-on*) para ensinar os alunos de graduação os conceitos de IoT, computação em nuvem e também *blockchain*. O projeto consiste na criação de um sistema que deveria obter imagens, salvar registros criptografados imutáveis, transmitindo e armazenando-os na nuvem.

Os autores então dividiram o projeto prático em dois exercícios de laboratório, no primeiro os alunos deveriam realizar a captura da imagem, a transmissão e o armazenamento na nuvem. Para este primeiro exercício foi solicitado aos alunos que estudassem conceitos básicos de comandos Linux e a linguagem de programação Python, além disso foram instruídos sobre o básico da plataforma Raspberry Pi. No exercício prático os alunos então deveriam criar um código em Python para a captura de uma imagem utilizando o módulo de câmera do Raspberry Pi, posteriormente os alunos deveriam codificar a etapa de envio da imagem para uma conta criada no Google Drive.

No segundo exercício os alunos são apresentados previamente aos protocolos de segurança SHA-256, um conjunto de algoritmos de criptografia baseados em funções matemáticas *hash*. Neste exercício os alunos então deveriam converter a imagem capturada no primeiro exercício em uma cadeia de caracteres e então transformá-la em código *hash* utilizando uma biblioteca de python chamada *hashlib*, segundo os autores utilizando este exercício os alunos puderam entender o fundamento de criptografia e demonstrar a característica de imutabilidade contido na base da tecnologia *blockchain*.

Apesar das afirmações dos autores sugerirem um ensino mais abrangente de *blockchain*, no trabalho descrito apenas foi apresentado o conceito de criptografia comumente usado neste tipo de rede, tópicos como instalação, configuração e o funcionamento real da tecnologia não foram abordados pelos autores, o trabalho apresenta alguns conceitos de segurança da informação, limitando a abordagem aos conceitos de criptografia. Mesmo não sendo explicitamente abordados, conceitos de Redes, Sistemas Operacionais e Programação foram exercitados no citado trabalho.

Uma outra abordagem para o ensino de *blockchain* foi descrita por Negash e Thomas [16], neste trabalho os autores apresentaram um projeto baseados em sete cenários da indústria para transmitir conhecimentos teóricos e técnicos (práticos) de *blockchain* para um conjunto de estudantes de negócios com poucos conhecimentos técnicos. Para exemplificar quatro dos sete cenários propostos pelos autores estão descritos abaixo:

- **Educação**: neste cenário é descrito a utilização de um sistema baseado em *blockchain* para a verificação e autenticação de diplomas, as universidades registram os diplomas numa rede *blockchain* pública que permite a verificação da autenticidade de um diploma posteriormente apresentado.
- **Saúde**: o cenário descreve a possibilidade de utilização da *blockchain* para o armazenamento e controle de prontuários médicos, segundo os autores uma abordagem com *blockchain* permite que pacientes tenham o controle de seus prontuários, permitindo acesso apenas aos dados necessários para cada atendimento.
- **Aviação**: neste cenário é descrito uma oportunidade de negócios onde as passagens aéreas poderiam ser vendidas entre passageiros com o auxílio de um sistema de *blockchain*, onde um indivíduo que comprasse uma passagem poderia vendê-la para outra pessoa diretamente, registrando a transação numa *blockchain* compartilhada com as companhias aéreas.
- **Cadeia de suprimentos**: o cenário descreve a automatização do controle de estoque de empresas, para isso utiliza um sistema *blockchain* baseados em contratos inteligentes com execução semi autônoma onde um pedido de compra pode ser lançado automaticamente quando o estoque da empresa estiver num nível determinado.

Os demais cenários utilizados pelos autores incluem a descrição de sistemas das áreas de Governança, Internet das Coisas (IoT) e FinTech (finanças digitais). Para promover uma experiência significativa aos estudantes os autores projetaram interações reais para para demonstrar a aplicabilidade da tecnologia, para isso utilizaram a infraestrutura da LinuxOne Foundation (com suporte da IBM), utilizando a plataforma Hyperledge-Fabric (plataforma de desenvolvimento *blockchain*), desenvolveram práticas para demonstrar os cenários propostos.

Apesar de uma descrição básica e de alguns exemplos práticos de funcionamento da tecnologia, nesta abordagem o foco é voltado mais para a apresentação das possibilidades de uso da tecnologia *blockchain* do que propriamente para a construção dos sistemas

descritos. Além disso, esta abordagem necessita de mais recursos de infraestrutura para serem aplicadas, o que pode inviabilizar sua utilização em algumas situações.

Uma terceira abordagem para o ensino de *blockchain* é o framework apresentado por Bornelus, Chi e Shahriar (2019), neste propõe a utilização de diversos laboratórios que de forma modular apresentam todos aspectos da aplicação da tecnologia *blockchain*. A descrição dos laboratório *hands-on* está apresentada abaixo:

- **Entendendo a segurança por trás da *Blockchain*:** segundo os autores o objetivo é apresentar a criptografia por trás dessa tecnologia - são demonstrados tópicos como - árvores Merkle, criptografia de curva elíptica e SHA256.
- **Laboratório prático - Criando seu próprio cripto-sistema:** O objetivo deste laboratório é apresentar aos alunos a plataforma Ethereum, utilizando a criação de contratos inteligentes usando a linguagem Solidity e o Remix, uma ferramenta poderosa para escrita de contratos diretamente no navegador.
- **Passado, presente e futuro:** O objetivo deste tópico é demonstrar os aplicativos de *blockchain* da vida real: são demonstrados exemplos como Bitcoin, AWS Quantum Ledger Database, Azure MS *Blockchain*, IBM Hyperledger, e a perspectiva de utilizações futuras da tecnologia *blockchain* como o Block-Lattice.
- **Laboratório prático dApps:** O objetivo deste laboratório é aumentar a capacidade de desenvolvimento do aluno, criando um aplicativo descentralizado (d-Apps), para isso são utilizadas ferramentas como Solidity, Ethereum, Truffle, Ganache, Meta Maks entre outros.

A representação gráfica do framework com o conteúdo completo de cada laboratório é apresentada na Figura 2.

Entendendo a segurança por trás da *Blockchain*	- SHA256 - Árvore Merkle - Curva Elíptica - Chaves Pública-Privada
Laboratório Prático: Crie Seu próprio cripto-sistema	- Criando seu próprio cripto-sistema parte 1: Usando Solidity, Remix na plataforma Ethereum - Vários Artigos e eventos atuais sobre desenvolvimento *blockchain*
Passado, Presente e Futuro do dessenvolvimento *Blockchain*	- Bitcoin e outras criptomoedas - Desenvolvimento de aplicações Ethereum - Block-Lattice - Vários artigos e eventos atuais sobre desenvolvimento *blockchain*
Laboratório Prático: dApp cripto-sistema	- Crie seu próprio cripto-sistema parte 2: Usando Ethereum, código aberto para criar seu ambiente local de desenvolvimento com Truffle e Ganache para lançar dApps

Figura 2: Conteúdos dos laboratórios *hands-on*, Adaptado de Bornelus, Chi e Shahriar [5] .

Nesta abordagem a tecnologia *blockchain* é ensinada de forma bastante robusta e avançada, todos os conceitos são apresentados de forma teórica e em sequência são realizadas as atividades práticas para fixação dos conhecimentos apresentados. No entanto é necessário por parte dos alunos um nível mais avançado de conhecimentos teóricos fundamentais, nesta abordagem os professores constroem toda a base teórica para depois utilizarem os laboratórios para as práticas ensinadas, numa abordagem que utiliza a exposição tradicional do conhecimento com atividades mais práticas. Neste trabalho os alunos devem de antemão terem determinado domínio sobre outras disciplinas de computação, sendo trabalhados conceitos mais avançados nos laboratórios sugeridos pelos autores.

4 METODOLOGIA DESENVOLVIMENTO DOS SCRIPTS

Inicialmente a necessidade de criação de um *script* para inicialização e configuração de uma rede *blockchain* surgiu em um projeto para o desenvolvimento de um sistema, no entanto logo percebeu-se a possibilidade de uso deste *script* como recurso didático, uma vez que diversos conceitos da computação tiveram que ser estudados para sua criação. Dentre as restrições impostas pelo projeto de origem estavam a necessidade de código aberto, suporte à *smart contracts* e a compatibilidade da rede com a linguagem de programação JAVA. Desse modo, o primeiro passo para o desenvolvimento dos *scripts* foi a definição da plataforma *blockchain* a ser utilizada. Foram analisadas as redes Bitcoin, Ethereum, Hyperledger Fabric, Quorum, EOS e R3 Corda.

Na tabela na Figura 2 segue um *benchmark* com algumas características levantadas para a escolha da plataforma deste projeto dentre elas: proposta da plataforma, tipo de rede se permite ou não a participação de partes sem ser previamente autorizadas, protocolos de consenso, interfaces de programação de aplicações (em inglês: Application Programming Interface - API) disponíveis e o suporte para *Smart Contracts*.

A Ethereum *Blockchain* foi escolhida por garantir as restrições mencionadas e após as comparações notou-se que a possibilidade de criar uma rede não permissionada seria a ideal para atingir objetivos futuros do projeto inicial, já que este tipo de rede é projetada para permitir a participação pública (por exemplo, alguns aplicativos que dependem de dados gerenciados pelos usuários).

Com a plataforma escolhida a próxima questão a ser resolvida foi a escolha da forma de instalação que posteriormente deveria ser automatizada. Foram identificadas três formas distintas para a instalação da rede *blockchain* da Ethereum:

- através de sistemas de gerenciamento de pacotes;
- através da compilação de códigos fontes e;
- através de download de arquivo binário já compilado.

No primeiro caso, os sistema de gerenciamento de pacotes do Linux e do MacOS podem auxiliar na instalação da rede Ethereum, precisamos para isso, adicionar um repositório PPA no caso do Linux ou instalar o Homebrew no caso do MacOS, sendo que para o sistema da microsoft esta forma de instalação não está disponível. A problemática deste modo ficaria a cargo de seguir tutoriais desatualizados do Ethereum que poderiam indicar versões não mais suportadas em sistemas operacionais mais recentes, devendo fazer a correção das versões manualmente à medida que forem identificadas versões não mais existentes ou incompatíveis com dependências instaladas.

Para o segundo modo, algumas dependências são requeridas, sendo necessário baixá-las antes de se iniciar o processo de instalação. Aqui novamente, podemos ter problemas quanto a versão das dependências e do sistema operacional da máquina, o que no futuro poderia ser um complicador quanto a utilização das mesmas

Tabela 2: Benchmark das plataformas *blockchain*

	Bitcoin	Ethereum	Hyperledger	Quorum	EOS	R3 Corda
Principal uso	Criptomoeda	Plataforma genérica de *blockchain*	*Blockchain* voltado para empresas	Para aplicativos que requerem alto nível de privacidade.	Criar uma plataforma escalável para dapps em escala industrial	Plataforma especializada para a indústria financeira (ativos digitais)
Tipo de Rede	Não permissionada	Não permissionada ou permissionada	permissionada	Permissionada	Permissionada	Permissionada
Consenso	PoW	PoW, PoS	Kafka, PoET, BFT	QuorumChain, RAFT(baseado)	DPOS	RAFT, BFT
Smart Contracts	Limitado	Sim	Sim	Sim	Sim	Sim
APIs	bitcoin-cli (RPC)	Java, Python, Javascript,Go, Rust, .NET, Delphi	CLI, REST, Java e Node.js	Ferramentas familiares da Ethereum	Javascript, Swift, Java	Kotlin, Java
Possui Código Aberto	Sim	Sim	Sim	Sim	Sim	Sim

dependências utilizadas em um tutorial já que estas poderiam apresentar depreciação e incompatibilidade ao passo que estas forem sendo atualizadas.

O último meio de instalação é através do download de arquivo binário, deve-se baixar o arquivo compactado e extraí-lo para sua utilização, este meio tem menores riscos de problemas com dependências, assim basicamente o problema que pode ocorrer é escolher um arquivo desatualizado e incompatível com seu sistema operacional, o que geralmente pode ser contornado baixando a versão mais atual do arquivo.

No entanto, todos os três meios têm em comum a desvantagem de não ter um único arquivo, ou um único comando 100% funcional em todos os sistemas operacionais, já que para cada um deles existe uma série de comandos específicos e/ou um link exclusivo para download dos arquivos necessários. Para este projeto, o intuito é fornecer um ambiente configurado e pronto para uso com menor esforço para instalá-lo. Assim, a fim de tornar os passo únicos para instalação e configuração da rede decidiu-se no primeiro momento pela utilização do Docker, que através de um *script* único criaria-se um contêiner linux ubuntu em uma versão 19.04 com seus comandos de instalação e configuração já predefinidos através do repositório PPA da Ethereum, já que o sistema operacional e sua versão serão sempre o mesmo, a desvantagem anterior não se aplica a esta abordagem.

A abordagem do docker, no primeiro momento pareceu eficiente, uma vez que foi possível criar e configurar nós da rede totalmente funcionais, mas para a comunicação de containers em máquinas diferentes até com o mesmo sistema base, são necessárias configurações adicionais de infraestrutura que aumentaram consideravelmente a complexidade do *script* fugindo da idéia inicial de simplicidade na instalação, então decidiu-se procurar outra abordagem.

Mesmo com as diferenças entre os sistemas operacionais anteriormente citados, para a confecção de um novo *script* foi retirado o container docker e adicionados todos os comandos necessários para criar e configurar o ambiente nos três sistemas operacionais escolhidos, ficando a cargo do *script* primeiramente reconhecer qual o sistema operacional o usuário está utilizando e escolher qual a série de comandos deve ser executada. Para simplificar a quantidade de comandos, a abordagem selecionada foi o download de um arquivo binário que também é escolhido de acordo com o sistema em que for executado.

5 RESULTADOS

Foram desenvolvidos dois *scripts* cada um com o objetivo de iniciar um tipo de nó e alguns arquivos com configurações e parâmetros que serão utilizados durante a execução dos arquivos.

Antes de começar a utilizá-los, caso esteja utilizando o sistema operacional da microsoft, primeiramente instale o git através da url https://git-scm.com/download/win ou caso utilize o windows 10 o mais indicado seria ativar o Subsistema do Windows para Linux (WSL) seguindo as instruções oficias em https://docs.microsoft.com/pt-br/windows/wsl/install-win10.

Os principais componentes do *script* são os arquivos:

- genesis.json
- boot.sh
- start.sh
- .accountpassword
- .privatekey

A seguir explicamos as principais funcionalidades de cada um destes componentes:

5.1 Arquivo Genesis

Para iniciar uma nova cadeia precisamos definir o bloco inicial com algumas configurações que indicaram como novos blocos serão inseridos, dentre estas definições destacamos:

- **config**: a configuração da *blockchain*. Em suas definições temos o "chainId", um identificador utilizado na proteção

contra ataque de repetição. Por exemplo, se uma ação é validada combinando certo valor que depende do ID da cadeia, os atacantes não podem obter facilmente o mesmo valor com um ID diferente.

- **coinbase**: é um endereço onde todas as recompensas coletadas com a validação de bloco bem-sucedida serão transferidas. Uma recompensa é uma soma do pagamento pela mineração e dos reembolsos da execução de transações de contrato. Como é um bloco de inicial, o seu valor não é relevante. Para todos os próximos blocos, o valor será um endereço definido pelo mineiro que validou esse bloco.
- **difficulty**: dificuldade de mineração, para desenvolvimento e testes defina esse valor baixo para que você não precise esperar muito pelos blocos de mineração.
- **gasLimit**: o limite do custo do gás por bloco.
- **nonce**: é o número de transações enviadas de um determinado endereço. É usado em combinação com *mixhash* para provar que uma quantidade suficiente de computação foi realizada neste bloco.
- **mixHash**: um *hash* de 256 bits que, combinado com o "nonce", prova que uma quantidade suficiente de computação foi realizada no bloco. A combinação de "nonce"e *mixhash* deve satisfazer uma condição matemática.
- **parentHash**: é o *hash* do cabeçalho do bloco pai. Familiar a um ponteiro para o bloco pai necessário para formar uma cadeia real de blocos. Um bloco de gênese não possui um bloco pai, portanto, o resultado será apenas neste caso igual a 0.
- **alloc**: esse parâmetro é usado para pré-financiar alguns endereços com *ether* (criptomoeda da rede Ethereum). Ele contém dois parâmetros, o endereço da carteira que deve ser um *hash* de 160 bits e o número de *ether* com o qual uma conta deve ser financiada.

A seguir na Figura 3 temos o arquivo genesis com duas contas já pré-financiadas para não ser necessário criar uma conta manualmente e colocá-la para minerar a fim de receber fundos necessário para realizar transações.

5.2 Execução dos Scripts

A ferramenta apresenta dois *scripts* executáveis o **boot.sh** e **start.sh**, o primeiro responsável pelo nó de Boot (bootnode), o qual deve ser instanciado apenas uma única vez e apenas em uma máquina, e o segundo responsável pela instância de nós de aplicação e mineradores. As tarefas dos nós foram divididas para melhor observar as funcionalidades e tarefas executadas pelos nós da *blockchain*, de forma a tentar se aproximar de uma rede de múltiplas máquinas bem como veríamos com a rede em produção.

O processo executado por cada um dos *scripts* é basicamente o mesmo, com as diferenças apenas nas configurações necessárias para especialização de cada nó.

A Figura 4 mostra o fluxograma dos processos executados pelo usuário e pelos *scripts* ao iniciar cada nó componente da rede *blockchain*.

A seguir discutiremos mais a fundo o funcionamento e peculiaridade de cada um dos *scripts*.


```
{
  "config": {
    "chainId": 1288,
    "homesteadBlock": 0,
    "eip150Block": 0,
    "eip150Hash": "0x0000000000000000000000000000000000000000000000000000000000000000",
    "eip155Block": 0,
    "eip158Block": 0,
    "byzantiumBlock": 0,
    "constantinopleBlock": 0,
    "petersburgBlock": 0,
    "istanbulBlock": 0,
    "ethash": {}
  },
  "nonce": "0x0",
  "timestamp": "0x5f527daa",
  "extraData": "0x0000000000000000000000000000000000000000000000000000000000000000",
  "gasLimit": "0x2fefd8fffffffffff",
  "difficulty": "0x80000",
  "mixHash": "0x0000000000000000000000000000000000000000000000000000000000000000",
  "coinbase": "0x0000000000000000000000000000000000000000",
  "alloc": {
    "40ebd26453a3a3ec06df9b1cf6cb17355d95e78d": {
      "balance": "0x200000000000000000000000000000000000000000000000000000000000000"
    },
    "fd96fcc76da5e04604270bac93cd0e2acdcd670d": {
      "balance": "0x200000000000000000000000000000000000000000000000000000000000000"
    }
  },
  "number": "0x0",
  "gasUsed": "0x0",
  "parentHash": "0x0000000000000000000000000000000000000000000000000000000000000000"
}
```


Figura 3: Exemplo de um arquivo genesis.json

5.3 BootNode

Um passo importante para o correto funcionamento de uma rede privada conectada por vários nós e a definição de um nó central o qual os demais se ligaram. Nomeamos o *script* para criação deste nó como **boot.sh**.

Para a execução deste e dos próximos nós faz necessária a definição de alguns parâmetros referentes à conexão da rede. Todos os parâmetros estão definidos no início do *script* e podem ser editados ou passados por meio de *flags* na chamada de sua execução. Os parâmetros referentes ao nó do Boot e as *flags* utilizadas para alterar seus valores ao executar a função são:

- **VERSION** (-v): Versão do arquivo binário do Ethereum a ser instalado.
- **NETWORKID** (-n): Deve ser o mesmo valor do "chainId"presente no arquivo genesis.
- **BOOTDATADIR** (-d): Pasta no computador em que os arquivos da rede serão armazenados. Por padrão: **$HOME/.ethereum/private/boot**.
- **BOOTNODEKEY** (-k): Um nó de inicialização pede uma chave hexadecimal e através dela será gerado um ID com um esquema de URL chamado "enode" para conexão de outros nós, deixamos esse valor pré-definido para podermos ter certeza da url de conexão que será utilizado pelos demais nós. Esse valor pode ser gerado pelo comando: **bootnode -genkey bootnode.key**.
- **BOOTNODEIP** (-b): O IP da máquina em que será instanciado o bootnode.
- **BOOTNODEPORT** (-p): A porta em que o boot node deverá expor à rede. Por padrão: **30301**.

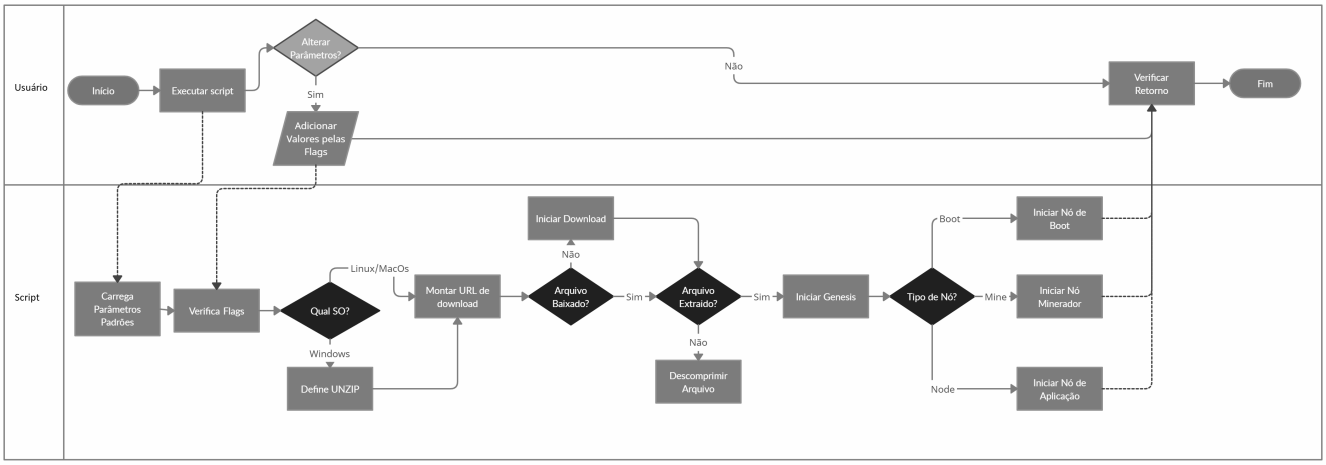

Figura 4: Fluxo geral dos *scripts* para iniciar um nó.

Depois de definidos os parâmetros, o *script* identifica qual o sistema operacional que está sendo utilizado e seleciona os comando adequados para baixar e descompactar, executar a rede, a Figura 5 mostra um exemplo deste trecho do *script*.

```
UNZIP="tar -xvzf"        #comando para descompactar o arquivo binário (linux e mac)
if [[ "$OSTYPE" == "linux-gnu" ]]; then     #verifica se é um sistema linux
  OS='linux'
  EXT='tar.gz'                              #extensão do arquivo binário
elif [[ "$OSTYPE" == "darwin"* ]]; then   #verifica se é um sistema mac
  OS='darwin'
  EXT='tar.gz'                              #extensão do arquivo binário
else                                        #adimite que é um sistema windows
  OS='windows'
  EXT='zip'                                 #extensão do arquivo binário
  UNZIP="unzip"               #modifica o comando para descompactar o arquivo binário
fi
...
if [  -f "$FILEEXT" ]; then       #verifica se é o arquivo já foi baixado
   echo 'Arquivo encontrado'
else
    curl -O $URL            #efetua download do arquivo binário
fi
if [  -d "$FILE" ]; then      #verifica se é o arquivo foi descompactado
    echo 'Arquivo descomprimido'
else
    $UNZIP $FILEEXT         #descompacta o arquivo binário
fi
```

Figura 5: Comandos de para baixar o arquivo conforme o sistema operacional identificado

Em seguida, na Figura 6 temos o trecho do *script* responsável pelos comandos que executam o nó central.

```
#inicializar a cadeia com o bloco genesis
 $FILE/geth --datadir=$DATADIR init genesis.json
#iniciar o bootnode da rede
 $FILE/geth --datadir=$DATADIR --nodekeyhex=$BOOTNODEKEY  --networkid $NETWORKID
 --nat extip:$BOOTNODEIP --port $BOOTNODEPORT&>boot.log&
```

Figura 6: Comandos de execução do nó central

O primeiro comando gera o bloco inicial e o segundo inicia a rede com os parâmetros definidos anteriormente que será executado em background e guardando as saídas da execução no arquivo **boot.log**.

Ao executar o arquivo **boot.sh** por linha de comando, caso não seja passados nenhum argumento a rede será instanciada com todos os parâmetros padrões, dentre eles o que pode inviabilizar a utilização da rede, caso incorreto, é o IP da máquina, logo certifique-se que este parâmetro foi definido corretamente.

Exemplo da utilização do *script* **boot.sh** é exibido na Figura 7 abaixo.

```
./boot.sh                    #iniciar com todos os parâmetros padrões
./boot.sh -i 192.168.1.158  #iniciar alterando ip do bootnode
```

Figura 7: Comandos para inicar o boot.sh

O trecho abaixo apresenta a saída esperada escrita no log, indicando que a rede foi inicializada e qual é o endereço de conexão (enode) de novos nós.

```
INFO [09-24|18:00:36.897] Started P2P networking self
=enode://4e87faaa0ed677c3ec389f3ac37f8b0e366876f73e72
764e3518031daca322768befb783be5c4aea4200f3439f4361571
e860c38776142094adc35913964096b@192.168.1.158:30301
```

Se estiver utilizando o sistema windows certifique-se que tenha instalado o git e execute os *scripts* através do terminal do wsl ou git bash. Uma forma mais rápida de utilizá-lo seria dentro da pasta dos *scripts* clicar com o botão direito do mouse e escolher a opção "Git Bash Here", ou abri-lo através do menu de programas

5.4 Nós de aplicação e mineração

Com o bootnode criado, podemos integrar à redes mais dois tipos de nós, o de aplicação (responsável por externar a API que será utilizado para inserção e consulta dos dados da *blockchain*) e outro nó para mineração dos dados enviados para serem inseridos na rede.

Para estes dois tipos de nós foi criado apenas um *script* sendo indicado qual o tipo de nó deseja ao iniciar o *script*. Dessa forma, a diferença no *script* para os dois tipos de nós é apenas os parâmetros indicados para execução da rede.

Os arquivos criados para este fim são o start.sh (*script* executável), .accountpassword (contendo a senha da carteira a ser pré-alocada) e .privatekey (chave privada da carteira pré-alocada). A senha e a chave privadas foram pré definidas por estarmos importando uma conta ao invés de criar uma nova, já que para pré-financiar uma conta devemos colocá-la no arquivo genesis.json antes de iniciarmos a rede.

Como no arquivo anterior, temos no início do arquivo a definição de parâmetros. Os parâmetros referentes a esses nós e as *flags* utilizadas para alterar seus valores ao executar a função são:

- **NODETYPE** (-t): Identifica o tipo de nó, aceita como valores: 'node' para um nó de aplicação, este definido por padrão, e 'mine' para um nó minerador.
- **OPERATIONTYPE** (-o): Aceita os comandos 'start' e 'stop' para, respectivamente, iniciar e para a rede *blockchain*.
- **MYNODEPORT** (-p): Porta em que será executada a rede no computador que está iniciando o nó. Por padrão: **30303**.
- **DATADIR** (-d): Pasta no computador em que os arquivos da rede serão armazenados. Por padrão: **$HOME/.ethereum/private/node**.
- **BOOTNODEIP** (-i): Deve ser o IP da máquina que está rodando o bootnode.
- **BOOTNODEID** (-b): Deve ser o id ("enode") gerado pela execução do bootnode, se não foi alterado o **BOOTNODEKEY** no boot.sh este valor já está configurado.
- **BOOTNODEPORT** (-r): Porta em que está sendo executado bootnode. Por padrão: **30301**.
- **NETWORKID** (-n): É o mesmo "chainId"do arquivo genesis.json.

Estes parâmetros podem ser alterados diretamente no *script* ou passado como argumentos em sua execução. Um exemplo é demonstrado na figura 8.

```
./start.sh

./start.sh -t node

./start.sh -t mine -i 192.168.1.18
```

Figura 8: Comandos e parâmetros para iniciar a rede.

No primeiro comando iniciamos a rede com todos os parâmetros pré definidos, no segundo deixamos explícito que queremos iniciar um nó do tipo aplicação, e no último iniciamos um nó minerador indicando um outro valor para o IP do bootnode.

Quanto ao funcionamento do *script*, assim como no anterior após a definição dos parâmetros é identificado o sistema operacional e selecionado os comandos corretos. Em seguida é necessário iniciar a rede *blockchain* com o mesmo arquivo genesis do bootnode, e posteriormente o seguinte comando da Figura 9 serve para iniciar o novo nó e o conectando a rede já iniciada.

Neste comando podemos notar que comumente para os dois tipos de nó ao ser iniciados o argumento –bootnodes indica a url de

```
#NÓ DE APLICAÇÃO
$FILE/geth --datadir=$DATADIR --bootnodes "enode://$BOOTNODEID@$BOOTNODEIP:
 $BOOTNODEPORT" --networkid $NETWORKID --port $MYNODEPORT --verbosity=4
 --rpc --rpcaddr "0.0.0.0" --rpcapi "eth,web3,net,admin,debug,personal"
 --rpccorsdomain "*" --syncmode="full"  $IPC console

#NÓ MINERADOR
$FILE/geth --datadir=$DATADIR --bootnodes "enode://$BOOTNODEID@$BOOTNODEIP:
 $BOOTNODEPORT" --networkid $NETWORKID --port $MYNODEPORT --verbosity=4
 --syncmode="full"  --gasprice "0"  --etherbase $ADRESSACCOUNT
 --unlock $ADRESSACCOUNT --password $ACCOUNTFILE --mine
 --miner.threads 1  $IPC
```

Figura 9: Comandos para iniciar e conectar um novo nó a rede.

conexão a rede iniciada pelo bootnode e o networkid confirma que o ID de todos os nós são iguais para compartilhar as informações.

O que define que o novo nó será de aplicação são os argumentos "–rpc –rpcaddr –rpcapi –rpccorsdomain", responsáveis pela configuração de um servidor responsável pela API de comunicação com serviços externos, dentre estas configurações temos quais as funções que serão liberadas pela API pelo argumento –rpcapi e quais endereços IP terão acesso a requisições com –rpccorsdomain.

O nó minerador tem como características principais os argumentos "–etherbase $adressAccount –unlock $adressAccount –password $accountFile –mine". Indicando assim, qual o endereço da base de *ether* ou seja o endereço da conta mineradora bem como destravando a conta para realizar as transações e o argumento –mine para que já seja iniciada a tarefa de mineração ao iniciar o nó.

Abaixo temos a saída esperada do nó de aplicação quando iniciado, podemos notar que a última linha indica que o servidor HTTP foi ativado, característica existente apenas nesse tipo de nó.

```
INFO [09-24|18:10:56.117] HTTP server started
endpoint=127.0.0.1:8545 cors= vhosts=localhost
```

Enquanto no próximo trecho temos a saída esperada da execução de um nó minerador, este tem como característica o início do trabalho de mineração indicado pela saída "Commit new mining work".

```
INFO [09-24|18:15:13.672] Commit new mining work
nunber=1 sealhash="c8ecb8...6394dc" uncles=0 txs=0
gas=0 fess=0 elapsed="216.9s"
```

Para utilizar a ferramenta, basta acessar o repositório (https://github.com/meloflavio/private_ethereum_scripts) o qual estão descritos o seu funcionamento e apresenta um video tutorial demostrando sua utilização.

6 DISCUSSÕES

O desenvolvimento deste trabalho tinha o objetivo de apresentar um produto educacional destinado àqueles que pretendem iniciar seus estudos práticos na área do *blockchain*. Foram desenvolvidos *scripts* e um tutorial para a criação de um ambiente completo de uma rede Ethereum. Com estes *scripts* não só o ambiente é construído

como também é apresentando uma parte teórica sobre os conceitos necessários para criar uma cadeia de blocos.

Dessa forma, este trabalho pode ser utilizado para introduzir o conceito de *blockchain* bem como explicar seu funcionamento e detalhes necessários para sua configuração resultando em uma aula prática na qual o aluno poderá construir sua própria rede *blockchain*, exemplificando também um sistema distribuído. Todavia, um maior aprofundamento no básico da tecnologia *blockchain* é desejável, pois os conceitos apresentados estão concentrados apenas na estrutura do bloco.

Uma aula de Segurança em Tecnologia da Informação, por exemplo, seria interessante também ser apresentada a criptografia empregada na rede *blockchain* como uma técnica de proteção para comunicação segura. Já em aulas sobre Banco de Dados, pode se fazer um paralelo entre as duas tecnologias para indicar as diferenças e em que situação devemos utilizar cada uma dessas tecnologias. Neste sentido, a análise do *script* pode abordar conceitos de outras disciplinas, o *script* como um todo é um bom exemplo de algoritmo podendo ser utilizado em aulas como Introdução a Programação e Algoritmos, por exemplo as verificações do sistema operacional, se o download ou descompressão do arquivo já foram executadas podem demonstrar o funcionamento de estruturas de seleção.

Em aulas de Sistemas Operacionais fazendo uso do *script* pode-se abordar chamadas de sistema, explicar o que são processos, seus estados, execução em primeiro e segundo plano e o que os diferencia dos programas. Detalhes como o redirecionamento de portas, o servidor HTTP do nó de aplicação e as permissões de acesso à api da *blockchain* poderão também ser utilizados nas disciplinas que abordam configurações de redes.

A primeira versão deste *script* foi utilizada durante uma aula de Computação e Sociedade, disciplina do primeiro semestre do curso de Ciência da Computação da Universidade Federal do Tocantins, nesta aula foram apresentados cada um dos passos de execução do *script* e os conceitos envolvidos a fim de explanar novas tecnologias e abrir uma troca de informações com os conceitos familiares aos aluno. Neste caso, o maior resultado dessa experiência não é necessariamente o resultado do *script*, mas a exposição de todas áreas de estudos envolvidas em sua execução que possibilita o debate de todas as possibilidades que a computação nos traz.

Durante a apresentação, os alunos e o professor da disciplina puderam discutir cada um dos conceitos apresentados utilizando o *script*, os alunos puderam identificar de uma forma prática a utilização de diversas disciplinas que eles estudarão no decorrer de seu curso de graduação, nesta perspectivas diversos alunos interagiram com perguntas e comentários que demonstravam seus interesses e alguns conhecimentos básicos sobre cada um dos conceitos apresentados, de uma forma orgânica ocorreram debates mais aprofundados sobre os assuntos que os alunos demonstravam maior interesse.

Ao final da aula, alguns dos alunos continuaram discutindo sobre a apresentação, solicitando algumas dicas e materiais sobre as disciplinas que mais lhes chamaram a atenção. Neste momento, foi possível observar também que a apresentação despertou a curiosidade sobre algumas novas possibilidades oferecidas pela tecnologia *blockchain*.

Desse modo, a apresentação dos *scripts* nesta aula serviu não apenas para demonstrar a criação de uma rede de *blockchain*, mas também para ensinar alguns dos conceitos básicos das disciplinas envolvidas no desenvolvimento dos *scripts*, além disso a apresentação despertou o interesse dos alunos em se aprofundarem nestas disciplinas demonstradas.

7 CONSIDERAÇÕES FINAIS

Por fim, este produto educacional, ou recurso didático, pode ser utilizado por outros professores em sala de aula para apresentarem o comportamento de rede *blockchain* na prática e discutir os demais conceitos envolvidos. Além disso, o material pode auxiliar as pessoas que estão estudando por conta própria na criação de suas redes *blockchain* privadas iniciais na plataforma Ethereum, já prontas para interação com outros sistemas.

Além dos produtos já descritos neste trabalho, espera-se que este trabalho continue a evoluir, já estão em desenvolvimento para próximas etapas a implantação de exemplos de contratos inteligentes e um tutorial para compor este produto educacional. Essa e outras atualizações serão incorporadas ao repositório no GitHub.

REFERÊNCIAS

[1] Tesnim Abdellatif and Kei-Léo Brousmiche. 2018. Formal verification of smart contracts based on users and blockchain behaviors models. In *2018 9th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*. IEEE, 1–5.

[2] Andreas M Antonopoulos. 2014. *Mastering Bitcoin: unlocking digital cryptocurrencies*. "O'Reilly Media, Inc.".

[3] Eduardo Fernandes Barbosa and Dácio Guimarães de Moura. 2013. Metodologias ativas de aprendizagem na educação profissional e tecnológica. *Boletim Técnico do Senac* 39, 2 (2013), 48–67.

[4] William N Bender. 2012. *Project-based learning: Differentiating instruction for the 21st century*. Corwin Press.

[5] Bertony Bornelus, Hongmei Chi, and Hossain Shahriar. 2019. A Novel Framework to Teach Hands-on Laboratory Exercises in Blockchains. (2019).

[6] Jiin-Chiou Cheng, Narn-Yih Lee, Chien Chi, and Yi-Hua Chen. 2018. Blockchain and smart contract for digital certificate. In *2018 IEEE international conference on applied system invention (ICASI)*. IEEE, 1046–1051.

[7] DANIEL G COSTA. 2010. *Administração de redes com scripts: Bash Script, Python e VBScript*. Brasport.

[8] Antonio Sávio dA SilvA Pinto, Marcilene Rodrigues Pereira Bueno, Maria Aparecida Félix do Amaral, Milena Zampieri Sellmann, Sônia Maria Ferreira Koehler, et al. 2012. Inovação Didática-Projeto de Reflexão e Aplicação de Metodologias Ativas de Aprendizagem no Ensino Superior: uma experiência com "peer instruction". *Janus* 9, 15 (2012).

[9] Edna de Almeida Rodrigues and Adriana Maria Procópio de Araújo. 2007. O ensino da contabilidade: aplicação do método PBL nas disciplinas de contabilidade em uma instituição de ensino superior particular. *Revista de Educação* 10, 10 (2007).

[10] Wenliang Du. 2011. SEED: hands-on lab exercises for computer security education. *IEEE Security & Privacy* 9, 5 (2011), 70–73.

[11] Luiz Otávio Ramos Gavaza, Laıs do Nascimento Salvador, and David Moises Barreto dos Santos. 2017. Uma experiência de aplicação de uma abordagem baseada em problemas no ensino de teoria da computação em sala de aula tradicional. In *Anais do XXV Workshop sobre Educação em Computação*. SBC.

[12] Marco Iansiti and Karim R Lakhani. 2017. The truth about blockchain. *Harvard Business Review* 95, 1 (2017), 118–127.

[13] Paola YB Ogawa Letouze, Patrick Letouze, JIM de Souza Junior, Bruna Laisy C Everton, Denise S Araujo, and Gentil Veloso Barbosa. 2020. Court-Ordered Government Debt Payment in Brazil: Perspectives for Blockchain Technology. *International Journal of Social Science and Humanity* 10, 4 (2020).

[14] Xiuping Lin. 2017. Semi-centralized Blockchain Smart Contracts: Centralized Verification and Smart Computing under Chains in the Ethereum Blockchain. *Department of Information Engineering, National Taiwan University, Taiwan, ROC* (2017).

[15] Satoshi Nakamoto and A Bitcoin. 2008. A peer-to-peer electronic cash system. *Bitcoin.–URL: https://bitcoin. org/bitcoin. pdf* (2008).

[16] Solomon Negash and Dominic Thomas. 2019. Teaching Blockchain for Business. In *2019 IEEE Canadian Conference of Electrical and Computer Engineering (CCECE)*. IEEE, 1–4.

[17] Benedikt Notheisen, Jacob Benjamin Cholewa, and Arun Prasad Shanmugam. 2017. Trading real-world assets on blockchain. *Business & Information Systems Engineering* 59, 6 (2017), 425–440.

[18] Eduardo Oliveira and Angilberto Freitas. 2020. Os porquês da tecnologia blockchain ainda não ter sido popularizada: um ensaio teórico. *Revista Gestão & Tecnologia* 20, 1 (2020), 332–343.
[19] Michael Orey. 2010. *Emerging perspectives on learning, teaching and technology*. CreateSpace North Charleston.
[20] A Ravishankar Rao and Riddhi Dave. 2019. Developing hands-on laboratory exercises for teaching STEM students the internet-of-things, cloud computing and blockchain applications. In *2019 IEEE Integrated STEM Education Conference (ISEC)*. IEEE, 191–198.
[21] Henrique Martins Rocha and Washington de Macedo LEMOS. 2014. Metodologias ativas: do que estamos falando? Base conceitual e relato de pesquisa em andamento. *IX Simpósio Pedagógico e Pesquisas em Comunicação. Resende, Brazil: Associação Educacional Dom Boston* 12 (2014).
[22] John R Savery. 2015. Overview of problem-based learning: Definitions and distinctions. *Essential readings in problem-based learning: Exploring and extending the legacy of Howard S. Barrows* 9 (2015), 5–15.
[23] Célia Maria Piva Cabral Senna and Graziela Miê Peres Lopes. [n.d.]. Aprendizagem baseada em projetos como forma de inclusão. ([n. d.]).
[24] Elianny Sousa Silva, Brenda Jamille Costa Dias, João Lucas Moraes Souza, and Mariana Souza de Lima. 2019. Aprendizagem baseada em problema aplicada no ensino de urgência e emergência na enfermagem: um relato de experiência/Learning based on a problem applied in emergency and nursing education in nursing: an experience report. *Brazilian Journal of Health Review* 2, 4 (2019), 2525–2529.
[25] José Itamar Souza Junior, Denise Sampaio de Araujo, Gentil Veloso, and Patrick Letouze. 2019. An international accreditation system for healthcare professionals based on blockchain. *International Journal of Information and Education Technology* 9, 7 (2019), 462–469.
[26] Brasil. Tribunal de Contas da União. 2020. Levantamento da tecnologia blockchain. (2020). https://portal.tcu.gov.br/levantamento-da-tecnologia-blockchain.htm