

UNIVERSIDADE FEDERAL DE MINAS GERAIS - UFMG
CIÊNCIA DA COMPUTAÇÃO
BLOCKCHAIN, CRIPTOMOEDAS E OUTRAS APLICAÇÕES

SISTEMA DE ACOMPANHAMENTO ACADÊMICO DESCENTRALIZADO

André Lustosa Cabral de Paula Motta
Artur Marzano
Esthefanie Lanza

Belo Horizonte, Maio de 2018

INTRODUÇÃO

A proposta deste trabalho foi desenvolver um protótipo de aplicação de acompanhamento acadêmico para escolas e universidades capaz de proporcionar os seguintes aspectos:

- Realizar lançamento de notas de atividades acadêmicas;
- Capaz de vinculação de alunos a uma turma e criação de tarefas para a classe;
- Garantir que o sistema seja seguro, à prova de fraudes e que esteja sempre disponível;

A vantagem da escolha de uma blockchain distribuída e *permissionless* para a aplicação no escopo acadêmico em particular é (a) prover transparência, integridade e responsabilidade aos agentes envolvidos e (b) garantir disponibilidade 24/7 do sistema acadêmico.

Um aluno prejudicado que tenha uma reclamação, por exemplo, terá registradas as ações de seus professores de maneira segura e confiável para dar prosseguimento a uma denúncia. De maneira análoga, um professor terá as ações de seus alunos registradas para justificar suas ações. Órgãos superiores como o corpo discente ou um colegiado também serão capazes de averiguar as ações dos professores e dos alunos com as mesmas propriedades, como para gerenciar promoções de carreira ou alocações de disciplinas.

Quanto ao segundo item, um sistema distribuído com mecanismos de incentivo estará sempre *online* desde que alguns nós estejam, de forma que ninguém será prejudicado pela indisponibilidade clássica de sistemas acadêmicos centralizados, como o Moodle da UFMG.

DESENVOLVIMENTO

O Contrato Classe

Para desenvolver o sistema, usamos a linguagem de programação própria da blockchain do Ethereum, a Solidity. O contrato classe possui métodos para inserção de alunos em uma classe. Além disso, os professores deverão ser capazes de realizar o lançamento de atividades e notas. O contrato é apenas um protótipo do produto final que foi idealizado como um sistema completo de ensino, mas o mesmo já foi escrito pensando na sua expansão, toda classe pode ser criada por um construtor que passa o nome da disciplina e o endereço do professor responsável, possibilitando futuramente expandir o conceito para criação de diversas disciplinas e alocação de professores por disciplinas.

Também foi considerado que usuários do tipo aluno (ou não-usuários) não possuem permissão para alterar qualquer tipo de informação. Em outras palavras, estes são usuários passivos do sistema e só poderão consultar informações e verificar sua autenticidade.

Os contratos foram publicados na rede de testes *Rinkeby* e o ambiente de desenvolvimento será composto das ferramentas *Geth* e *Truffle*.

A interação com os contratos foi feita via Javascript com o *Web3* injetado pela extensão do *MetaMask*, combinação usada pela grande maioria das interfaces dessa espécie de aplicação. Verificamos os contratos desenvolvidos por *bugs* comuns e problemas de segurança com análise estática e dinâmica por meio de ferramentas como o *Mythril*, *Oyente*, *Manticore*, etc.

A Interface em ReactJS

Para facilitar o uso do sistema, criamos uma interface simples capaz de interagir com a blockchain. A lógica do front-end dessa interface foi desenvolvida com a biblioteca *ReactJS*, um framework para aplicações web que hoje é mantido principalmente pelo Facebook e o Instagram e tem ganhado visibilidade por sua facilidade em criar interfaces e gerenciar estados de uma aplicação.

Usamos a biblioteca *Material UI* que implementa componentes React usando os princípios de *Material Design* da Google. Dessa forma, será possível desenvolver uma interface funcional e visualmente agradável sem grande esforço, pois não será necessário desenvolver

componentes de interface do zero.

Na interface é possível consultar notas e faltas, inserir alunos e atividades, lançar notas das atividades e faltas aos alunos. A interface tem usabilidade simplificada tratando-se de um protótipo do escopo completo do trabalho.

DIFICULDADES

A dificuldade principal que encontramos está principalmente relacionado ao fato de se tratar de uma tecnologia com a qual não temos familiaridade, embora o sistema em si seja logicamente simples de ser desenvolvido. Um dos problemas que tivemos foi garantir a sincronia na apresentação dos resultados ao usuário em virtude da natureza assíncrona dos processos que interagem com a blockchain. Outros problemas menores que experienciamos, são relacionados a interface javascript do web3 injetado pela MetaMask que, por se tratar de uma tecnologia relativamente nova, apresenta bugs e outras complicações como documentação incompleta e falta de suporte.

DIREÇÕES FUTURAS

Embora o tempo da entrega da proposta e a finalização do projeto tenha sido curto, limitando o escopo da aplicação, desenvolvemos a base necessária para uma implementação completa. O projeto em geral consistiria na criação de novos contratos que interagem com o contrato classe possibilitando a gerência completa e transparente de uma entidade educacional. Em termos de implementação, uma direção futura é implementar um contrato “Escola” que manteria uma instância do contrato classe para cada disciplina ofertada e gerenciaria a alocação de professores e outras características desta disciplina.

CONCLUSÃO

Por fim o trabalho foi desenvolvido sem grandes problemas, e cumprimos a proposta de um protótipo funcional de um sistema acadêmico já preparado para receber expansões de módulos e novas funcionalidades. O trabalho foi extremamente interessante para se aprender um pouco mais os conceitos passados durante a disciplina e para entender e estudar a

viabilidade de certos projetos numa blockchain.

REFERÊNCIAS

- Solidity Documentation, <https://solidity.readthedocs.io/en/v0.4.24/>
- The Hitchhiker's Guide to Smart Contracts in Ethereum,
<https://blog.zeppelin.solutions/the-hitchhikers-guide-to-smart-contracts-in-ethereum-848f08001f05>
- Blockchain: smart contract benefits and vulnerabilities
<https://medium.com/swlh/blockchain-smart-contract-benefits-and-vulnerabilities-7543b3955ac9>
- A Gentle Introduction to Ethereum Programming
<https://blog.zeppelin.solutions/a-gentle-introduction-to-ethereum-programming-part-1-783cc7796094>
- dApp Radar. <https://dappradar.com/>