

Sistema de Recarga de VEs Baseado em Blockchain

Caick Wendell Lopes dos Santos¹, João Lucas Silva Serafim Silva¹

¹Departamento de Tecnologia – Universidade Estadual de Feira de Santana (UEFS)
Av. Transnordestina, s/n, Novo Horizonte / Feira de Santana – BA, Brasil- 44036-9

Resumo. A demanda dentro do contexto de inovação está atrelada a necessidade de priorizar alguns aspectos do produto. Neste sentido, as crescentes atualizações no contexto dos veículos elétricos põem em dúvida a confiabilidade e auditabilidade do cenário. O documento a seguir descreve o processo para implementação de tecnologia de *blockchain* em um sistema de recarga de VEs. O sistema utiliza de *blockchain* de tipo Ethereum para comunicações periódicas com o servidor. A aplicação permite o registro e sincronização de compras através de um banco de dados descentralizado.

Abstract. Demand within the context of innovation is linked to the need to prioritize certain aspects of the product. In this sense, the increasing updates in the context of electric vehicles raise questions about the reliability and auditability of the scenario. The following document describes the process for implementing blockchain technology in an EV charging system. The system uses Ethereum-type blockchain for periodic communications with the servers. The application enables the registration and synchronization of purchases through a decentralized database.

1. Introdução

O crescimento da indústria de veículos elétricos requer que cada vez mais demandas sejam atendidas de forma a tornar a experiência dos usuários e das empresas a mais confortável e efetiva possível. Entretanto, além das questões mais notáveis e tangíveis como o gerenciamento de rotas, a integração entre os sistemas e a escassez de auditabilidade acarretam na carência de confiabilidade.

Este relatório apresenta a descrição do processo de aprimoramento de um sistema de recarga de VEs. De forma que o mesmo adquira maior confiabilidade e auditabilidade através da implementação de tecnologia de *blockchain*. A aplicação permanece utilizando a linguagem Python em sua maior parte, com o auxílio dos protocolos *Message Queue Telemetry Transport*(MQTT) com broker Mosquitto e arquitetura *Rest*. Esta atualização conta com o uso da biblioteca W3 para Python e linguagem de programação Solidity. A aplicação foi projetada em sistemas Linux com o auxílio da IDE Visual Studio Code, simulador Ethereum Ganache e plataforma GitHub.

O programa conta com todas as funcionalidades de sua versão anterior e com a adição das funcionalidades de registro de compras em um banco de dados descentralizado e sincronização do mesmo entre os servidores. Este relatório está organizado da seguinte forma. A seção 2 contém o conhecimento teórico utilizado para o desenvolvimento da aplicação. A seção 3 apresenta as ferramentas e métodos utilizados. A seção 4 demonstra os resultados e falhas obtidos. Por último a seção 5 refere-se as conclusões obtidas ao término do projeto.

2. Fundamentação Teórica

A seção a seguir tem por objetivo fazer um breve detalhamento do conhecimento teórico necessário para obtenção do produto final.

2.1. *Blockchain*

A primeira execução do conceito de *blockchain* em uma rede a vincula como base para o surgimento das criptomoedas e está presente no paper “Bitcoin: A Peer-to-Peer Electronic Cash System” por Satoshi Nakamoto[Di Pierro 2017]. A *blockchain* é uma tecnologia de registro distribuído que organiza dados em forma de blocos protegidos por criptografia e interligados de forma a criar uma cadeia segura e imutável.

Funciona como um livro-razão composto por uma rede *peer-to-peer* e um banco de dados distribuído descentralizado.

2.1.1. Contratos inteligentes

Um contrato inteligente é um código que roda em uma *blockchain* com o objetivo de facilitar, executar e impor os termos de um acordo entre partes não confiáveis[Alharby and Van Moorsel 2017].

3. Metodologia

A princípio, fizeram-se necessárias pesquisas que esclarecessem o comportamento e implementação da tecnologia de *blockchain* junto a discussões com o intuito de direcionar os primeiros passos com base na tentativa de prever possíveis alterações.

As discussões resultaram na conclusão de que a estrutura do projeto anterior seria mantida e o planejamento seguinte centralizou-se na adição de novas funcionalidades e integração das mesmas.

Ao analisar o funcionamento do estágio anterior da aplicação. Nota-se que grande parte das funcionalidades incluem o emprego de listas encadeadas, o planejamento inicial do projeto voltou-se para esta característica e então foram criados protótipos que desempenhassem funções semelhantes até que houvesse uma execução desejada.

Os protótipos giravam em torno do processo de envio e leitura de mensagens e impressão de strings em um terminal e serviram de ponte para compreender por completo o manuseio de *blockchains*. Os mesmos já contavam com contratos inteligentes escritos em linguagem Solidity compilados, implantados e executados em linguagem Python e testados através de simulador Ethereum Ganache.

O protótipo final detinha um contrato o qual continha uma lista de *strings* e era capaz de adicionar, ler e obter o tamanho da mesma.

```

// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;

contract ListaCodificada {
    string[] private listaJson;

    event ListaAdicionada(string jsonData);

    function adicionarLista(string memory jsonData) public {
        listaJson.push(jsonData);
        emit ListaAdicionada(jsonData);
    }

    function obterLista(uint index) public view returns (string memory) {
        require(index < listaJson.length, 'indice invalido');
        return listaJson[index];
    }

    function totalListas() public view returns (uint) {
        return listaJson.length;
    }
}

```

Figura 1. Contrato utilizado nas operações do sistema

O formato do contrato em seu protótipo final mostrou-se ideal para integração com o sistema e seu processo de incorporação foi iniciado.

O protótipo foi dividido em duas partes para melhor coesão. A implantação e as transações foram integradas separadamente, logo, faz-se necessário que o processo de implantação do contrato seja iniciado antes de qualquer servidor.

No que tange as funcionalidades novas, foram adicionadas duas funções relacionadas ao processo de registro de compras aos servidores: *addToBlockchain()* e *syncWithBlockchain()*. A primeira é responsável por registrar que uma compra foi feita e é executada de maneira assíncrona. Enquanto a segunda é executada em um período pré-determinado de dez minutos e é responsável por sincronizar o servidor com a *blockchain*, ou seja, atualizar os dados locais permitindo a confiabilidade de dados entre servidores.

4. Resultados

Esta seção é dedicada a discutir os resultados obtidos durante a produção.

O atualização do produto demonstrou-se um sucesso e o sistema de recarga apresenta uma confiabilidade muito maior que a versão anterior. Os testes feitos englobavam os protótipos citados na seção 3 e a execução do sistema como um todo de maneira sequencial junto a averiguação do processo através das ferramentas disponíveis no simulador Ganache.

ACCOUNTS

BLOCKS

TRANSACTIONS

CONTRACTS

EVENTS

LOGS

SEARCH FOR BLOCK NUMBERS OR TX HASHES

CURRENT BLOCK

2

GAS PRICE

20000000000

GAS LIMIT

6721975

HARDFORK

MERGE

NETWORK ID

5777

RPC SERVER

HTTP://127.0.0.1:7545

MINING STATUS

AUTOMINING

WORKSPACE

QUICKSTART

SAVE

SWITCH

TX HASH

0x56cfa2639cd681ac26617b08e6c1ba86338ffa6e15ccb5a6c547032d31e57256

CONTRACT CALL

FROM ADDRESS

0xEfCB283b48342d629e20a3cB1564A86025CABe3f

TO CONTRACT ADDRESS

0xa4848fe4Aa8AFB597ea028D235D3e6e278dA4972

GAS USED

114683

VALUE

0

TX HASH

0xfb30a5af5bba43114c2ba44693e6b7152a2e85a81dda24b01573564f5a2a58b3

CONTRACT CREATION

FROM ADDRESS

0xEfCB283b48342d629e20a3cB1564A86025CABe3f

CREATED CONTRACT ADDRESS

0xa4848fe4Aa8AFB597ea028D235D3e6e278dA4972

GAS USED

408372

VALUE

0

Figura 2. Aba de transações na interface do Ganache

CURRENT BLOCK 2	GAS PRICE 20000000000	GAS LIMIT 6721975	HARDFORK MERGE	NETWORK ID 5777	RPC SERVER HTTP://127.0.0.1:7545	MINING STATUS AUTOMINING	WORKSPACE QUICKSTART	SAVE	SWITCH	⚙️
BLOCK 2	MINED ON 2025-06-13 17:53:49				GAS USED 114683		1 TRANSACTION			
BLOCK 1	MINED ON 2025-06-13 17:53:49				GAS USED 408372		1 TRANSACTION			
BLOCK 0	MINED ON 2025-06-13 17:50:40				GAS USED 0		NO TRANSACTIONS			

(a) Aba de blocos na interface do Ganache

O sistema inclui as capacidades de registrar e sincronizar compras de forma descentralizada e confiável.

5. Conclusão

O sistema é capaz de oferecer uma experiência muito mais segura ao usuário ao proporcionar a capacidade de auditoria confiável por conta da tecnologia de *blockchain* adicionada ao sistema de gerenciamento de compras.

O desenvolvimento da aplicação mostrou-se essencial para o aprendizado no contexto de manuseio de *blockchain*, além do exercício dos conceitos de confiabilidade, integridade e auditoria.

Um ponto de melhoria para possíveis atualizações seria a adição de recursos para registros de adição de reserva de maneira semelhante ao sistema de compra já existente.

Referências

- Alharby, M. and Van Moorsel, A. (2017). Blockchain-based smart contracts: A systematic mapping study. *arXiv preprint arXiv:1710.06372*.
- Di Pierro, M. (2017). What is the blockchain? *Computing in Science & Engineering*, 19(5):92–95.