

USO DE BLOCKCHAIN E CONTRATOS INTELIGENTES NA GESTÃO DO CICLO DE VIDA DE CARROS

Aluno: Yuri Matheus Hartmann

Orientador: Marcel Hugo

Roteiro

- Introdução
- Objetivos
- Fundamentação teórica
- Trabalhos correlatos
- Especificação
- Implementação
- Resultados
- Conclusão
- Sugestões

Introdução

- Grande volume de burocracia abre margem para simplificar e automatizar com tecnologias
- Ao possuir um carro geram-se varias informações durante todo seu ciclo de vida
- As informações ficam pulverizadas em várias bases de dados
- Os dados não são confiáveis
- Com tecnologia podemos simplificar a burocracia, garantir os dados e ter centralizado as informações

Objetivos

- **Objetivo Principal: Desenvolver um protótipo para gerenciar o ciclo de vida de um carro utilizando *blockchain* privada e contratos inteligentes**
 - a) identificar e modelar o ciclo de vida de carros;
 - b) criar uma rede *blockchain* privada usando Hyperledger Fabric;
 - c) construir contratos inteligentes para gerir as regras de negócio envolvendo o carro;
 - d) avaliar a viabilidade do protótipo para funcionamento no mundo real.

Fundamentação Teórica

Blockchain

Contratos inteligentes

Hyperledger Fabric

Blockchain e Contratos inteligentes

- *Blockchain* é utilizada em vários setores como: financeiro, saúde, logística etc.
- Principais recursos de uma *blockchain*: **Descentralização, Imutabilidade e o Consenso**
- *Blockchain* privada, possui mecanismo de autenticação, apenas organizações com acesso podem fazer parte da rede
- Contratos inteligentes, ou chaincode, são códigos no *blockchain*, com regras de negócio que pode ser executado e validados por todos na rede

Hyperledger Fabric

- Uma plataforma *blockchain* de código aberto desenvolvida pela Linux Foundation para criar soluções de negócios para empresas
- *Blockchain* privada, possui mecanismo de autenticação, apenas organizações com acesso podem fazer parte da rede
- Multicanais de comunicação dentro da rede *blockchain*
- Chaincodes personalizados e livre para customização das empresas



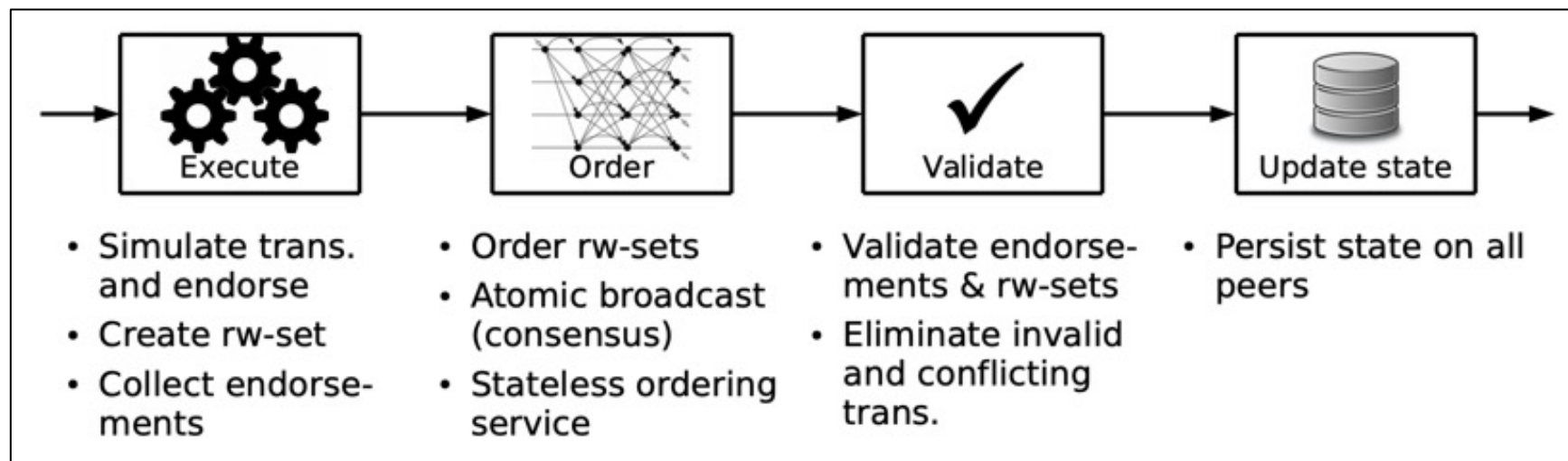
Atores da rede

- **Cliente:** submete as transações
- **Nó par (ou apenas nó):** Gerencia o livro-razão e executa os chaincodes
- **Serviço de ordenação:** Recebe as transações, executa a ordenação, agrupa em um bloco e envia para os nós. Não possui salvo o livro-razão.

Fluxo de transação

- Execução de uma transação dentro da Hyperledger Fabric
 - Etapa de execução
 - Etapa de ordenação
 - Etapa de validação

Fluxo de transações no Hyperledger Fabric



Trabalhos Correlatos

Quadro dos trabalhos correlatos

Trabalho Correlato 1

Referência	MENEZES, Leonardo Dias. Blockchain e cartórios: uma solução viável?. 2020. 61 folhas. Dissertação (Mestrado em Ciências) - Universidade de São Paulo, São Paulo.
Objetivos	Verificar a viabilidade da aplicação de uma solução de <i>blockchain</i> dentro dos serviços de Cartórios de Notas Brasileiros sem a necessidade de alteração da legislação atual.
Principais funcionalidades	O usuário poderá realizar o upload de documento na plataforma, fazendo sua assinatura com seu certificado digital emitido e reconhecido pela Autoridade Certificadora Central e por fim o usuário pode incluir novas pessoas para fazer o processo de assinatura.
Ferramentas de desenvolvimento	Utilizou a rede de <i>blockchain</i> da Ethereum, escrevendo os contratos inteligentes em Solidity Foi utilizado para desenvolvimento o Truffle e o Ganache que são ferramentas para rodar a rede localmente e fazer testes. Para a interface Web foi utilizado o JavaScript e DApp.
Resultados e conclusões	O estudo realizado aprovou a viabilidade da aplicação respeitando os princípios Histórico, Disciplinares e Legais. A solução se mostra capaz, porém deve haver uma demanda da população e o reconhecimento dos órgãos públicos assim como levar em conta os custos envolvidos para incluir as transações dentro da rede Ethereum que possui uma taxa de transação e execução.

Trabalho Correlato 2

Referência	JUNQUEIRA, Natália Rodrigues. Concessão de permissão a dados de saúde baseada em contratos inteligentes em plataforma de blockchain. 2020. 90 folhas. Dissertação (Mestrado em Ciência da Computação) - Universidade Federal de Goiás, Goiânia.
Objetivos	<p>Explorar o estado da arte na tecnologia <i>blockchain</i> para os dados de saúde usando contratos inteligentes.</p> <p>Desenvolver uma aplicação onde o paciente possa controlar a permissão dos seus dados pessoais de saúde.</p> <p>Analisar as plataformas Ethereum e Hyperledger Fabric como possíveis ferramentas para se desenvolver.</p>
Principais funcionalidades	<p>O cadastramento do paciente dentro da rede <i>blockchain</i>.</p> <p>O contrato inteligente para a concessão de permissão aos dados pessoais de saúde.</p> <p>O histórico de compartilhamento dos dados.</p>
Ferramentas de desenvolvimento	A ferramenta escolhida foi o Hyperledger Fabric para a implementação dos contratos inteligentes.
Resultados e conclusões	<p>A ferramenta escolhida foi o Hyperledger Fabric pois ele permite a configuração de permissão de cada organização, sendo assim não sendo uma <i>blockchain</i> pública, com dados expostos para todos.</p> <p>O projeto conseguiu alcançar os objetivos colocando como ressalva a criptografia dos dados e o baixo desempenho da rede por estar sendo executada em uma máquina virtual.</p>

Trabalho Correlato 3

Referência	ABREU, Antônio Welligton dos Santos. Uma abordagem baseada em blockchain para armazenamento e controle de acesso aos dados de certificados de alunos do ensino superior. 2020. 146 folhas. Dissertação (mestrado) - Universidade Federal do Ceará, Quixadá.
Objetivos	Desenvolver uma aplicação onde seja possível publicar e validar diplomas. Aplicar um questionário para profissionais da área. Analisar o desempenho da aplicação construída.
Principais funcionalidades	Consultar a veracidade dos diplomas. Poder cadastrar novos diplomas e revogá-los. Poder fazer o cadastro no IES manualmente e automaticamente.
Ferramentas de desenvolvimento	Interface frontend chamado Educ-Dapp para a interação dos usuários, construído com JavaScript, HTML e CSS. A camada de blockchain, rede Ethereum, responsável por gravar os dados.
Resultados e conclusões	O sistema pode fornecer uma alternativa viável para armazenar e controlar os diplomas, pois os dados ficam armazenados de forma segura, aumentando a confiança e transparência na veracidade das informações.

Especificação

Requisitos

Ciclo de vida do carro

Requisitos

▪ Requisito Funcional - RF

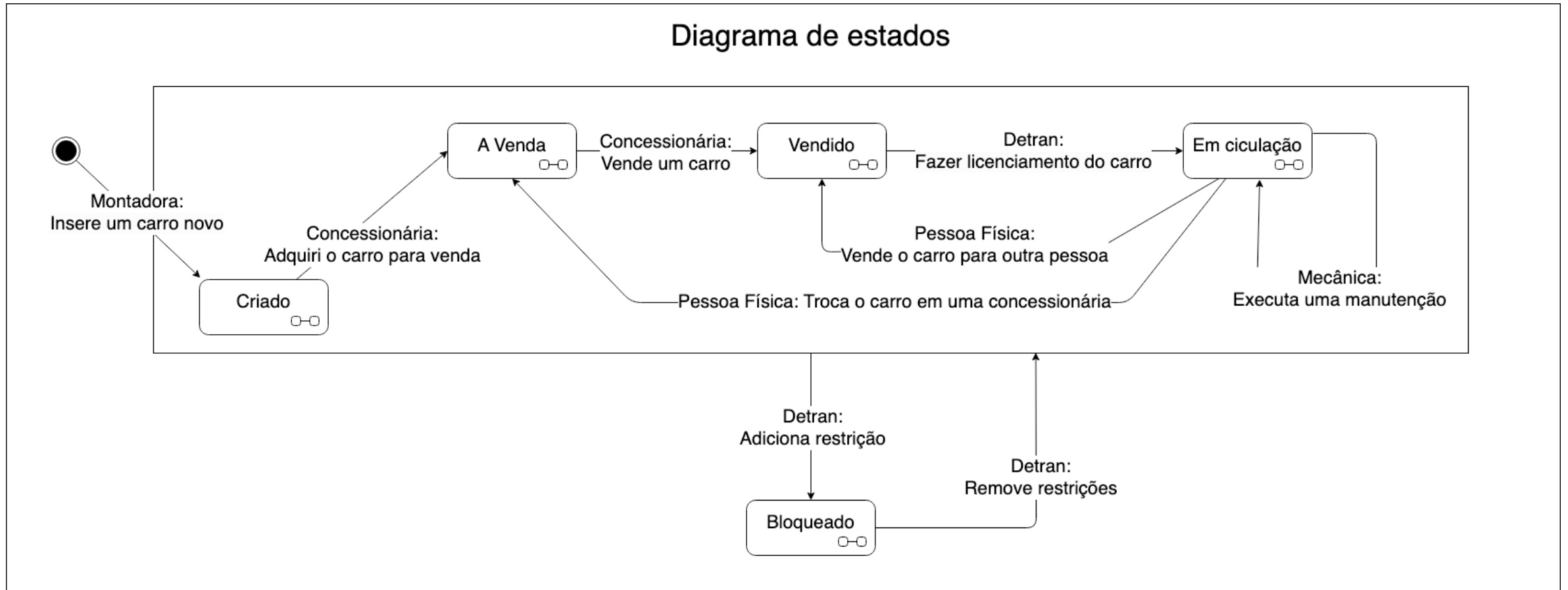
- a) armazenar o cadastro de um carro;
- b) permitir montadoras insiram novos carros;
- c) permitir concessionárias vendam um carro para uma pessoa física;
- d) permitir mecânicas adicionem manutenções efetuadas em um carro;
- e) permitir transferências de carros entre pessoas físicas;
- f) ser capaz de identificar transferências inválidas de carros entre pessoas;

Requisitos

- **Requisito Não Funcional – RNF**

- a) disponibilizar uma API que se comunique com a *blockchain*;
- b) disponibilizar uma interface web para interação;
- c) utilizar rede Hyperledger Fabric;
- d) utilizar contratos inteligentes para definir regras das transações.

Ciclo de vida de um carro



Implementação

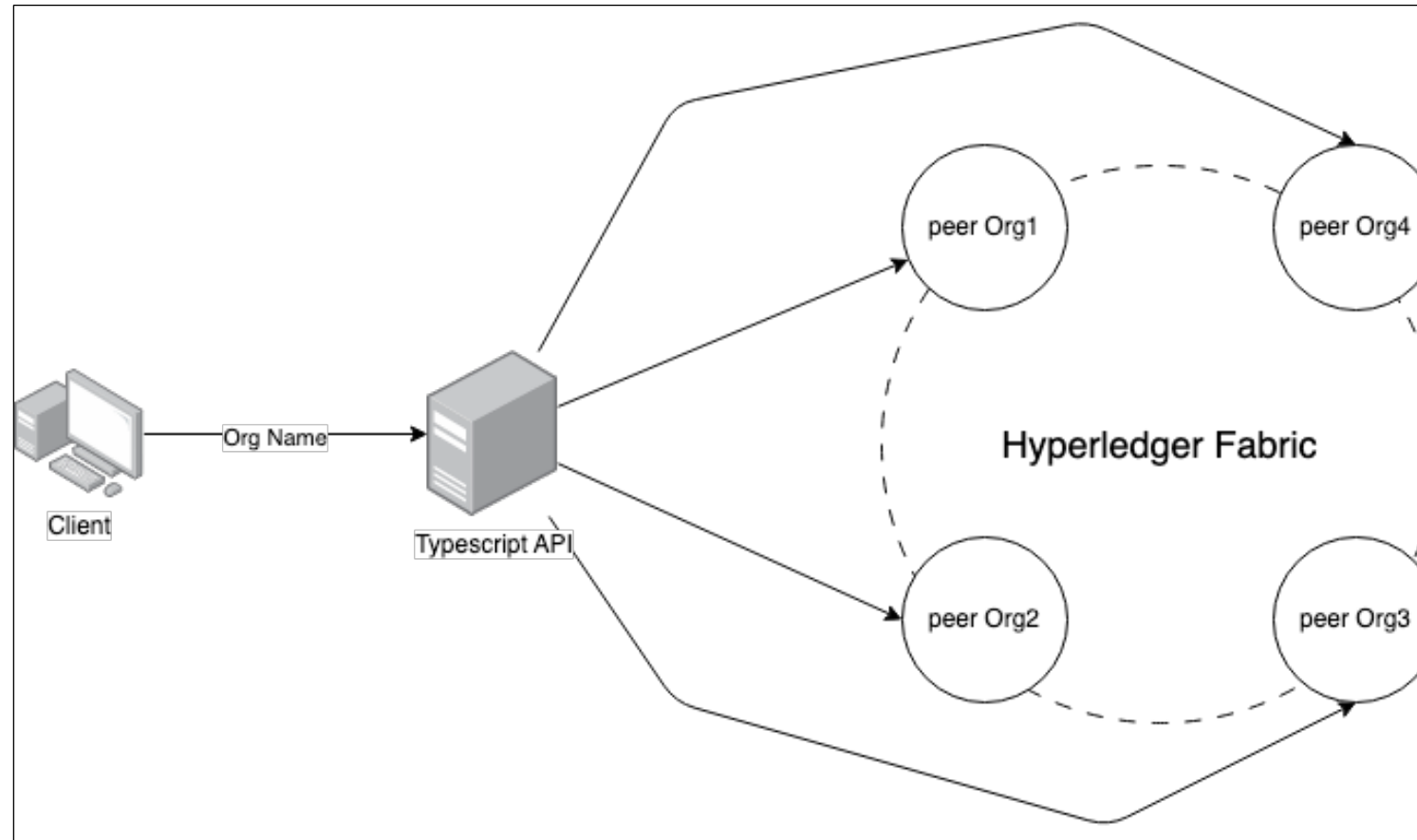
Arquitetura

Contratos inteligentes

API

Interface

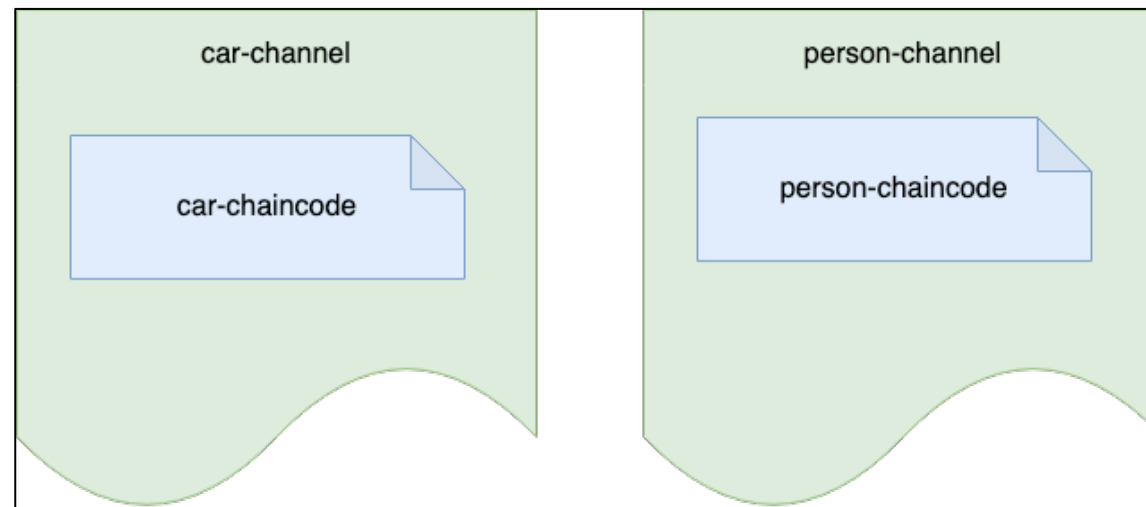
Arquitetura do protótipo



Implementação dos contratos inteligentes

- Desenvolvidos em Typescript
- Desenvolvidos com toda a validação dos dados e da organizações permitidas para executar a transação
- Conseguimos executar transações de outros canais e outros chaincodes

Estrutura dos canais e contratos inteligentes



Exemplo de métodos

```
64 ✓ export class BaseContract extends Contract {  
65  
66     protected async PutState(ctx: Context, key: string, value: object): Promise<void> {  
67         await ctx.stub.putState(key, Buffer.from(stringify(sortKeysRecursive(value))));  
68     }  
69  
70 ✓ protected async HasState(ctx: Context, key: string): Promise<boolean> {  
71     const stateJSON = await ctx.stub.getState(key)  
72     if (!stateJSON || stateJSON.length === 0) {  
73         return false;  
74     }  
75  
76     return true;  
77 }  
78  
79 ✓ protected async GetState(ctx: Context, key: string): Promise<any> {  
80     const stateJSON = await ctx.stub.getState(key)  
81     if (!stateJSON || stateJSON.length === 0) {  
82         throw new Error(`The object with key=${key} does not exist`);  
83     }  
84  
85     return JSON.parse(stateJSON.toString());  
86 }  
87
```

API de comunicação com a blockchain

- Desenvolvido em Typescript com framework express, usando biblioteca oficial do Hyperledger para conexão com a rede
- Comunicação com a *blockchain* é via Google Remote Procedure Call (gRPC)
- Ajuda na comunicação entre a *blockchain* e outros clientes utilizando o protocolo HTTP
- Recebe a organização que será usada para executar a transação dentro da *blockchain*

API de comunicação com a blockchain

PUT

▼

{{URL_3000}}/submit/person-channel/person/CreatePerson

Params

Authorization

Headers (10)

Body ●

Pre-request Script

Tests

Settings

☐ none

☐ form-data

☐ x-www-form-urlencoded

☒ raw

☐ binary

☐ GraphQL

JSON ▼

1

["111.222.333-44", "Yuri Hartmann", "06/07/2001", "marilda"]

Headers

👁 9 hidden















	Key	Value
✓	X-API-Key	gov

Interface

- Foi construída em Typescript com o framework React
- Mesma interface para todas as organizações
- Massagens de erros

Listagem de carros

Pesquisar... INSERIR CARRO +

Ações	ID do chassi	Marca	Modelo	Cor	CPF do dono	Concessionária	Ano	Financiado por
   	028191f9-3a9...	montadoraC	model 1	blue	333.333.333-33	Já possui um dono!	2023	+
   	02899732-9a9...	montadoraC	model 1	blue	-		2023	+
   	0546828c-8d2...	montadoraC	model 1	blue	-		2023	+

! No valid responses from any peers. Errors: peer=peer0.montadora-d.car-lifes-cicle.com:7081, status=500, message=The organization mecanicaK not allowed for this method

Selecione a organização

- detran
- gov
- montadoraC
- montadoraD
- concessionariaF
- concessionariaG
- mecanicaK
- mecanicaL
- financiadoraR

Análise dos Resultados

- Modularidade da rede Hyperledger Fabric
- Contratos inteligentes conseguiram atender os cenários e as regras de negócio
- A API facilitou bastante a execução das transações, simplificando o processo
- Com o protótipo conseguimos utilizar o potencial que a Hyperledger Fabric tem a oferecer

Comparativo com os correlatos

Trabalhos Correlatos Características	Menezes (2020)	Junqueira (2020)	Abreu (2020)	Protótipo
Entidades que gerencia	Documentos de cartórios	Dados pessoais de saúde	Diplomas do ensino superior	Ciclo de vida do carro
Rede de blockchain	Ethereum	Hyperledger Fabric	Ethereum	Hyperledger Fabric
Tem custo para fazer transações	Sim	Não	Sim	Não
Modo de participação dos pares	Público	Privado	Público	Privado
Interface	Aplicação Web	Aplicativo Móvel	Aplicação Web	Aplicação Web
Linguagem de programação do contrato inteligente	Solidity	Golang	Solidity	Typescript

Cenário do teste de velocidade

- Entender como a rede *blockchain* do Hyperledger Fabric lida com número de entidades, numero de requisições simultâneas e tipo de transações.

Número de entidades	Requisições simultâneas	Transações
<ul style="list-style-type: none">- 1.000- 10.000- 100.000- 1.000.000	<ul style="list-style-type: none">- 1 req/simultânea- 30 req/simultâneas	<ul style="list-style-type: none">- Ler dados do carro- Inserir um carro novo- Fazer manutenção no carro

Resultado do teste de velocidade

Resultado Ler dados do carro

Número de carros na rede	(a) Executando 1 requisição (milissegundos)	(b) Executando 30 requisições simultâneas (milissegundos)	Diferença entre volume de requisições simultâneas (b) / (a)
1.000	2163	2619	+ 21%
10.000	2165	2589	+ 19,5%
100.000	2156	2632	+ 22%
1.000.000	2186	2553	+ 16,7%

Resultado Inserir um carro novo

Número de carros na rede	(a) Executando 1 requisição (milissegundos)	(b) Executando 30 requisições simultâneas (milissegundos)	Aumento de tempo entre volume de requisições simultâneas (b) / (a)
1.000	2180	3387	+ 55,3%
10.000	2182	3419	+ 56,6%
100.000	2179	3418	+ 56,8%
1.000.000	2179	3377	+ 54,9%

Limitação encontrada

- No método “adicionar manutenção” encontramos limitação
- 1 requisição simultânea sem problemas
- Mais de 1 requisição simultânea apresentava o problema
- Tentativa de atualização de um mesmo registro por várias transações ao mesmo tempo
- A rede não é adequada para altas taxas de execução de atualizações de registro
- Caminho seria criar outro contrato inteligente que apenas cuidasse das manutenções

Conclusão

- Objetivos Alcançados
 - Modelar o ciclo de vida do carro
 - Criar rede *blockchain* privada com Hyperledger Fabric
 - Construir contratos inteligentes para gerir as regras de negócio envolvendo o carro
- Se mostra uma solução viável no mundo real
 - Unifica os sistemas sem um único ponto de falha
 - Garantia dos dados
 - Histórico dos dados

Sugestões

- Segregação mais forte por canais e contatos diferentes para evitar as limitações encontradas.
- Capacidade de enviar imagens para a rede, muitos processos dependem de imagens
- Aumentar o escopo de veículos e organizações envolvidas no ciclo de vida

Referências

- ABREU, Antônio Welligton dos Santos. **Uma abordagem baseada em blockchain para armazenamento e controle de acesso aos dados de certificados de alunos do ensino superior.** 2020. 146 f. Dissertação (Programa de Pós-Graduação em Computação) - Universidade Federal do Ceará, Quixadá.
- JUNQUEIRA, Natália Rodrigues. **Concessão de permissão a dados de saúde baseada em contratos inteligentes em plataforma de blockchain.** 2020. 90 folhas. Dissertação (Mestrado em Ciência da Computação) - Universidade Federal de Goiás, Goiânia.
- MENEZES, Leonardo Dias. **Blockchain e cartórios: uma solução viável?.** 2020. 61 f. Dissertação (Mestrado em Ciências) - Universidade de São Paulo, São Paulo.