

Aplicação de Blockchain em Sistemas Corporativos: Um Estudo Prático

Gabriel Mazilão Ferreira da Silva¹

Centro Universitário Academia, Juiz de Fora, MG

Gabriel Monteiro da Fonseca²

Centro Universitário Academia, Juiz de Fora, MG

Gustavo Sobreira Pinto³

Centro Universitário Academia, Juiz de Fora, MG

Jhonathan Meireles de Oliveira⁴

Centro Universitário Academia, Juiz de Fora, MG

Tassio Ferenzini Martins Sirqueira⁵

Centro Universitário Academia, Juiz de Fora, MG

Linha de Pesquisa: Engenharia de Software

RESUMO

Atualmente, na área da cadeia de abastecimento (*Supply Chain*), temos um grande mercado, onde existem várias empresas que sofrem com perda de dados ou desvio de mercadorias, por não terem uma auditoria adequada em suas aplicações. A tecnologia de supply chain que as empresas de logística estão utilizando atualmente precisam de maior robustez na questão de segurança, e isso é possível através da tecnologia de Blockchain. Algumas das soluções que temos hoje sofrem de fragmentação de dados e falta de proveniência confiável. É um grande desafio gerenciar a integridade de produtos e processos em um ambiente multistakeholder. Este trabalho se justifica pelo fato de Blockchain em SCM (*Supply Chain Management*) ser uma necessidade do mercado e talvez o futuro da tecnologia SCM. No contexto de Blockchain incorporado ao SCM, visamos fornecer transparência, rastreamento, sustentabilidade e proteção da marca, garantindo a autenticidade sustentável dos dados e, assim, adquirindo transparência dos processos dentro da empresa.

¹ Discente do Curso de Engenharia de Software do Centro Universitário Academia – UniAcademia.
E-mail: gabriel.silva@delage.com.br.

² Discente do Curso de Engenharia de Software do Centro Universitário Academia – UniAcademia.
E-mail: gabriel.fonseca@delage.com.br.

³ Discente do Curso de Sistemas de Informação do Centro Universitário Academia – UniAcademia.
E-mail: gustavo.pinto@dalege.com.br

⁴ Discente do Curso de Sistemas de Informação do Centro Universitário Academia – UniAcademia.
E-mail: jhonathan.oliveira@delage.com.br.

⁵ Docente do Curso de Engenharia de Software e Sistemas de Informação do Centro Universitário Academia. Orientador.



Palavras-chave: Blockchain, Supply Chain, Tecnologia, Segurança .

ABSTRACT

Currently, in the area of the supply chain, we have a large market, where several companies suffer from data loss or detour of goods, for not having an adequate audit in their applications. The supply chain technology that logistics companies are currently using needs more robustness on the security issue, and this is possible through blockchain technology. Some of the solutions we have today suffer from data fragmentation and a lack of reliable provenance. It is a big challenge to manage the integrity of products and processes in a multistakeholder environment. This work is justified by the fact that Blockchain in SCM (Supply Chain Management) is a market need and perhaps the future of SCM technology. In the context of Blockchain incorporated into SCM, we aim to provide transparency, traceability, sustainability, and brand protection, ensuring sustainable authenticity of data and thus acquiring transparency of processes within the company.

Keywords: Blockchain, Supply Chain, Technology, Security.

1 INTRODUÇÃO

Ao longo do estudo sobre o blockchain, percebemos a importância de desenvolver este artigo para explicar o funcionamento e implementação da tecnologia blockchain nas diferentes camadas do mercado, exemplificando e mostrando os árduos caminhos que devemos seguir atrás de maior segurança para nossas informações. As tecnologias atualmente têm um grande poder de transformar toda a nossa vida, facilitando a organização de tarefas ou até mesmo melhorias e agilidade no trabalho, além de trazer uma distração nos momentos de lazer. A blockchain foi criada para nos trazer mais segurança nas transações e uma melhor rastreabilidade, pois podemos rastrear todos os tipos de transações que são feitas dentro de uma blockchain. A ideia de ter uma rede descentralizada dispensa o trabalho de ter uma autoridade central, além de tornar as informações transparentes e igualitárias (ALENCAR e DE FRANÇA, 2022).

Segundo a documentação do Hyperledger Fabric⁶, o blockchain nada mais é do que um Distributed Ledger, que é responsável por conter todas as informações de transações ou até mesmo dados privados no caso de um blockchain privado. A ideia central do blockchain é replicar todos os registros em todos os computadores que compõem a rede, ou seja, cada um contribuirá para a manutenção do blockchain. Todas as informações que são inseridas na rede, passam por criptografia que torna os dados imutáveis e confirma a teoria do Ledger, cada blockchain tem seu sistema de criptografia que é chamado de sistema de prova. Para ter uma atualização mais consistente de nossas informações, utilizamos contratos inteligentes, responsáveis por encapsular todos os dados persistidos na blockchain. É de grande importância ressaltarmos a existência do Consensus, que garante a sincronização do ledger em nossa rede, além disso, nos permite ter segurança pois todos os participantes têm que aprovar qualquer modificação dentro do blockchain.

⁶ Hyperledger Fabric. Disponível em: <<https://hyperledger-fabric.readthedocs.io/pt/latest/blockchain.html#o-que-uma-blockchain>>. Acessado em 30 de nov. de 2022



2 REFERENCIAL TEÓRICO

Considerando que a tecnologia blockchain é uma das mais novas do mercado atual, ainda não temos um grande número de artigos relacionados a ela. Apresentaremos a seguir uma fundamentação teórica para melhor compreensão do estudo, que será indispensável para entender porque as empresas estão buscando mais informações sobre essa tecnologia revolucionária.

2.1 Blockchain

Podemos dizer que a Blockchain é uma tecnologia bastante conhecida na área das criptomoedas, sendo a Bitcoin Blockchain a pioneira nesta nova tecnologia. Nasceu em 2008 para reduzir gastos duplos com transações financeiras e aumentar a segurança, não se sabe quem foi o criador, alguns especulam ser uma pessoa ou um grupo sob o pseudônimo, Satoshi Nakamoto. Sendo uma rede que funciona como uma cadeia de blocos muito seguros que contém os dados como uma impressão digital que chamamos de hash, o bloco subsequente desta cadeia sempre conterá o hash do anterior, e se um hash for invalidado no meio da cadeia, toda a estrutura à sua frente será inválida. Com isso, podemos ter mais confiança sabendo que o ledger distribuído é um ledger que pode ser compartilhado por toda a rede, no caso de um blockchain público. No caso de empresas privadas que estão adotando essa tecnologia, temos as blockchains privadas, que exigem permissão de acesso privado para leitura dos dados contidos nas blockchains.

Segundo (GREVE *et al.*, 2018) confirmamos que o blockchain original é o do Bitcoin, que faz uma incorporação de estados muito simples, apesar de ter a ideia inicial de ser usado fundamentalmente em transações, podemos usar o blockchain em praticamente qualquer sistema de informação que gera algum tipo de dado. Vários projetos robustos estão sendo desenvolvidos com base em blockchains. Esses projetos são privados e alguns usam principalmente a tecnologia Hyperledger Fabric.

Por definição a IBM⁷ define o funcionamento de uma blockchain como um **livro-razão imutável e compartilhado**, também diz que praticamente qualquer item de valor pode ser controlado e comercializado em uma rede blockchain.

Segundo o blog Contabilix⁸, para contabilidade, **livro-razão** é o lugar onde há o registro em ordem cronológica de todas movimentações registradas, com este livro podemos ter o controle de todos os saldos e movimentações de uma empresa, onde é possível ter um grande acompanhamento de todas operações realizadas em determinado período.

⁷ IBM: Disponível em: <<https://www.ibm.com/br-pt/topics/what-is-blockchain>>.

Acessado em 30 de nov. de 2022

⁸ Blog Contabilix: Disponível em: <<https://www.contabilix.com.br/contabilidade-online/livro-razao-na-contabilidade-qual-o-objetivo-e-principais-caracteristicas/>>.

Acessado em 30 de nov. de 2022



Razão Analítico

XPTO XXX

Data: 17 / 10 / 2022

CNPJ: XXXXXXXX/XXXX-XX

Período: 15 a 17 / 10 / 2022

DATA	RE F	Contrapartida	Histórico	Débito	Crédito	Saldo
15 / 10 / 2022	1		Saldo inicial			25.00
16 / 10 / 2022	2	Estoque	Valor ref a NF N° XXX		10.00	15.00
17 / 10 / 2022	3	Caixa	Vendas feitas	15.00		30.00
			Totais	15.00	10.00	30.00

Fonte: Elaboração Própria.

Mas se colocarmos esse arquivo em um banco de dados, não importa o quão seguro seja, ainda é mutável; qualquer malfeitor que tenha acesso ao banco de dados pode alterar os dados desta tabela, pois não há nada que diga imediatamente que esta tabela foi alterada, e o erro pode durar dias ou meses sem que ninguém perceba a alteração.

A chave da resolução desta falha de segurança pode ser encontrada nas blockchains, podemos dizer que seu surgimento foi em um artigo de 2008 "Bitcoin: uma aplicação de finanças eletrônico peer-to-peer"⁹ desenvolvido por Satoshi Nakamoto como já dito anteriormente. Já que uma vez que algo é escrito em uma blockchain recebe um hash (por enquanto, entenda hash como se fosse um CPF dado ao conteúdo), quando este hash é gerado, qualquer alteração do conteúdo implica em um hash totalmente diferente. Utilizaremos a ferramenta de criptografia online, ANDTI¹⁰, onde podemos gerar hash a partir de um texto qualquer e testar diferentes formas de criptografia, sendo as mais conhecidas o SHA-1 (usado no Bitcoin) e o SHA-256 (usado no Ethereum), para que tenha um melhor entendimento, a seguir deixamos uma sequência de imagens que mostram essa mudança nos hashes.

⁹ Nakamoto, Satoshi. bitcoin.org. p 1-8. <https://bitcoin.org/bitcoin.pdf>
Acessado em 30 de Nov. de 2022

¹⁰ ANDTI: Disponível em: <<http://andti.com.br/tool/hash>>.
Acessado em 01 de Dez. de 2022



Exemplo 1 (com exclamação no final):

Mensagem

Testando a mudança do hash!

MD5 SHA-1 SHA-256 SHA-512 Whirlpool Gost

HASH

8e1fae59599a716adef058b2bae0eab73d85f259ed2ca499f4fbca7e30cd1a08

Fonte: <http://andti.com.br/tool/hash>.

Exemplo 2 (sem exclamação no final):

Mensagem

Testando a mudança do hash

MD5 SHA-1 SHA-256 SHA-512 Whirlpool Gost

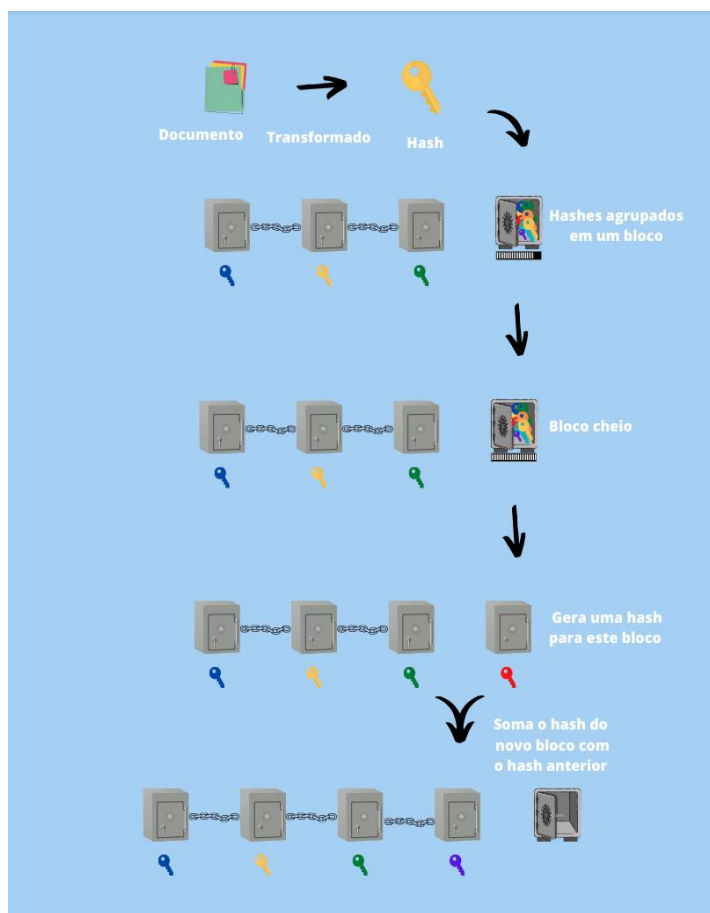
HASH

f6545e82cec5d12c83da6958398538181dabeb7ae3854dd9c0779c751e60f6ad

Fonte: <http://andti.com.br/tool/hash>.

Como podemos ver, uma simples mudança de pontuação gera um hash completamente diferente, pois da forma como essa criptografia é feita, cada símbolo recebe um valor que interfere diretamente no resultado da expressão, esse resultado recebe o nome de hash (hashes são bits, geralmente em base hexadecimal, que permite a visualização em letras e números (0 a 9 e A a F))¹¹, mas só gerar o hash não resolve o problema de alteração, pois ainda podemos mudar a tabela e o máximo que vai acontecer é a mudança de hash, para isso não acontecer esse hash é enviado para uma espécie de cofre, onde também são enviados vários outros hashes, até esse cofre encher, quando isso ocorre é fechado e recebe uma "chave" em forma de hash, essa chave é feita a partir da soma de todos os hashes dentro do cofre juntamente com o hash do cofre anterior (chamado de block). Um exemplo disto é apresentado na Figura 1.

Figura 1. Exemplos de Hashs em bloco.



Fonte: Elaboração Própria.

¹¹ Terminal Root: Disponível em: <<https://terminalroot.com.br/2019/05/o-que-e-e-como-gerar-uma-hash.html>>. Acessado em: 01 de Dez. de 2022



O último bloco (block) cria um elo com seu bloco anterior, este elo recebe o nome de “chain”, por isso o nome BlockChain, que por sua vez também cria um elo para seu antecessor, esta ligação entre os blocos gera uma reação em cadeia.

Para que todos estes blocos sejam vistos como algo seguro precisam ser validados, ou seja devem atender a um determinado requisito imposto pela própria blockchain (esse requisito varia de blockchain para blockchain), por exemplo, todo bloco válido nessa blockchain deve começar com a seguinte ordem 0000, então para que o bloco torna-se válido passa por várias somas até que o hash atenda a condição desejada, pessoas que fazem esse tipo de validação são chamados de mineradores, pois em blockchain públicas, recebem uma parte da transação.

Figura 2. Informações do bloco.

The image shows a web interface for mining a blockchain block. It features several input fields and a button:

- Bloco:** A field with a '#' icon and the value '1'.
- Nonce:** A field with the value '45219'.
- Dados:** A large text area containing the text '1º Bloco também chamado de Bloco Gêneses'.
- Prévio:** A field containing a long string of 32 zeros.
- Hash:** A field containing the hexadecimal string 'aa875f29caf71b7bfe53266ec76f294736fe70badb2dc7efaf03b9b488a3c169'.
- Minerar:** A blue button located below the Hash field.

Fonte: <https://andersbrownworth.com/blockchain/blockchain>.

Note (Figura 2) que o hash atual começa com “aa875...” não atendendo assim a condição imposta, indicando então que o bloco ainda precisa ser minerado.



Figura 3. Validação do bloco.

Bloco: # 1

Nonce: 122950

Dados: 1º Bloco também chamado de Bloco Gênese

Prévio: 00

Hash: 0000f29b20c5fa0f3fe21367aa63aeb295a1f3e04aa0266835e3ac2224025c3

Minerar

Fonte: <https://andersbrownworth.com/blockchain/blockchain>.

Note (Figura 3) que após a validação do bloco, seu hash atende à condição imposta pela blockchain. Agora todos os blocos estão ligados e validados, caso algum arquivo for alterado dentro do bloco de chave amarela (segundo bloco da Figura 3) por exemplo, o seu hash, mudaria tornando-o inválido, porém como já existem blocos ligados a este bloco todos seus sucessores também são invalidados, causando um erro, já que quanto maior a blockchain maior será o poder computacional para que o malfeitor valide todos os blocos antes de ser detectado, por isso é chamado de **livro-razão imutável**.

Na imagem a seguir (Figura 4), todos os blocos estão devidamente validados, faremos uma pequena mudança no primeiro bloco, note a reação em cadeia dos demais (Figura 5).



Figura 4. Bloco validados.

Bloco	Nonce	Hash
# 1	122958	0000f29b29c5fa0f3fe21367aa63eeb295a1f3e04aa026835c3ac224825c3
# 2	38360	000027c0534976461e8649e7688bec5454e8e8035a4a0b15275fc76da4f9
# 3	46734	00007a86be835e87b23c6b87093cfc0d0549b1047e5313408

Fonte: <https://andersbrownworth.com/blockchain/blockchain>.

Figura 5. Blocos invalidos.

Bloco	Nonce	Hash
# 1	122958	697652a9262e144d38e4cf1d4243f9991a3e5f74b6187025cb86a1222f1105
# 2	38360	ac27378ea5d6e27e63e1955e9537ba497037e0bfa98a66df6683bccd8d74
# 3	46734	0ea5e0eccc32a8b4f769511baca956fb717f59208f354453

Fonte: <https://andersbrownworth.com/blockchain/blockchain>.

Agora em uma alteração no segundo bloco (Figura 6), nos permite enxergar de melhor forma esta reação em cadeia, uma vez que apenas o bloco que teve seu conteúdo alterado e seus subsequentes tiveram o hash invalidado.

Figura 6. Bloco invalido no meio da hash.

Bloco	Nonce	Hash
# 1	53506	00000fe4848e4d5aeb0da0e931fa51bb2a4b7f372bb3bc298e2488e7691144f
# 2	32312	f4db182e7bd25b7f85da6a2b3345206b272e9c1efa18a566713970f1cb47399
# 3	20812	c1fc5077a7d8446683e99fb644ec0727c79844854ab2309

Fonte: <https://andersbrownworth.com/blockchain/blockchain>.

Se esse nível de segurança não for suficiente, lembre-se que sua definição é um **livro-razão (ledger) imutável e compartilhado**, o que significa que esse banco de dados não está em apenas um lugar, pode estar na rede de forma pública como Bitcoin, Ethereum,



etc... Mas também pode estar na rede de forma privada, onde as empresas utilizam máquinas proprietárias para alocar a rede blockchain. Tendo o banco de dados de forma descentralizada, ganhamos um nível extra de segurança, pois, caso ocorra uma alteração de arquivos em uma das máquinas que estão na rede blockchain (chamadas de nodes), as demais (nodes) irão apontar a incompatibilidade no hash e o node que tiver a divergência, terá seu banco de dados restaurado a partir de uma cópia dos demais nodes. Agora, além de necessitar de muito poder computacional para validar todos os blocos do alterado, o malfeitor também precisará alterar 51% dos nodes que estão na rede, para que assim tenha a maioria simples e sua alteração torne-se aceita pelos demais nodes.

Com o crescimento do banco de dados e expansão do número de nodes, para que uma falha de segurança advinda do software acontecer torna-se quase impossível, segundo o site: Portal do Bitcoin¹², a cripto no mês de Janeiro de 2021 bateu 320 GB de dados armazenados, contido em milhares de nodes pelo mundo, então ao contrário de muitas outras soluções de segurança que com o tempo torna-se mais vulneráveis uma blockchain com o passar dos anos tende a ter sua segurança ainda maior.

2.2 Contratos Inteligentes

O Contrato Inteligente (Smart Contracts), inaugurado em 1996, é um algoritmo codificado para rodar de forma automática e autônoma dentro de um banco de dados, que visa se interligar com o blockchain e garantir maior segurança e praticidade a todos que estão participando da rede. Especula-se que a aplicação do blockchain aos Smart Contracts pode trazer benefícios, mas pode ser prejudicial por se tratar de uma tecnologia nova e pouco desvendada proporcionando, a ideia dos smart contracts veio de um sistema de máquinas automatizadas de venda, que o nome já diz, era uma operação automática, onde o produto era enviado assim que o contrato reconhecia o dinheiro. Ou seja, quem depositasse o dinheiro poderia receber o produto. O ponto mais importante seria a verificação e segurança desses contratos, e a rapidez com que cada contrato é executado.(DE CARVALHO e ÁVILA, 2019).

No entanto, podemos considerar que os contratos têm um valor significativo, e geralmente podem ser adicionados a essa tecnologia blockchain, que funcionará como um banco de dados para ingerir as informações do contrato e, claro, tornar a funcionalidade dessa rede automatizada e o banco de dados descentralizado.

3 METODOLOGIA

A pesquisa começou com o estudo sobre segurança e implementações do sistema blockchain, através de sites e fóruns, que nos possibilitaram um maior entendimento de

¹² Portal do Bitcoin: Disponível em: <<https://portaldobitcoin.uol.com.br/bitcoin-deve-resolver-problema-de-tamanho-do-blockchain-para-seguir-crescendo/>>.

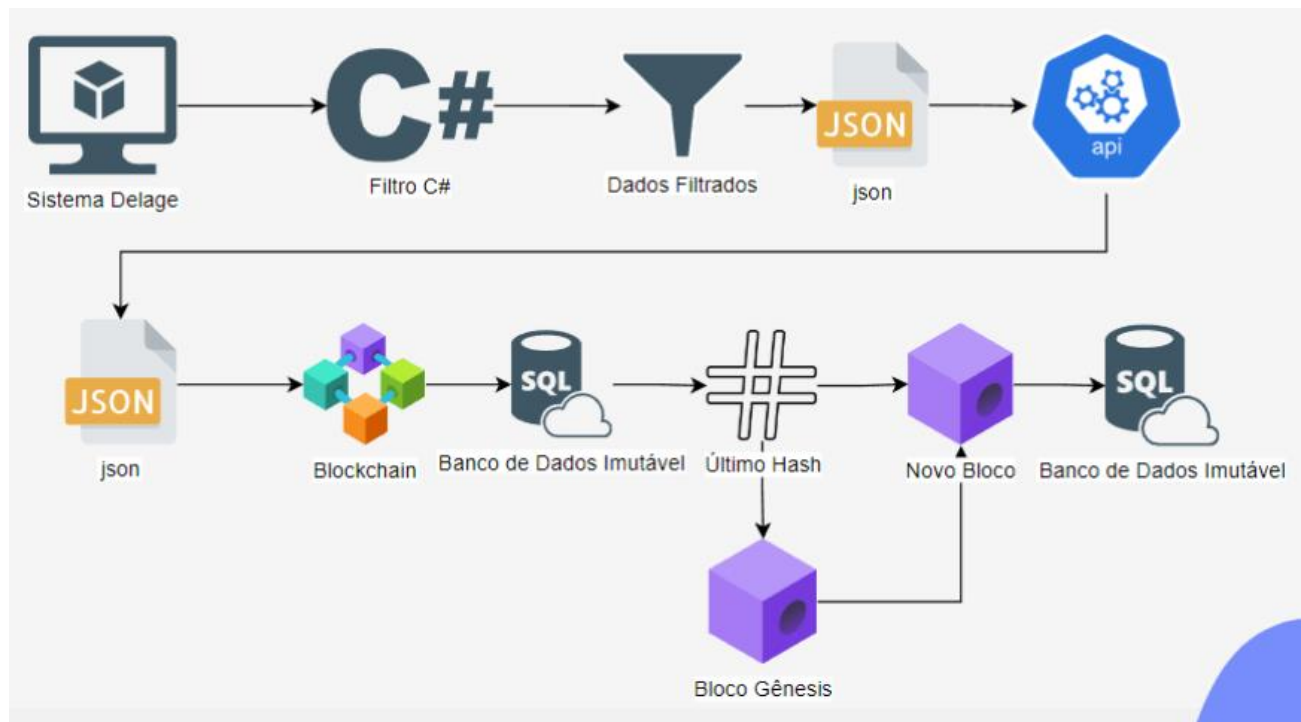
Acessado em 01 de Dez. de 2022



como usar este sistema em aplicações mais simples. Optamos primeiramente em criar uma API Blockchain na linguagem Python, por se tratar de uma linguagem alto nível, que poderíamos ter um ambiente de testes mais rápido, podendo focar os esforços em aprender a como construir o sistema, para os testes utilizamos também um banco de dados imutável, Azure Ledger Database, na tentativa de que ao usarmos um banco onde qualquer alteração feita é gerado um histórico imutável, obtendo assim mais segurança. Após a apresentação da API em Python, começamos os estudos de migração da linguagem Python para C#, uma vez que apresentou um menor tempo de resposta.

Os testes de implementação foram feitos em cima de um sistema de Crud. Ao final dos testes do projeto, apresenta-se a seguinte arquitetura (Figura 7):

Figura 7. Arquitetura da aplicação.



Fonte: Elaboração Própria.

Explicando a arquitetura (Figura 7), a ideia inicial era desenvolvermos um sistema de filtragem de dados em C#, que iria interligar com o sistema da Delage, e todos os dados que precisam ser inseridos na blockchain seria enviado através de um json para nossa API desenvolvida inicialmente em Python, sendo assim, a API seria responsável por manipular toda a informação passada, e enviar para um banco de dados imutável “Azure Ledger Database”.

O ciclo de desenvolvimento inicial foi planejado da seguinte forma:

- Passo 1 : Desenvolvimento de um CRUD com armazenamento de dados (Seria o sistema da delage, porém resolvemos criar o CRUD para familiarizarmos com a linguagem e para testes.)
- Passo 2 : Escolha de um banco de dados Imutável, logo encontramos o Azure

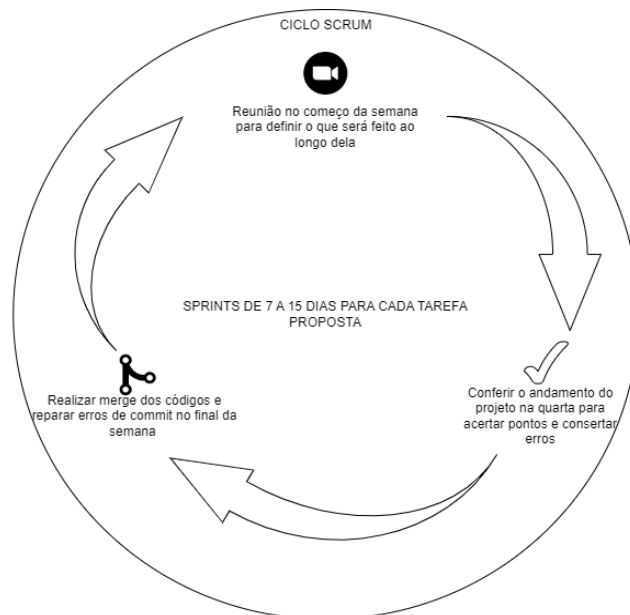


Ledger Database que torna todos os dados excluídos dele imutáveis, então temos registrado todos os dados de forma que não haverá adulterações.

- Passo 3 : Criação do filtro em C# para gerar auditoria do sistema, nesse passo foi desenvolvido por nossa equipe um filtro utilizando Action Filter Attribute do próprio .NET Framework com o objetivo de capturar todas as páginas que o usuário logado acessou, assim como, registrar todos os dados inseridos, excluídos ou editados no sistema. E só assim enviar essas informações em forma de json para nossa API em python.
- Passo 4 : Criação de um protótipo de Blockchain, então criamos a API em python para manipular toda informação recebidas do C# e criptografar trazendo a ideia de uma blockchain, mas sem os contratos inteligentes. Só para termos uma ideia de como realmente funciona uma blockchain.
- Passo 5 : Finalização e verificações, aqui nós usamos o SonarQube para fazer verificação de todo o código, removendo todos os códigos duplicados e maus cheiros de códigos (code smells).

Após isso, foi iniciada a criação da blockchain explorer, que é basicamente o front-end do sistema. Este período foi organizado com o método de scrum com daily todos os dias, e a utilização do Azure DevOps, foi extremamente importante para gerenciamento de tarefas, criação de work items, gerenciamento de repositórios e versionamento da aplicação. Esse ciclo gerencial pode ser visto na Figura 8.

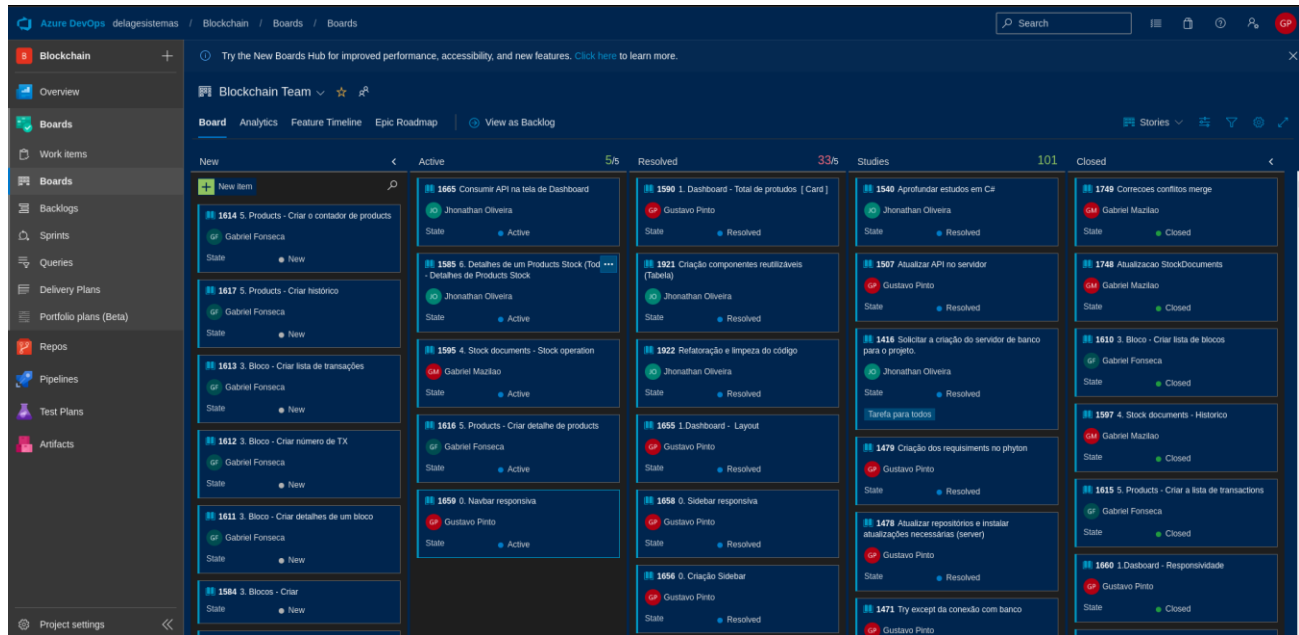
Figura 8. Ciclo dos sprints do projeto.



Fonte: Elaboração Própria.

Na Figura 9, podemos ver toda metodologia scrum que utilizamos para estar realizando as tarefas proposta para cada membro:

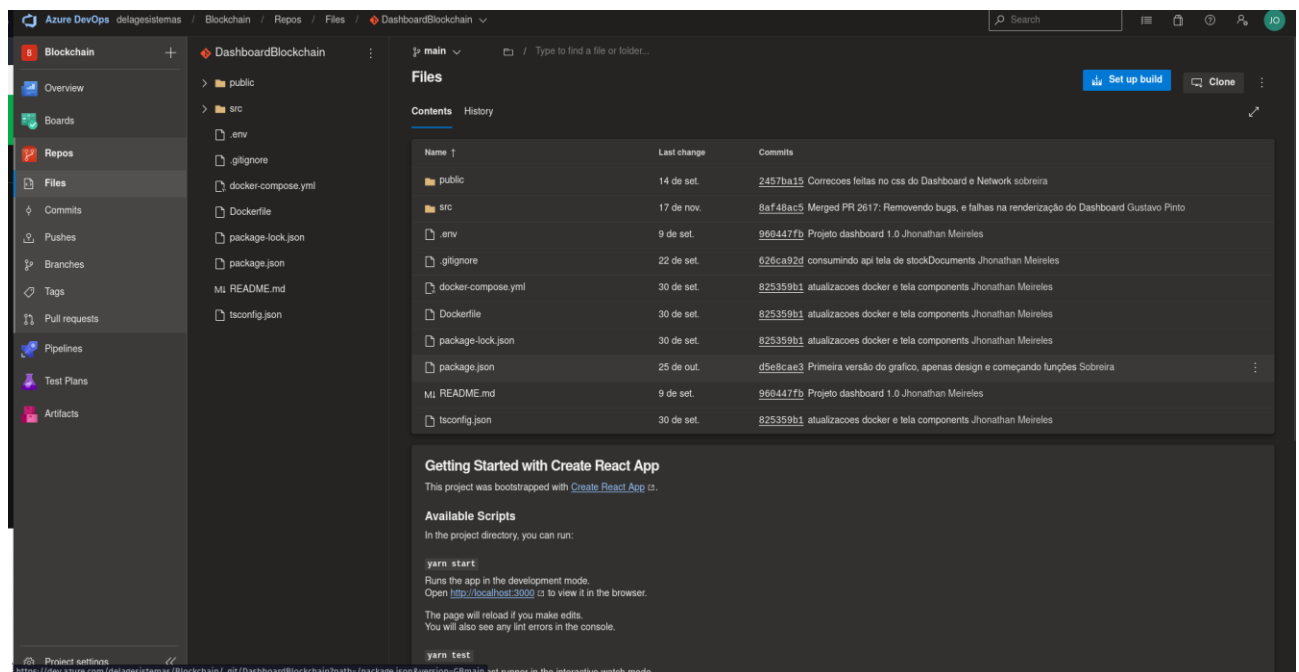
Figura 9. Quadro de Tarefas.



Fonte: Elaboração Própria.

Além disso, utilizamos o repositório do git que o próprio Azure DevOps disponibiliza, para fazermos o gerenciamento do desenvolvimento da aplicação, e gerenciar versões. Na Figura 10 podemos ver o repositório da aplicação:

Figura 10. Repositório GIT do projeto.



Fonte: Elaboração Própria.



A Figura 11 temos o gerenciamento de versões, com as releases que foram lançadas.

Figura 11. Versão lançadas do projeto.

Tag	Commit	Tagger	Creation Date
V1.0.0	c12c52e2	Jhonian Oliveira	16 de set.
V1.1.0	6cf298ad	Gabriel Fonseca	30 de set.
V1.2.0	af4f4516	Gustavo Pinto	17 de nov.

Fonte: Elaboração Própria.

Além disso, temos também a opção de analisar todos os commits realizados no projeto, como mostra a Figura 12.

Figura 12. Historico de *commits*.

Commit ID	Author	Date	Message	Count
730d5e8a	Jhonian Oliveira	23 de set. at 14:18	Criação tela de detalhesProductStock	2487
a130a004	Jhonian Oliveira	23 de set. at 13:48	configuracao api tela network	2487
1e0d8c92	Jhonian Oliveira	22 de set. at 14:15	consumindo api tela de stockDocuments	2487
626ca32d	Jhonian Oliveira	22 de set. at 12:11	criação tela transações e bloco	2487
51750911	Jhonian Oliveira	19 de set. at 21:20	Merged PR 2486: Stock Document Historico	2486
c2804900	Gabriel Fonseca	23 de set. at 14:08	Stock Document Historico	2486
74034197	Maziao	23 de set. at 14:06	Merged PR 2485: criacao da tela de detalhes de transacao	2485
65879348	Gabriel Fonseca	23 de set. at 14:05	criacao da tela de detalhes de transacao	2485
704fa166	unknown	23 de set. at 14:02	Merged PR 2484: Dashboard compatíveis a diversos navegadores	2484
e783a273	Gustavo Pinto	23 de set. at 13:54	pagina Dashboard sem nenhum bug visual e compatível para todos navegadores	2484
0a204217	sobera	23 de set. at 13:51	total de blocos criados	2484
e14e0eb4	Jhonian Oliveira	22 de set. at 15:11	teste	2484
89ee0935	Jhonian Oliveira	22 de set. at 14:23	Correcao de mais um bug na tabela de blocos e adição de uma barra quando passa o mouse na sidebar	2484
029a4016	sobera	22 de set. at 11:57	Remoção de bugs ref a tabelas	2484
a9332390	sobera	21 de set. at 16:59	Padronização de Navbar e Tables	2484
73088ee8	sobera	21 de set. at 15:58	Padronização maior entras as pastas ref a Dashboard e Tempalate	2484
850937c3	sobera	21 de set. at 13:00	Merged PR 2476: Dashboard em escala e termino de criacao de botoes	2476
0a270362	Gustavo Pinto	21 de set. at 10:44	Dashboard em escala e termino de criacao de botoes	2476

Fonte: Elaboração Própria.

4 Tecnologias e Frameworks

4.1 React:

Utilizamos o React JS uma biblioteca JavaScript para a criação de interfaces de usuário. O React por ser uma biblioteca front-end tem como objetivo facilitar a conexão entre

diferentes partes de uma página, utiliza a lógica de componentes reutilizáveis, que são utilizados para aproveitar o código, seguir padrões de projetos, manutenibilidade de código.

4.2 TypeScript:

O TypeScript entra com a função de solucionar uma questão muito específica: fazer com que o desenvolvimento de Javascript possa ser realizado em grandes escalas e para diversos tipos de aplicações diferentes. Possibilita a inclusão do paradigma de Programação Orientada a Objetos ao Javascript, e uma tipagem estática para as variáveis declaradas, a checagem de tipos serve para minimizar a possibilidade de erros causados por tipos de dados errados. Com a ideia de criação de componentes reutilizáveis, o principal que o React nos traz, e o desenvolvimento em grandes escalas do TypeScript, nós criamos uma tabela genérica que facilitará o desenvolvimento de outras futuras tabelas, tornando o desenvolvimento mais ágil e prático. O resultado podemos observar na Figura 13.

Figura 13. Registros da Blockchain.

Hash	Nº Bloco	Number of Tx	Previous Hash	Controller	Times
91c0d43638cbf32bcbcee73de31	1	551504869	0	Blockchain Started.	2022-11-08T23:40:26.7537454
6fae4320a8792d60eee1979815	2	183074058	91c0d43638cbf32bcbcee73de31	Testando dados	2022-11-08T23:40:26.9472269
33038e6fc26aa454605741cc81	3	1257467861	6fae4320a8792d60eee1979815	Hello Blockchain	2022-11-08T23:40:40.9041776
474e2548e8f0140ab383a080a	4	402961054	33038e6fc26aa454605741cc81	Hello Blockchain	2022-11-08T23:40:41.5242728
27d093e7336edc8dc688e4d19	5	454747183	474e2548e8f0140ab383a080a	Hello Blockchain	2022-11-08T23:40:41.617004
4bb03cef020c9b1d3b388fcd0e	6	1957888117	27d093e7336edc8dc688e4d19	Hello Blockchain	2022-11-08T23:40:41.7791774
cc4cd74845ebda2eed7148853	7	246678370	4bb03cef020c9b1d3b388fcd0e	Hello Blockchain	2022-11-08T23:40:41.901548
efad70041a27f52f86d87933e0e	8	884879717	cc4cd74845ebda2eed7148853	Hash anterior	2022-11-08T23:40:53.6851056

Fonte: Elaboração Própria.

E para utilização dessa tabela (Figura 13), o desenvolvedor deve criar uma Interface primeiramente contendo todas as informações que serão recebidas da API demonstrado a Figura 14.



Figura 14. Dados para criação do bloco.

```
JhJonathan Meireles, há 2 meses | 1 autor (Jonathan Meireles)
export interface IBlocks {      JhJonathan Meireles, há 2 meses • criação tela transações e bloco ...
  id: number;
  current_hash: string;
  last_hash: string;
  nonce: number;
  controller: string;
  times: number;
}
```

Fonte: Elaboração Própria.

Logo após isso, devemos criar a função que fará requisição na API e retornará todos os dados em formato de json (Figura 15):

Figura 15. Consulta de blocos.

```
const getAllBlocks = useCallback(async () => {
  const { status, data } = await HttpClient.getAllBlocks();
  if (status !== 200) throw new Error();

  setAllBlocks(data);
}, []);
```

Fonte: Elaboração Própria.

Com isso já podemos criar o cabeçalho (nomes de cada coluna) onde será passado todas as informações que serão renderizados na tabela, conforme Figura 16. Como resultado temos o componente de tabela criado na tela de Blocks (Figura 17).

Deve-se importar o componente de tabela criado, junto com a chamada da API contendo os dados JSON recebidos(data), e as colunas que do DataBlocks, para assim renderizar a tabela (Figura 17).

Figura 16. Informações para criação da tabela.

```
You, há 2 semanas | 1 autor (You)
import { ColumnDefinitionType } from '../components/Table';
import { IBlocks } from '../interfaces/IBlocks';
export const DataBlocks: ColumnDefinitionType<IBlocks, keyof IBlocks>[] = [
  {
    key: 'current_hash',
    header: 'Hash',
    width: 200,
  },
  {
    key: 'id',
    header: '№ Bloco',
    width: 125,
  },
  {
    key: 'nonce',
    header: 'Número de Tx',
    width: 125,
  },
  {
    key: 'last_hash',
    header: 'Previous Hash',
    width: 200,
  },
  {
    key: 'controller',
    header: 'controller',
    width: 200,
  },
  {
    key: 'times',
    header: 'Times',
    width: 200,
  },
];
```

Fonte: Elaboração Própria.

Figura 17. Criação da tela de blocos.

```
You, há 2 semanas | 2 autores (You and others)
import React, { useEffect } from 'react';
import { useBlocks } from '../services/hooks';
import ProgressBar from '../components/Loadable/ProgressBar';
import { Table } from '../components/Table';
import { DataBlocks } from '../Data/DataBlocks';

const progress = new ProgressBar({
  size: 4,
  color: '#3B82F6',
  className: 'progress-bar',
  delay: 100,
});

export const TableBlocks = () => {
  const { AllBlocks, getAllBlocks } = useBlocks();

  useEffect(() => {
    progress.start();
    getAllBlocks();
  }, [getAllBlocks]);

  if (AllBlocks) {
    progress.finish();
  }

  return (
    <div>
      <Table data={AllBlocks} columns={DataBlocks}></Table>
    </div>
  );
};
```

Fonte: Elaboração Própria.

4.3 Azure SQL Database Ledger:

O Database Ledger traz o poder da Blockchain para o banco de dados SQL. Tornando os dados no SQL invioláveis usando criptografia com os mesmos padrões de uma Blockchain. Fornecendo uma maneira para que partes externas que não hospedam esse banco de dados possam ter uma prova criptográfica de que os dados não foram adulterados. Sendo um recurso do Azure SQL Database, basta ativar em um de seus bancos e, em seguida, ter esse nível de evidência de adulteração disponível para seus aplicativos.

5 Desenvolvimento

A blockchain final está sendo desenvolvida pela empresa "CCONSENSUS¹³" utilizando o framework Hyperledger Fabric, além de utilizar tecnologias como Node.js com desenvolvimento em Typescript e Kubernetes para distribuir os nós das redes.

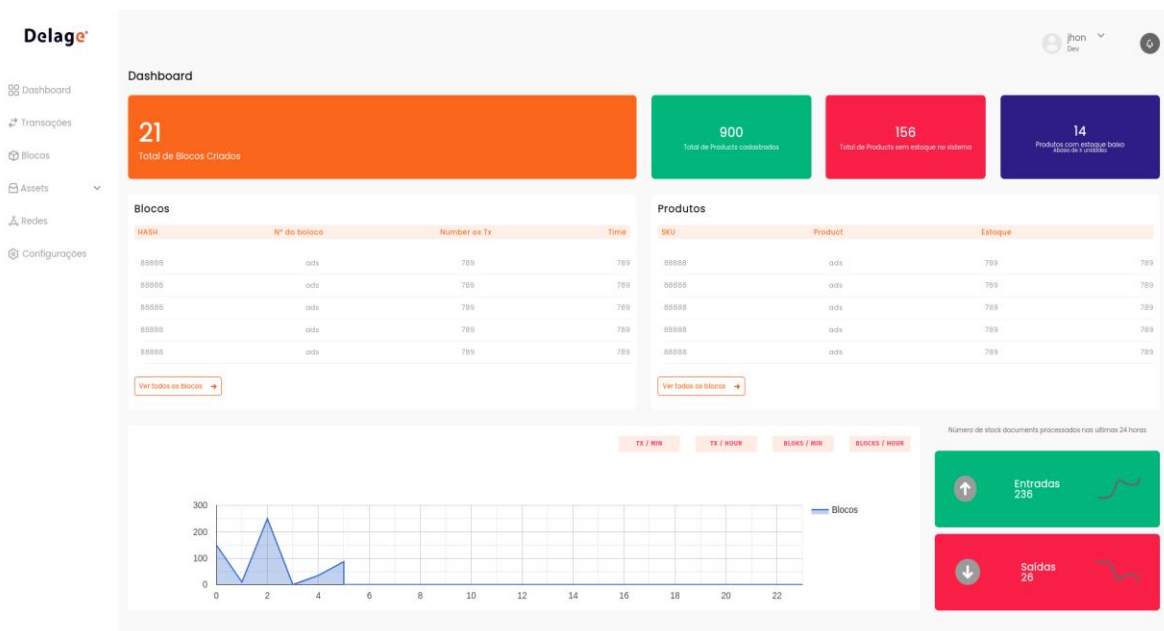
Iniciamos após muito estudo o desenvolvimento da blockchain explorer, que será o responsável por trazer todas as informações da rede, blocos criados, quantidades de blocos, ou seja, todas informações passados do back-end será expostos nesse explorer, e por ser dados sigilosos, nós protegemos esse explorer com uma autenticação.

Na Figura 18 podemos ver a tela principal do Blockchain Explorer.

¹³ CCONSENSUS. Disponível em: < <https://www.linkedin.com/company/cconsensus/> >. Acessado em 02 de dezembro de 22.



Figura 18. Dashboard Blockchain Explorer.



Fonte: Elaboração Própria.

4 CONSIDERAÇÕES FINAIS

Com a evolução do blockchain e a necessidade mundial de desenvolvimento de novas tecnologias, percebemos que os profissionais buscam incansavelmente melhorias em segurança, qualidade e desempenho. E percebemos a cada dia que devemos estar sempre atualizados com essas novas ferramentas que estão sendo desenvolvidas porque a tecnologia é uma área que está em constante mudança.

No processo de desenvolvimento encontramos algumas dificuldades no caminho, como:

- Criação de componentes: Como sabemos a tecnologia que usamos para desenvolver o front end da aplicação, nos permite criar componentes que serão reaproveitados ao longo do desenvolvimento, caso esses componentes não sejam criados de forma que possamos utilizá-los em outra parte do projeto, temos um problema, diante disso, tivemos um grande desafio em transformar componentes já desenvolvidos em componentes reutilizáveis, como chamamos na programação os componentes genéricos.
- Criação de Autenticação: Na parte de autenticação, temos ótimo desafio, pois proteger a aplicação é um trabalho árduo, e mesmo depois de proteger a aplicação não podemos dizer que estaremos 100% protegidos, por utilizar uma tecnologia de front-end que aprendemos recentemente, tivemos dificuldades em implementar uma segurança maior na aplicação.

Contudo, temos agora o desafio de continuar desenvolvendo mais funcionalidades no front end da aplicação (blockchain explorer), para tornar o sistema mais robusto e os componentes reutilizáveis, até o final do desenvolvimento do backend da blockchain, onde passaremos a administrar e aplicar nos demais sistemas da empresa.



AGRADECIMENTOS

A equipe de desenvolvimento do blockchain, agradece a empresa Delage por todo suporte técnico, financeiro e oportunidades de aprendizado desta nova tecnologia, e à UniAcademia pela oportunidade do projeto.

REFERÊNCIAS

ALENCAR, Levi Carmo; DE FRANÇA, Adriano Alves. **Blockchain: uma transformação no setor tributário e contábil**. Research, Society and Development, v. 11, n. 11, p. e235111133718-e235111133718, 2022.

DE CARVALHO, Carla Arigony; ÁVILA, Lucas Veiga. **A tecnologia Blockchain aplicada aos contratos inteligentes**. Revista Em Tempo, v. 18, n. 01, p. 156-176, 2019.

FRIEDRICH, Denise Bittencourt; PHILIPPI, Juliana Horn Machado. **Inclusão digital e Blockchain como instrumentos para o desenvolvimento econômico: Digital inclusion and Blockchain as instruments for economic development**. International Journal of Digital Law, v. 1, n. 1, p. 97-116, 2020.

GREVE, Fabíola; et al. **Blockchain e a Revolução do Consenso sob Demanda**. Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC)-Minicursos, 2018.

Blog Contabilix. **Livro razão na contabilidade: qual o objetivo e principais características**. Disponível em: <https://www.contabilix.com.br/contabilidade-online/livro-razao-na-contabilidade-qual-o-objetivo-e-principais-caracteristicas/>

IBM. O que é a tecnologia Blockchain. Disponível em: <https://www.ibm.com/br-pt/topics/what-is-blockchain>. Acessado em 02 de dezembro de 2022.

ANDTI. Gerador de Hash. Disponível em: <http://andti.com.br/tool/hash>. Acessado em 02 de dezembro de 2022.

Terminal Root. O que é e como gerar um hash?. Disponível em: <https://terminalroot.com.br/2019/05/o-que-e-e-como-gerar-uma-hash.html>. Acessado em 02 de dezembro de 2022.

Blockchain Demo. Disponível em: <https://andersbrownworth.com/blockchain/blockchain>. Acessado em 02 de



dezembro de 2022.

Amazon. AWS. Disponível em: <https://aws.amazon.com/pt/what-is/blockchain/>. Acessado em 02 de dezembro de 2022.

React. O que é, como funciona e porque usar e como aprender - Disponível em: <https://kenzie.com.br/blog/react/>. Acessado em 02 de dezembro de 2022.

TypeScript. Saiba tudo sobre a tecnologia - Disponível em: <https://programathor.com.br/blog/typescript/>. Acessado em 02 de dezembro de 2022.

____ O que são as tipagens estática e dinâmica em programação - Disponível em: https://www.alura.com.br/artigos/o-que-sao-as-tipagens-estatica-e-dinamica-em-programacao?gclid=Cj0KCQiAm5ycBhCXARIsAPIdzoUEteZ-n2-81PPZZhTy-qyPMfTB2faMW75qIndFVh9eqRoExXUhdToaAvnzEALw_wcB. Acessado em 02 de dezembro de 2022.

____ Azure SQL Database Ledger Getting Started and Examples - Disponível em: <https://www.mssqltips.com/sqlservertip/6890/azure-sql-database-ledger-getting-started-and-examples/>. Acessado em 02 de dezembro de 2022.

____ Introduction to Azure SQL Database Ledger for Data Tampering Protection - Disponível em: <https://www.mssqltips.com/sqlservertip/6889/azure-sql-database-ledger-data-tampering-protection/>. Acessado em 02 de dezembro de 2022.

MARCHINI, Daniela Maria Feltrin; DE CAMARGO JUNIOR, João Batista; PIRES, Silvio Roberto Ignacio. **Análise sobre a Contribuição da Tecnologia Blockchain na Gestão da Cadeia de Suprimentos**. Teoria e Prática em Administração, v. 10, n. 2, p. 154-168, 2020.

Satoshi Nakamoto. **Bitcoin: A peer-to-peer electronic cash system**. Consulted, 1:2012, 2008. URL <http://nakamotoinstitute.org/bitcoin/>