Chapter 1

Introdução aos Smart Contracts

Frank Cesár L. Verás e James D. Sousa

Abstract

Smart contracts are self-executing programs stored on blockchain, ensuring secure and transparent execution of digital agreements. This minicourse aims to present the fundamental concepts, advantages, applications, and challenges of smart contracts.

Resumo

Os smart contracts (contratos inteligentes) são programas autoexecutáveis armazenados em blockchain, garantindo execução segura e transparente de acordos digitais. Este minicurso tem como objetivo apresentar os conceitos fundamentais, vantagens, aplicações e desafios dos smart contracts.

1.1. Introdução

A transformação digital tem promovido profundas mudanças na forma como a sociedade interage, realiza negócios e gerencia informações. Nesse contexto, os contratos inteligentes, ou *smart contracts*, surgem como uma evolução natural das tecnologias baseadas em blockchain, proporcionando uma maneira automatizada, segura e transparente de executar acordos digitais.

A ideia dos contratos inteligentes foi introduzida pela primeira vez na década de 1990 por Nick Szabo, um criptógrafo e cientista da computação com interesse profundo em direito contratual. Em seu artigo seminal de 1994, Szabo definiu os *smart contracts* como protocolos computacionais capazes de formalizar, verificar e executar automaticamente os termos de um contrato, sem a necessidade de intermediários ou autoridades centrais.

Para ilustrar o conceito, Szabo utilizou o exemplo de uma máquina de venda automática. Ao inserir uma quantia específica de dinheiro e selecionar um item, a máquina realiza automaticamente a transação, entregando o produto sem intervenção humana. Essa

analogia descreve o funcionamento básico dos contratos inteligentes: uma condição pré-programada que, quando satisfeita, aciona automaticamente uma ação.

Apesar da inovação teórica, a proposta de Szabo permaneceu por anos sem aplicação prática, devido à inexistência de uma infraestrutura tecnológica descentralizada que garantisse integridade, imutabilidade e confiabilidade. Essa limitação foi superada com o surgimento do blockchain — especialmente com o lançamento da plataforma Ethereum, em 2015.

Idealizado por Vitalik Buterin, o Ethereum introduziu uma plataforma descentralizada programável, com suporte à linguagem Solidity, que permitiu o desenvolvimento de contratos inteligentes personalizados. Por meio do uso do blockchain como base de dados imutável e validada por consenso distribuído, os contratos inteligentes do Ethereum permitiram a criação de aplicações descentralizadas (*dApps*) com segurança e transparência inigualáveis até então.

Desde então, outras plataformas blockchain como Binance Smart Chain, Solana, Cardano e Avalanche também passaram a oferecer suporte a contratos inteligentes, cada uma com suas próprias linguagens de programação e mecanismos de consenso. Essa diversidade tecnológica ampliou as possibilidades de uso e impulsionou inovações em setores como finanças descentralizadas (DeFi), gestão de cadeias de suprimento, propriedade intelectual, sistemas eleitorais e governança digital.

O estudo dos contratos inteligentes torna-se, portanto, essencial para profissionais e pesquisadores que atuam nas áreas de tecnologia, direito, economia e segurança da informação. Este minicurso tem como objetivo apresentar os fundamentos teóricos, a evolução histórica, as plataformas atuais e as aplicações mais relevantes dos contratos inteligentes no cenário contemporâneo.

1.2. O que são Contratos Inteligentes

Contratos inteligentes, ou *smart contracts*, são programas de computador armazenados em uma rede blockchain que executam automaticamente termos de um acordo quando determinadas condições pré-definidas são satisfeitas. Diferentemente dos contratos tradicionais, que dependem da confiança entre as partes e da mediação de terceiros (como advogados, cartórios ou bancos), os contratos inteligentes utilizam lógica computacional para garantir a execução imutável e autônoma das cláusulas estabelecidas.

A essência de um contrato inteligente está na ideia de que "o código é a lei" (*code is law*). Ou seja, uma vez implementado e implantado na blockchain, o contrato age conforme o código que o define, sem a possibilidade de intervenção ou modificação externa. Isso garante transparência, previsibilidade e confiança na execução do acordo, já que todos os participantes podem verificar publicamente o seu funcionamento.

A estrutura básica de um contrato inteligente inclui:

- **Condições**: instruções que determinam quando e como uma ação será executada;
- **Ações**: operações automáticas que ocorrem quando as condições são satisfeitas (como transferir um valor, conceder acesso, registrar um dado, etc.);

- **Estado**: dados armazenados na blockchain que representam o contexto ou progresso do contrato;
- **Eventos**: notificações que informam à rede que uma determinada operação foi realizada.

Por estarem implantados em blockchains, os contratos inteligentes herdam suas características fundamentais: descentralização, imutabilidade, auditabilidade e resistência à censura. Isso significa que qualquer tentativa de adulteração, interrupção ou manipulação do contrato é tecnicamente inviável, a menos que haja consenso da rede — o que garante um alto nível de segurança.

Além disso, contratos inteligentes podem interagir com outros contratos e aplicações descentralizadas, formando ecossistemas complexos e autossustentáveis. Essa capacidade de composição é um dos fatores que torna os *smart contracts* fundamentais para o funcionamento da Web3 e das chamadas Finanças Descentralizadas (DeFi).

Em resumo, contratos inteligentes representam uma mudança de paradigma na forma como sistemas digitais podem ser programados para estabelecer confiança, reduzir custos e eliminar intermediários em processos automatizados e transparentes.

1.2.1. Benefícios dos Smart Contracts

Os smart contracts oferecem uma série de benefícios que justificam seu crescente uso em diversas aplicações, especialmente em ambientes que exigem confiança e automação. Esses contratos programáveis tornam possível a execução automática de acordos sem a necessidade de intermediários, o que contribui para maior eficiência e segurança. A seguir, destacam-se os principais benefícios proporcionados por essa tecnologia:

- **Autonomia**: Um dos principais atrativos dos smart contracts é a eliminação de intermediários. Como as cláusulas contratuais são autoexecutáveis e registradas diretamente no código, as partes envolvidas não precisam depender de terceiros, como advogados, cartórios ou instituições financeiras, para validar ou supervisionar o cumprimento do contrato. Isso proporciona maior independência e controle para os usuários, além de reduzir possíveis conflitos de interesse.
- **Segurança**: Os smart contracts são implantados em redes blockchain, que oferecem elevados níveis de segurança criptográfica. Uma vez inserido na blockchain, o código do contrato é praticamente imutável, tornando extremamente difícil qualquer tentativa de manipulação ou fraude. Além disso, a natureza descentralizada da blockchain reforça a resiliência do sistema, garantindo que as transações ocorram mesmo em situações de ataque ou falhas de infraestrutura.
- **Transparência**: Toda a lógica e as condições de um smart contract são públicas e auditáveis por qualquer usuário da rede. Isso significa que todos os envolvidos podem verificar o código e assegurar que ele funciona exatamente como prometido. Essa transparência reduz drasticamente o risco de mal-entendidos ou interpretações ambíguas, fortalecendo a confiança entre as partes e proporcionando maior previsibilidade na execução dos acordos.

- **Eficiência**: A automação proporcionada pelos smart contracts elimina etapas burocráticas e manuais, acelerando a execução de transações e processos. Além disso, como não há necessidade de intermediários, os custos operacionais são significativamente reduzidos. A execução instantânea e a redução de falhas humanas tornam os contratos inteligentes uma ferramenta poderosa para otimizar recursos e aumentar a produtividade em diversas aplicações, desde pagamentos automatizados até cadeias logísticas complexas.

1.3. Funcionamento Geral e Arquitetura dos Contratos Inteligentes

Os contratos inteligentes funcionam com base na tecnologia blockchain, que fornece a infraestrutura necessária para garantir a execução descentralizada, segura e imutável dos programas. Para compreender como esses contratos operam, é essencial analisar sua arquitetura básica, os componentes envolvidos e o fluxo de execução típico.

1.3.1. Estrutura Técnica

Um contrato inteligente é essencialmente um código de programação que define regras e instruções armazenadas em um bloco da blockchain. Após ser implantado (deploy) na rede, ele passa a existir de forma permanente e pode ser acessado e executado por qualquer usuário ou outro contrato.

A estrutura interna de um contrato inteligente geralmente é composta por:

- **Variáveis de estado**: armazenam dados persistentes, como saldos, registros ou status.
- **Funções**: definem as ações que podem ser executadas, como transferências, atualizações ou verificações.
- **Eventos**: notificações que informam sobre mudanças ou execuções ocorridas.
- **Modificadores e restrições**: limitam o acesso a determinadas funções ou condições de execução.

A linguagem mais utilizada para escrever contratos inteligentes é a **Solidity**, que foi desenvolvida especificamente para a blockchain Ethereum. Outras linguagens populares incluem *Rust* (utilizada em Solana), *Michelson* (Tezos) e *Move* (Aptos e Sui).

1.3.2. Fluxo de Execução

O funcionamento de um contrato inteligente pode ser resumido nas seguintes etapas:

1. **Implantação (deploy)**: o contrato é compilado e enviado para a blockchain, gerando um endereço único.
2. **Interação**: usuários ou outros contratos interagem com ele por meio de transações.
3. **Execução**: o código do contrato é executado por todos os nós da rede, validado através do algoritmo de consenso da blockchain.

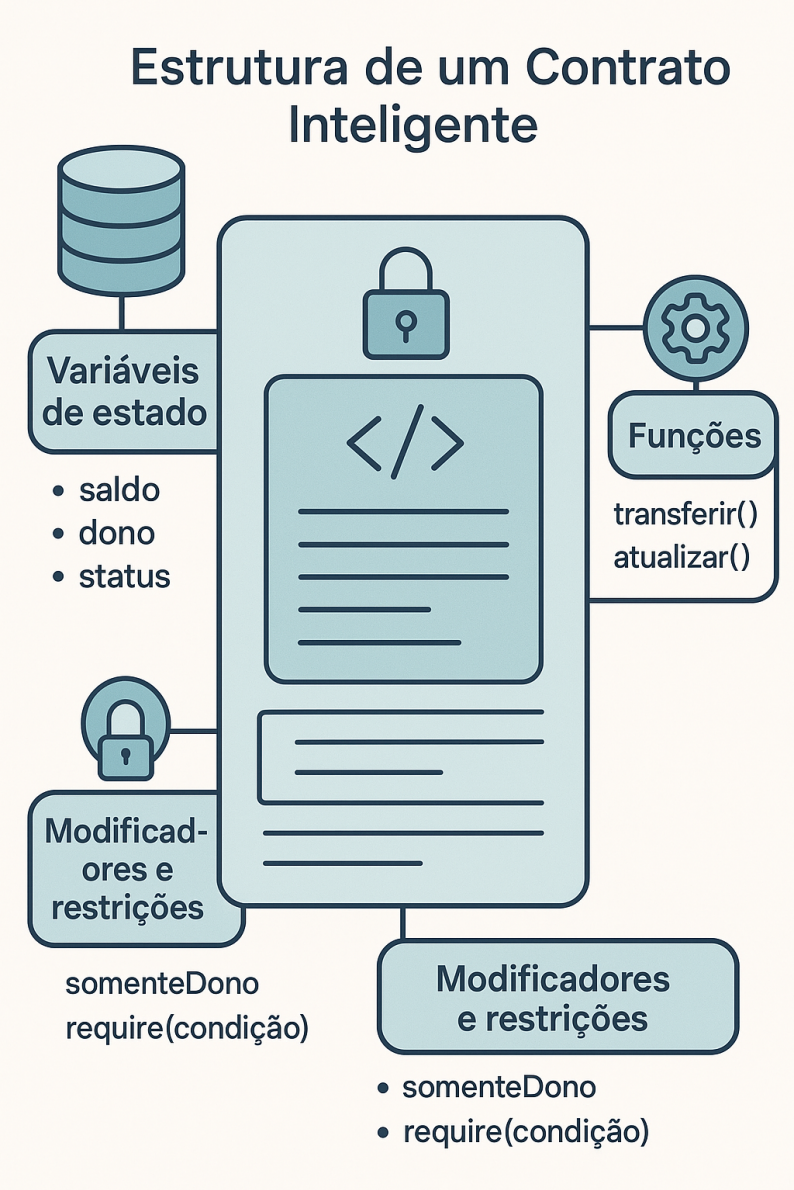

Figure 1.1. Estrutura Smart Contract

4. **Persistência**: o resultado da execução (como mudanças de estado) é armazenado de forma imutável.

Diferente de programas comuns que rodam em servidores, os contratos inteligentes são executados por todos os nós da rede blockchain, garantindo consenso e confiabilidade. Essa execução redundante assegura que o resultado será sempre o mesmo, independentemente de quem interaja com o contrato.

1.3.3. Modelo de Armazenamento e Custos

Toda operação realizada por um contrato inteligente consome recursos computacionais da rede. No Ethereum, esse custo é medido em *gas*, uma unidade que representa a quantidade

de trabalho necessário para realizar uma operação. O usuário que executa uma função do contrato deve pagar o custo correspondente em criptomoeda (por exemplo, ETH).

Além disso, os dados armazenados em contratos inteligentes ocupam espaço na blockchain, o que também gera custos de armazenamento. Por isso, desenvolvedores são incentivados a escrever códigos otimizados, que minimizem tanto o consumo de *gas* quanto o uso de armazenamento.

1.3.4. Oráculos e Interação com o Mundo Externo

Contratos inteligentes, por padrão, operam apenas com dados internos à blockchain. Para acessar informações externas como cotação de moedas, clima, ou resultados esportivos é necessário o uso de **oráculos**, que são serviços confiáveis responsáveis por trazer dados do mundo real para o ambiente blockchain. Sem os oráculos, os contratos ficariam isolados e incapazes de reagir a eventos externos.

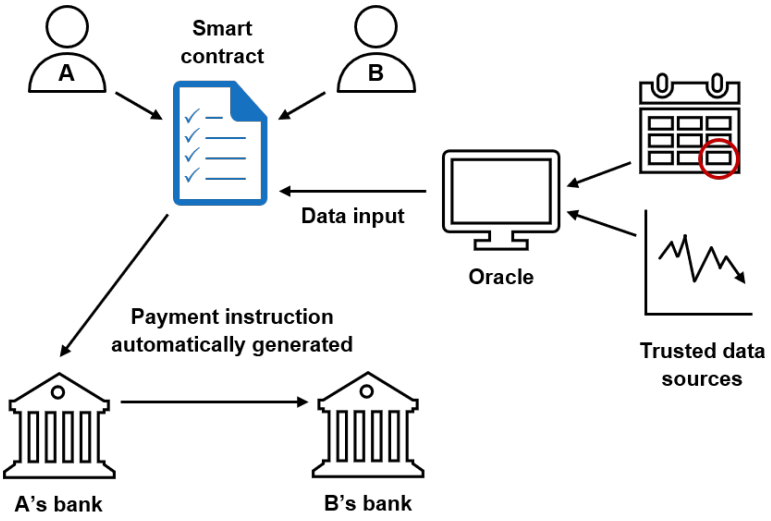

Figure 1.2. Oráculo Blockchain

O Universal Market Acess (UMA) é um importante oráculo que fornece uma infraestrutura para a criação de contratos inteligentes e aplicações financeiras descentralizadas. Ele utiliza uma combinação de dados on-chain e off-chain para alimentar contratos inteligentes com informações externas confiáveis. A sua criptomoeda se chama UMA.

1.3.5. Limitações Técnicas

Apesar de sua robustez, os contratos inteligentes ainda enfrentam algumas limitações:

- **Imutabilidade**: erros de código não podem ser corrigidos após o deploy.
- **Escalabilidade**: a execução por todos os nós pode gerar gargalos em redes congestionadas.
- **Complexidade no desenvolvimento**: exige conhecimento técnico específico e revisão rigorosa de segurança.

Apesar desses desafios, os contratos inteligentes continuam sendo uma das inovações mais promissoras da era blockchain, permitindo a construção de sistemas automatizados, confiáveis e transparentes.

1.4. Plataformas Populares para Contratos Inteligentes

A implementação de contratos inteligentes depende diretamente das plataformas de blockchain que oferecem suporte à criação, execução e validação desses contratos. Desde o surgimento do Ethereum, diversas blockchains surgiram com propostas específicas para otimizar desempenho, reduzir custos e melhorar a experiência de desenvolvedores e usuários. Nesta seção, destacam-se algumas das principais plataformas utilizadas atualmente.

1.4.1. Ethereum

O **Ethereum** é, sem dúvida, a plataforma mais consolidada e amplamente adotada para o desenvolvimento de contratos inteligentes. Lançado em 2015, o Ethereum introduziu o conceito de *Ethereum Virtual Machine* (EVM), um ambiente de execução que permite que qualquer nó da rede execute os contratos de forma padronizada.

A linguagem principal para programação no Ethereum é a **Solidity**, que permite escrever contratos com funcionalidades complexas. A plataforma também adota um sistema de taxas baseado em *gas*, o que torna sua utilização transparente e economicamente regulada.

Entre suas vantagens estão:

- Forte comunidade de desenvolvedores;
- Ferramentas maduras (Remix, Truffle, Hardhat);
- Compatibilidade com uma vasta gama de carteiras e dApps;
- Suporte a padrões como ERC-20 (tokens fungíveis) e ERC-721 (NFTs).

No entanto, o Ethereum enfrenta críticas relacionadas à escalabilidade e ao custo elevado de transações em momentos de congestionamento.

1.4.2. Binance Smart Chain (BSC)

A **Binance Smart Chain** surgiu como uma alternativa ao Ethereum, oferecendo maior velocidade e taxas mais baixas. É compatível com a EVM, o que facilita a migração de contratos existentes do Ethereum para a BSC sem necessidade de reescrita do código.

Seu foco principal está em aplicações DeFi e em permitir transações rápidas com baixa latência. Por ser uma blockchain mais centralizada, sua governança é frequentemente questionada, mas sua eficiência e integração com o ecossistema Binance são grandes atrativos.

1.4.3. Solana

A **Solana** destaca-se por sua arquitetura baseada em *Proof of History* (PoH), que permite processar milhares de transações por segundo com taxas extremamente baixas. Ao

contrário do Ethereum, a Solana não utiliza a EVM, e seus contratos são escritos majoritariamente em **Rust** ou **C**.

Essa plataforma tem sido amplamente utilizada para aplicações de alto desempenho, como marketplaces de NFTs e jogos em blockchain. Apesar disso, a complexidade do ambiente de desenvolvimento e as interrupções da rede em momentos críticos ainda são desafios a serem superados.

1.4.4. Cardano

A **Cardano** adota uma abordagem científica e baseada em revisão por pares. Seu modelo de contrato inteligente é alimentado pela linguagem funcional **Plutus**, que oferece segurança formal através de verificação matemática. Embora seja mais recente na implementação de contratos inteligentes, Cardano apresenta um ambiente promissor, com foco em sustentabilidade e escalabilidade.

Seu diferencial está no rigor acadêmico aplicado ao desenvolvimento da plataforma, o que a torna atrativa para projetos institucionais e de longo prazo.

1.4.5. Outras plataformas

Além das citadas acima, diversas outras blockchains vêm ganhando espaço, como:

- **Avalanche**: alta escalabilidade e compatibilidade com EVM;
- **Tezos**: contratos escritos em *Michelson*, com foco em governança on-chain;
- **Polygon**: solução de segunda camada para Ethereum, que reduz custos e aumenta a velocidade;
- **Near Protocol**: voltada à simplicidade para desenvolvedores e à eficiência energética.

Cada plataforma apresenta suas particularidades, vantagens e desafios, sendo a escolha ideal dependente dos requisitos do projeto, linguagem preferida, custo de transação, desempenho desejado e nível de descentralização necessário.

1.4.6. Principais Aplicações

Os contratos inteligentes têm se destacado como uma das tecnologias mais inovadoras no ecossistema blockchain, desempenhando um papel crucial na evolução de serviços digitais descentralizados. Sua capacidade de automatizar processos, garantir transparência, reforçar a segurança e eliminar intermediários vem impulsionando seu uso em uma ampla gama de setores econômicos e sociais. A seguir, detalham-se as principais aplicações dessa tecnologia, evidenciando seu impacto e potencial transformador.

- **Finanças Descentralizadas (DeFi)**: Uma das áreas que mais se beneficiou da implementação de contratos inteligentes é a das finanças descentralizadas (DeFi), que propõe um ecossistema financeiro aberto, sem a necessidade de intermediários

tradicionais como bancos ou corretoras. Plataformas DeFi utilizam contratos inteligentes para oferecer serviços como empréstimos colateralizados, onde usuários podem garantir ativos digitais como garantia e tomar empréstimos de forma automatizada; sistemas de staking, que recompensam usuários pela sua participação na segurança e manutenção da rede; yield farming, que permite a otimização de rendimentos através do fornecimento de liquidez a pools; e exchanges descentralizadas (DEX), que viabilizam a negociação peer-to-peer de criptoativos sem a necessidade de uma entidade central. Esses serviços não só ampliam a acessibilidade a produtos financeiros, especialmente em regiões sub-bancarizadas, como também promovem a autonomia dos usuários sobre seus próprios recursos. Um exemplo de DeFi é a Binance Smart chain(BSC): Embora a Binance seja amplamente conhecida como exchange centralizada, sua Binance Smart Chain (BSC) é uma blockchain pública que suporta contratos inteligentes e uma vasta gama de aplicações DeFi. Na BSC, projetos como *PancakeSwap* (DEX descentralizada) e *Venus Protocol* (empréstimos descentralizados) têm ganhado destaque. A BSC oferece taxas de transação mais baixas e alta velocidade em comparação com o Ethereum, o que atraiu um grande volume de usuários. A plataforma também suporta *yield farming*, staking e várias outras funcionalidades típicas de DeFi, consolidando-se como um ecossistema híbrido entre centralização e descentralização.

- **NFTs (Tokens Não Fungíveis)**: No campo da arte digital, entretenimento e propriedade intelectual, os NFTs (Non-Fungible Tokens) têm se consolidado como uma inovação disruptiva. Diferente dos tokens fungíveis como Bitcoin ou Ethereum, os NFTs representam ativos digitais únicos e indivisíveis, permitindo autenticar a originalidade e a propriedade de itens digitais. Contratos inteligentes são fundamentais nesse ecossistema, pois gerenciam a criação (minting), transferência e rastreamento de NFTs de forma segura e transparente. Além disso, possibilitam a implementação de regras automatizadas, como royalties programáveis, garantindo que criadores recebam uma porcentagem automaticamente a cada nova venda secundária. Essa funcionalidade inaugura um novo modelo de remuneração contínua, fortalecendo a sustentabilidade econômica para artistas e desenvolvedores de conteúdo. O *OpenSea* é atualmente a maior e mais popular plataforma de NFTs do mundo. Lançada em 2017, ela permite que qualquer usuário crie, compre, venda e explore ativos digitais tokenizados, incluindo arte digital, itens de jogos, música e colecionáveis virtuais. A plataforma opera principalmente na rede Ethereum e foi pioneira ao oferecer um marketplace abrangente para NFTs, promovendo milhões de transações e abrindo portas para artistas digitais e criadores de conteúdo em escala global.

- **Cadeia de Suprimentos (Supply Chain)**: A aplicação de contratos inteligentes na gestão da cadeia de suprimentos (supply chain) tem revolucionado a forma como produtos são monitorados ao longo de todo o seu ciclo de vida. Com a tecnologia blockchain, é possível registrar de maneira imutável e auditável cada etapa da jornada de um produto — desde a produção, passando pelo transporte, até a entrega ao consumidor final. Os contratos inteligentes automatizam processos como liberações de pagamento e verificação de conformidade em tempo real, tornando o sistema mais eficiente e confiável. Isso não apenas reduz riscos de fraudes e erros,

mas também melhora a transparência e a rastreabilidade, fatores cada vez mais valorizados por consumidores e parceiros comerciais preocupados com práticas éticas e sustentabilidade. A *VeChain* é uma das principais plataformas especializadas em rastreamento e gerenciamento de cadeias de suprimentos. Utilizando sensores IoT e blockchain, a VeChain permite registrar cada etapa da cadeia logística, desde a produção até a entrega final. Empresas de setores como alimentos, saúde e indústria automotiva já utilizam VeChain para garantir autenticidade, prevenir fraudes e melhorar a eficiência operacional. Um exemplo notável é a parceria da VeChain com a empresa de luxo *LVMH* para autenticar produtos de alta-costura.

- **Governança Descentralizada (DAOs)**: As Organizações Autônomas Descentralizadas (DAOs) representam um modelo inovador de governança corporativa e comunitária. Baseadas em contratos inteligentes, as DAOs operam sem estruturas hierárquicas tradicionais, permitindo que regras operacionais e mecanismos de tomada de decisão sejam codificados e executados automaticamente na blockchain. Membros dessas organizações podem propor mudanças, votar em decisões importantes e contribuir para a evolução da entidade de forma democrática e transparente. Esse modelo tem ganhado ampla adoção em projetos open source, plataformas DeFi e iniciativas de crowdfunding, onde a confiança é construída não através de líderes centrais, mas por meio da transparência e imutabilidade do código. As DAOs demonstram o potencial de criar sistemas de governança mais inclusivos, resilientes e responsivos às necessidades da comunidade. A plataforma *Aragon* permite a criação e gestão de Organizações Autônomas Descentralizadas (DAOs) de forma simples e segura. Fundada em 2016, Aragon fornece ferramentas para governança comunitária, votação, gestão de tesouraria e criação de regras organizacionais codificadas em contratos inteligentes. Usada amplamente em projetos de código aberto e iniciativas comunitárias, Aragon visa democratizar a governança digital, eliminando estruturas hierárquicas rígidas e promovendo maior transparência e participação dos membros.

1.4.7. Desafios e Limitações

Embora os smart contracts ofereçam benefícios significativos, sua adoção e implementação em larga escala ainda enfrentam diversos desafios e limitações. Tais obstáculos precisam ser cuidadosamente considerados por desenvolvedores, usuários e legisladores para garantir a segurança e a viabilidade dessa tecnologia. A seguir, são apresentados os principais desafios associados aos contratos inteligentes:

- **Vulnerabilidades e hacks**: Apesar da robustez criptográfica das blockchains, os smart contracts não estão imunes a falhas de programação. Um exemplo emblemático foi o *DAO Hack*, ocorrido em 2016, quando uma vulnerabilidade no código de um contrato inteligente permitiu que um atacante drenasse cerca de US$ 60 milhões em Ether da DAO (Decentralized Autonomous Organization). Esse incidente destacou a importância de auditorias rigorosas e práticas de desenvolvimento seguro, evidenciando que um erro no código pode ter consequências financeiras devastadoras.

- **Escalabilidade e custos de gas**: À medida que o uso de smart contracts cresce, redes blockchain — especialmente aquelas baseadas no modelo Proof of Work, como a Ethereum até sua transição para Proof of Stake — enfrentam problemas de congestionamento. Esse aumento de demanda resulta em elevação dos custos de transação (taxas de *gas*), tornando a execução de contratos complexos economicamente inviável para muitos usuários. A escalabilidade continua sendo um dos maiores desafios para viabilizar aplicações em massa, motivando o desenvolvimento de soluções como redes de camada 2 (*Layer 2*) e blockchains mais eficientes.

- **Aspectos regulatórios e incertezas legais**: A natureza descentralizada e global dos smart contracts gera incertezas quanto ao seu enquadramento jurídico. Questões relacionadas à validade legal dos contratos inteligentes, jurisdição em casos de disputa, e obrigações fiscais ainda estão em debate em muitos países. A ausência de regulamentações claras pode desencorajar empresas e instituições a adotarem plenamente essa tecnologia, especialmente em setores que exigem forte compliance legal, como finanças e saúde. Além disso, a falta de mecanismos para reverter contratos executados automaticamente levanta discussões sobre como proteger consumidores e partes vulneráveis.

1.5. Estudo de Caso: Sistema de Contratos Inteligentes para Aluguel

Figure 1.3. Tela principal

1.6. Contexto

Imaginemos uma plataforma onde proprietários (locadores) e inquilinos (locatários) gerenciam contratos de aluguel de forma transparente usando a blockchain Ethereum. O sistema permite:

- Registrar contratos entre locador e locatário;
- Depositar caução (depósito de segurança);
- Pagar aluguel mensal;
- Sacar caução ao fim do contrato;

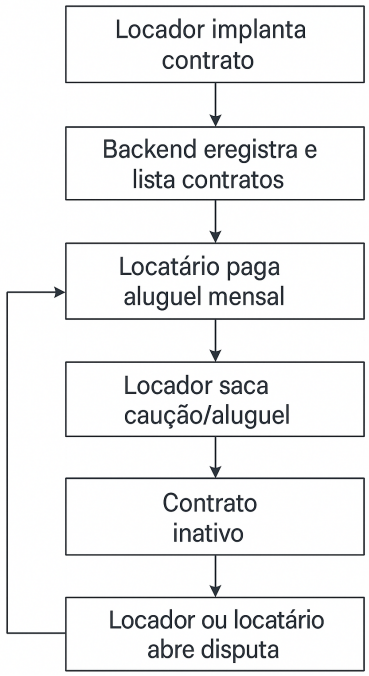

Figure 1.4. Fluxo de execução

- Abrir disputas em caso de desacordos;
- Encerrar contratos.

A vantagem do uso do contrato inteligente é garantir regras imutáveis e automação financeira segura, sem intermediários.

1.7. Parte 1 — Contrato Inteligente em Solidity

1.7.1. Objetivo

Criar um contrato inteligente que:

- Armazene as partes envolvidas (locador e locatário);
- Controle o depósito de caução e o pagamento de aluguel;
- Permita ações específicas apenas para locador ou locatário;
- Controle o status do contrato (ativo, encerrado, em disputa).

1.7.2. Funcionamento

O contrato é criado pelo locador, que passa o endereço do locatário e o valor do aluguel mensal. O funcionamento segue as seguintes etapas:

1. O locatário deposita a caução (em ether);

2. O locatário paga o aluguel mensal, enviando o valor exato;
3. O locador pode sacar a caução apenas após o encerramento do contrato;
4. O contrato pode ser encerrado pelo locador;
5. Qualquer uma das partes pode abrir uma disputa para resolução fora da blockchain.

1.8. Parte 2 — Backend Django com Web3.py

1.8.1. Objetivo

- Conectar-se a um nó Ethereum local (ex: Ganache);
- Consultar o estado do contrato (locador, locatário, status);
- Construir transações para as ações solicitadas pelo frontend;
- Executar ou simular transações.

1.8.2. Desafios Técnicos

- Assinatura e envio das transações com chaves privadas, que não devem permanecer no frontend;
- Atualizar o status do contrato exibido no frontend após cada ação;
- Segurança e validação dos dados fornecidos pelos usuários.

1.8.3. Estrutura Básica

- Uma lista (ou banco de dados) de contratos disponíveis;
- Função para renderizar a lista de contratos com seus respectivos status;
- Endpoint para receber ações do usuário, construir a transação correspondente e enviá-la para a blockchain.

1.9. Parte 3 — Frontend Django com Bootstrap

1.9.1. Objetivo

- Apresentar a lista de contratos com seus dados;
- Exibir status legível (*Ativo*, *Encerrado*, *Em Disputa*);
- Disponibilizar formulários para executar ações como:
  - Depositar caução;
  - Pagar aluguel;
  - Sacar caução;
  - Encerrar contrato;
  - Abrir disputa.
- Interface limpa e responsiva utilizando Bootstrap.

1.9.2. Fluxo de Uso

1. O usuário visualiza a lista de contratos;

2. Preenche o endereço Ethereum da carteira que será usada na transação e o valor (se aplicável);

3. Clica no botão da ação desejada;

4. O backend processa a solicitação, envia a transação e atualiza o estado exibido no frontend.

1.9.3. Fluxo Completo de uma Interação Típica

- **Criação do contrato:** o locador implanta o contrato passando o endereço do locatário e o valor do aluguel mensal;

- **Listagem no frontend:** o backend (Django) lê os dados do contrato via Web3.py e mostra o contrato com status *ativo*;

- **Locatário deposita caução:** o usuário insere seu endereço e valor, clica em *Depositar Caução*; o backend cria e envia a transação, que atualiza a variável `caucao`;

- **Locatário paga aluguel:** mensalmente, o locatário paga o aluguel por meio do botão *Pagar Aluguel*;

- **Locador encerra contrato:** ao final, o locador chama a função `encerrarContrato`, mudando o status para *encerrado*;

- **Locador saca caução:** com o contrato encerrado, o locador pode sacar a caução depositada;

- **Disputa:** em caso de desacordo, qualquer parte pode abrir uma disputa para resolução fora da blockchain.

1.10. Benefícios Práticos do Sistema

- **Transparência:** todos os pagamentos e ações são registrados na blockchain pública (ou testnet), proporcionando rastreabilidade;

- **Segurança:** regras imutáveis garantem que nenhuma das partes aja fora dos termos previamente acordados;

- **Automação:** transferências financeiras e controle de status são automatizados e não dependem de intermediários;

- **Facilidade:** com um frontend amigável, os usuários não precisam interagir diretamente com a blockchain.

1.10.1. Conclusão

O **Sistema de Contratos Inteligentes para Aluguel** ilustra de maneira prática como contratos inteligentes podem ser integrados a sistemas web modernos, promovendo segurança e descentralização em aplicações financeiras. Esse estudo de caso serve como uma excelente introdução aos conceitos de blockchain, Ethereum, smart contracts e integração com carteiras digitais como a MetaMask.

Considerações Finais

O minicurso **"Introdução aos Smart Contracts"** teve como proposta principal aproximar os participantes do universo da tecnologia blockchain por meio de experiências práticas e contextualizadas. Ao longo do curso, foram apresentados os fundamentos técnicos dos contratos inteligentes, suas aplicações reais e os principais desafios associados ao seu desenvolvimento.

Por meio do estudo de caso do **Sistema de Contratos Inteligentes para Aluguel**, os participantes puderam vivenciar todas as etapas de criação de um contrato inteligente: desde a definição de regras de negócio, a implementação em *Solidity*, até a construção de um sistema web funcional com *Django* e *Web3.py*. O uso do *Ganache* como ambiente local de testes e da carteira *MetaMask* para autenticação reforçou a integração entre o front-end e a blockchain, proporcionando um cenário completo e realista.

Espera-se que os conhecimentos adquiridos durante o minicurso sirvam como base sólida para que os participantes avancem em seus estudos e desenvolvam soluções descentralizadas mais robustas, seguras e inovadoras. A descentralização representa uma mudança de paradigma tecnológica, e a compreensão prática dos contratos inteligentes é um passo essencial para qualquer profissional que deseja atuar nessa nova fronteira da computação.

References

[1] H. Taherdoost, "Smart contracts in blockchain technology: A critical review," *Information*, vol. 14, no. 2, p. 117, 2023.

[2] S. Nakamoto, "Bitcoin whitepaper," 2008. [Online]. Available: *https://bitcoin.org/bitcoin.pdf*. Accessed: Jul. 17, 2019.

[3] H. Watanabe, S. Fujimura, A. Nakadaira, Y. Miyazaki, A. Akutsu, and J. Kishigami, "Blockchain contract: Securing a blockchain applied to smart contracts," in *Proc. 2016 IEEE Int. Conf. on Consumer Electronics (ICCE)*, pp. 467–468, 2016.

[4] A. M. Antonopoulos and G. Wood, *Mastering Ethereum: Building Smart Contracts and DApps*. O'Reilly Media, 2018.

[5] W. Metcalfe *et al.*, "Ethereum, smart contracts, DApps," *Blockchain and Crypt Currency*, vol. 77, pp. 77–93, 2020. Springer Singapore.

[6] N. Szabo, "Smart contracts: Building blocks for digital markets," 1996.

[7] Buterin, V. (2013). *Ethereum Whitepaper: A Next-Generation Smart Contract and Decentralized Application Platform*.

[8] Wood, G. (2014). *Ethereum: A Secure Decentralized Generalized Transaction Ledger*.

[9] Carvalho, C. A., & Ávila, L. V. (2020). *A Tecnologia Blockchain Aplicada aos Contratos Inteligentes*.

[10] Antonopoulos, A. M., & Wood, G. (2018). *Mastering Ethereum: Building Smart Contracts and DApps*. O'Reilly Media.

[11] Binance Smart Chain. *https://www.bnbchain.org*, acesso em: Maio de 2025.

[12] OpenSea. *https://opensea.io*, acesso em: Maio de 2025.

[13] VeChain. *https://www.vechain.org*, acesso em: Maio de 2025.

[14] Aragon. *https://aragon.org*, acesso em: Maio de 2025.

[15] Universal Market Access. *https://uma.xyz/*, acesso em: Maio de 2025.

[16] Blockchain Ganache. *https://www.blockchain-council.org/blockchain/ganache-blockchain-all-you-need-to-know/*, acesso em: Maio de 2025.

[17] Metamask. *https://criptonizando.com/aprenda/metamask-o-que-e/*, acesso em: Maio de 2025.

[18] Solidity. *https://docs.soliditylang.org/en/v0.8.30/*, acesso em: Maio de 2025.

[19] Web3 python. *https://web3py.readthedocs.io/en/stable/*, acesso em: Maio de 2025.

[20] Django frammework. *https://docs.djangoproject.com/pt-br/5.2/*, acesso: em Maio de 2025