

Scrypto

A Nova Fronteira da Programação para DeFi

Teo Pires Marques

2024

Teo Pires Marques

SCRYPTO

A Nova Fronteira da Programação para DeFi

2024 version 0.1

Free culture licence: Because loving is sharing!



*Creative Commons Attribution 4.0
International License*

E-mail: teopmarques@hotmail.com

Linkedin: linkedin.com/in/teopires/

The book was created using free and open-source software:

- *LibreOffice (AltSearch, Sk-spell) – typesetting*
- *PT Serif – font family*

The content has not been subjected to linguistic revision.

2024

ISBN xxx-x-xx-xxxxxx-0

*Cumpra
seu dever prescrito,
pois a ação é melhor
do que a inação.*

Sumário

Introdução ao Scrypto	1
O que é Scrypto?	2
Contexto e Origem	3
Importância do Scrypto no DeFi	4
Fundamentos do Scrypto	5
Arquitetura e Linguagem	6
Componentes Básicos	7
Vantagens do Scrypto	9
Como Desenvolver com Scrypto	11
Ambiente de Desenvolvimento	12
Primeiro Projeto em Scrypto	14
Testes e Debugging	16
Casos de Uso de Scrypto	xviii
DeFi e Aplicações Financeiras	19
Tokens e NFTs	20
Integrações e Expansões	21
Vantagens em Diferentes Setores	22
O Futuro do Scrypto	23
Evolução e Atualizações	24
Adaptação ao Crescimento do DeFi	25
Expansão da Comunidade	26
Conclusão	27
Referências	32

Introdução ao Scrypto



O que é Scrypto?

Imagine que você quer criar um aplicativo, mas em vez de ser um aplicativo comum como os que temos no celular, ele funciona em um sistema descentralizado, onde não há uma empresa controlando tudo. Esse tipo de aplicativo é chamado de **dApp** (aplicativo descentralizado). O **Scrypto** é uma linguagem de programação criada para facilitar o desenvolvimento desses dApps, especificamente dentro de um ambiente chamado **Radix**.

Para quem não está familiarizado, pense no Radix como uma plataforma onde esses dApps podem ser desenvolvidos e rodados com segurança e eficiência. O Scrypto, então, é a "ferramenta" que os programadores usam para criar esses aplicativos. Ele é como um conjunto de instruções que os programadores seguem para construir esses sistemas descentralizados que podem gerenciar dinheiro, contratos, e até novas formas de interação online, sem precisar de um banco ou outra instituição no meio.

Contexto e Origem

O Scrypto foi criado pela equipe por trás do Radix, uma plataforma que nasceu para melhorar o mundo das finanças descentralizadas, ou **DeFi**. DeFi é uma forma de fazer operações financeiras como empréstimos, investimentos, ou pagamentos, sem precisar de um banco tradicional. Imagine poder emprestar dinheiro diretamente para alguém do outro lado do mundo, ou investir em um fundo de forma instantânea e sem taxas abusivas – isso é possível com DeFi.

A equipe da Radix percebeu que, para que o DeFi realmente pudesse crescer e se tornar acessível para mais pessoas, era necessário criar uma linguagem de programação que fosse segura, flexível e, ao mesmo tempo, fácil de usar pelos desenvolvedores. Foi assim que o Scrypto surgiu: uma linguagem projetada desde o início com o foco em segurança, para evitar erros comuns que podem ocorrer em outras linguagens, e com ferramentas que facilitam a criação desses dApps.

Importância do Scrypto no DeFi

No mundo do DeFi, a segurança é fundamental. Se você está lidando com dinheiro e contratos digitais, não pode haver espaço para erros. Um pequeno bug pode causar perdas milionárias. O Scrypto foi desenvolvido exatamente para minimizar esses riscos, oferecendo uma maneira de escrever programas que são mais seguros e confiáveis.

Além disso, o Scrypto é flexível. Isso significa que ele permite que os desenvolvedores criem soluções que podem ser adaptadas a diferentes necessidades e cenários dentro do DeFi. Seja para criar uma nova moeda digital, uma plataforma de empréstimos, ou uma maneira inovadora de trocar ativos, o Scrypto dá as ferramentas necessárias para que isso aconteça de forma eficiente.

Em resumo, o Scrypto é mais do que apenas uma linguagem de programação. Ele é um passo importante para tornar o mundo das finanças descentralizadas mais acessível, seguro e poderoso, permitindo que novos tipos de aplicativos financeiros surjam e se desenvolvam.

Fundamentos do Scripto



Arquitetura e Linguagem

Para entender o Scrypto, é útil saber que ele é construído sobre uma base muito sólida: uma linguagem de programação chamada **Rust**. Rust é conhecida entre os programadores por ser extremamente segura e eficiente. Em termos simples, ela ajuda a evitar muitos dos problemas que podem surgir ao programar, como erros que tornam os aplicativos inseguros ou lentos. Rust faz isso de uma forma que permite que os aplicativos funcionem bem, sem desperdício de recursos, o que é essencial para aplicativos descentralizados no mundo das finanças.

O Scrypto, então, pega essa segurança e eficiência de Rust e a adapta para o ambiente específico do Radix. Isso significa que quando você escreve um programa em Scrypto, você está criando algo que não apenas funciona bem, mas também é protegido contra muitos dos problemas que poderiam ocorrer em outras linguagens de programação.

Componentes Básicos

Agora, vamos falar sobre como o Scrypto organiza as coisas internamente. A linguagem é composta de alguns blocos básicos que ajudam os desenvolvedores a construir seus aplicativos descentralizados. Esses blocos são chamados de **blueprints**, **components**, e **resources**.

- **Blueprints:** Pense nos blueprints como plantas baixas ou modelos. Quando você quer construir uma casa, você começa com um projeto, certo? Um blueprint é exatamente isso: um modelo que define a estrutura e o comportamento do que você quer criar. No contexto do Scrypto, um blueprint descreve como algo deve funcionar dentro do seu dApp, como uma instrução passo a passo para a construção.
- **Components:** Se o blueprint é o projeto, o **component** é a casa construída a partir desse projeto. Em outras palavras, um component é uma instância do blueprint. Quando você pega um blueprint e o coloca em ação, o que você obtém é um component. Esse component pode então ser

implantado na rede Radix, onde começa a funcionar como parte do seu aplicativo descentralizado.

- **Resources:** No mundo de Scrypto, **resources** são como os ativos digitais que você pode criar e gerenciar. Eles podem ser tokens (como o próprio Bitcoin) ou qualquer outro tipo de ativo digital que seu dApp precise gerenciar. Os resources são manipulados dentro do ecossistema Radix, e o Scrypto facilita a criação e gestão desses ativos de forma segura e eficiente.

Vantagens do Scrypto

Então, por que usar o Scrypto? A maior vantagem é a segurança. Como já mencionamos, o Scrypto é baseado em Rust, que já é uma linguagem projetada para evitar muitos erros comuns. Isso significa que quando você programa em Scrypto, tem menos chances de cometer aqueles erros que podem causar problemas graves, como falhas de segurança.

Outra vantagem importante é a **integração** com o ecossistema Radix. O Scrypto foi projetado para funcionar perfeitamente dentro da plataforma Radix, o que torna muito mais fácil para os desenvolvedores criar, testar e implantar seus dApps. Você não precisa se preocupar com compatibilidade ou com a complexidade de integrar diferentes sistemas – tudo é feito para funcionar de forma fluida e eficiente.

Além disso, o Scrypto oferece **flexibilidade**. Quer você esteja criando um simples token digital ou um complexo sistema financeiro, o Scrypto tem as ferramentas necessárias para construir o que você precisa. Isso significa que você pode começar com algo

simples e ir expandindo conforme suas necessidades crescem, tudo dentro do mesmo ambiente seguro e eficiente.

Em resumo, o Scrypto não é apenas uma linguagem de programação; é um conjunto completo de ferramentas projetado para ajudar a criar aplicativos descentralizados seguros, eficientes e adaptáveis no emocionante e dinâmico mundo das finanças descentralizadas.

Como Desenvolver com Scrypto



Ambiente de Desenvolvimento

Antes de começar a programar em Scrypto, você precisa configurar o seu ambiente de desenvolvimento. Isso é como preparar sua "mesa de trabalho" onde você vai construir seus aplicativos descentralizados (dApps). Felizmente, o processo é direto e, com as ferramentas certas, você estará pronto para começar em pouco tempo. A primeira coisa que você precisa é do **Radix Engine**, que é o motor por trás de tudo o que você vai criar com Scrypto. O Radix Engine é como o "cérebro" que executa seus contratos inteligentes e gerencia os dApps na rede Radix. Para começar, você vai precisar instalar o SDK (Software Development Kit) do Radix, que inclui todas as ferramentas necessárias para desenvolver, testar e implantar seus projetos.

Além do Radix Engine, é recomendável ter um ambiente de desenvolvimento integrado (IDE), como o **Visual Studio Code** ou qualquer outro de sua preferência. O IDE é o programa onde você realmente escreve e edita o código do seu dApp. Ele oferece várias funcionalidades que facilitam a escrita de código, como auto-completar, realce de sintaxe, e plugins específicos

para Scrypto.

Depois de configurar o SDK do Radix e o seu IDE, você estará pronto para começar a programar com Scrypto. É como ter todas as suas ferramentas à mão, prontas para você começar a construir.

Primeiro Projeto em Scrypto

Vamos começar com um exemplo simples para você ver como funciona o desenvolvimento em Scrypto. Imagine que você quer criar um dApp básico que permite a criação de um token simples – algo como o seu próprio Bitcoin.

1. **Criando o Projeto:** Primeiro, você vai criar um novo projeto Scrypto no seu ambiente de desenvolvimento. Isso é feito com um simples comando no terminal, que gera toda a estrutura de pastas e arquivos que você vai precisar.
2. **Escrevendo o Blueprint:** Lembra dos blueprints que mencionamos? O próximo passo é escrever o blueprint para o seu token. No blueprint, você define as regras de como esse token vai funcionar – como ele é criado, como pode ser transferido, e assim por diante.
3. **Instanciando o Component:** Uma vez que o blueprint está pronto, você pode criar uma instância dele – ou seja, criar o component. Esse

component é o que realmente vai rodar na rede Radix, seguindo as regras que você definiu no blueprint.

4. **Implantando na Rede:** Finalmente, você implanta o component na rede Radix. Isso significa que ele agora está ativo e pode ser usado por outras pessoas que interajam com o seu dApp.

Essa é uma visão simplificada, mas o importante é entender que Scrypto facilita muito o processo de ir da ideia até a execução, com uma estrutura clara e ferramentas que guiam o desenvolvedor em cada passo.

Testes e Debugging

Desenvolver um dApp não é só escrever código – você também precisa garantir que ele funciona corretamente. Isso é especialmente importante em DeFi, onde pequenos erros podem ter grandes consequências. Felizmente, o Radix Engine vem com ferramentas integradas para ajudar você a testar e depurar (debug) seus contratos inteligentes.

- **Testes Unitários:** Esses são pequenos testes que verificam se partes individuais do seu código funcionam como esperado. Por exemplo, você pode testar se a criação de um token funciona corretamente antes de seguir para o próximo passo. O Scrypto permite que você escreva esses testes junto com o seu código, facilitando a verificação de cada parte separadamente.
- **Testes de Integração:** Além dos testes unitários, você vai querer verificar se todas as partes do seu dApp funcionam bem juntas. Isso é feito com os testes de integração, que simulam o funcionamento do dApp como um todo. Eles são

cruciais para garantir que, mesmo quando várias partes do seu código interagem, tudo funciona sem problemas.

- **Debugging:** Às vezes, mesmo com testes, algo pode dar errado. O processo de encontrar e corrigir esses problemas é chamado de debugging. O Radix Engine facilita esse trabalho, oferecendo mensagens de erro detalhadas e ferramentas que ajudam a identificar onde o problema está ocorrendo. Com essas ferramentas, você pode corrigir bugs rapidamente e garantir que seu dApp funcione perfeitamente.

Com o ambiente de desenvolvimento configurado, um exemplo de projeto em mãos, e as ferramentas de teste e debugging à disposição, você está pronto para criar, testar e lançar seus próprios dApps com Scrypto. O processo é fluido e guiado, permitindo que você se concentre em criar soluções inovadoras sem se preocupar com os detalhes técnicos que o Scrypto e o Radix Engine já cuidam para você.

Casos de Uso de Scrypto



DeFi e Aplicações Financeiras

O Scrypto é uma ferramenta poderosa, para criar soluções no campo das **finanças descentralizadas** (DeFi). DeFi é um movimento que busca criar alternativas aos serviços financeiros tradicionais, como bancos e corretoras, utilizando tecnologias descentralizadas. Desenvolvedores podem construir uma variedade de aplicações financeiras transparentes e seguras, sem necessidade de intermediários. Scrypto pode ser usado para criar um **mercado de empréstimos** descentralizado. Nesse tipo de aplicação, os usuários podem emprestar e tomar emprestado ativos digitais sem precisar de um banco. Tudo é gerido por contratos inteligentes que rodam na plataforma Radix, garantindo que as regras do empréstimo sejam seguidas automaticamente. O Scrypto facilita a criação desses contratos inteligentes, fornecendo as ferramentas necessárias para definir taxas de juros, prazos de pagamento, e outras condições de empréstimo.

Outro exemplo é a construção de uma **exchange descentralizada (DEX)**. As DEXs permitem que as pessoas troquem criptomoedas diretamente umas com as outras, sem precisar confiar em uma plataforma centralizada. Com Scrypto, você pode criar uma DEX que permite aos usuários trocar ativos de forma segura e eficiente, utilizando os componentes e resources que a linguagem oferece.

Tokens e NFTs

Além de aplicações financeiras tradicionais, o Scrypto também é ideal para criar e gerenciar **tokens** e **NFTs** (tokens não fungíveis).

- **Tokens Fungíveis:** Com Scrypto, você pode criar seu próprio token fungível dentro do ecossistema Radix. Esses tokens podem ser usados para representar moedas digitais, pontos de fidelidade, ou até mesmo ações de uma empresa.
- **NFTs:** Diferente dos tokens fungíveis, os NFTs são únicos. Cada NFT pode representar algo exclusivo, como uma obra de arte digital, um item colecionável em um jogo, ou um título de propriedade. Com Scrypto, você pode definir as regras e propriedades desses NFTs, gerenciando-os de maneira eficiente e segura dentro do seu dApp.

Esses recursos tornam o Scrypto uma excelente escolha para desenvolvedores que querem criar novas formas de valor digital, desde moedas digitais até arte digital e muito mais.

Integrações e Expansões

Um dos grandes benefícios de usar Scrypto é sua capacidade de **integração** com outras plataformas e serviços. O Scrypto foi projetado para ser flexível, o que significa que você pode conectar seu dApp a outros sistemas e serviços, expandindo suas funcionalidades.

Por exemplo, imagine que você quer integrar seu dApp com um serviço de oráculos, que fornece dados do mundo real, como preços de mercado ou resultados de eventos esportivos, para contratos inteligentes. Com Scrypto, você pode facilmente adicionar essa funcionalidade, permitindo que seu dApp reaja a informações externas em tempo real.

Outra possibilidade é integrar seu dApp com outras redes blockchain. O ecossistema Radix, onde o Scrypto opera, é projetado para interagir com outras blockchains, o que significa que você pode criar aplicações que utilizem recursos de diferentes redes. Isso pode incluir transferências de ativos entre blockchains ou o uso de dados e contratos de outras plataformas.

Vantagens em Diferentes Setores

Embora o foco inicial do Scrypto seja o DeFi, suas aplicações vão muito além. Desenvolvedores de diversas indústrias podem usar Scrypto para criar soluções personalizadas:

- **Seguros:** Contratos inteligentes que automaticamente processam e pagam sinistros baseados em condições predefinidas.
- **Imobiliário:** Tokens que representam propriedade parcial de imóveis, permitindo a negociação e transferência de ativos imobiliários sem a necessidade de intermediários.
- **Jogos:** NFTs que representam itens ou personagens únicos em jogos, proporcionando novas maneiras de os jogadores interagirem e trocarem esses itens.

Cada um desses casos de uso mostra a versatilidade do Scrypto. Ele não é apenas uma linguagem de programação – é uma plataforma para inovação. Com Scrypto, as possibilidades são praticamente ilimitadas.

O Futuro do Scripto



Evolução e Atualizações

Uma das direções futuras mais promissoras para o Scrypto é a expansão de suas **bibliotecas e ferramentas**. Com o tempo, espera-se que mais bibliotecas específicas para diferentes indústrias sejam desenvolvidas, tornando ainda mais fácil para os desenvolvedores criarem soluções personalizadas. Além disso, ferramentas de desenvolvimento, como depuradores e simuladores, serão aprimoradas, tornando o processo de criação de dApps ainda mais acessível e eficiente.

Outra área de evolução é a **integração multi-chain**. Embora o Scrypto já permita a interação com outras blockchains, o futuro pode ver essas capacidades se expandirem ainda mais. Imagine um cenário em que um único dApp, escrito em Scrypto, possa operar simultaneamente em várias blockchains, aproveitando o melhor de cada uma. Isso abriria novas possibilidades para desenvolvedores e usuários, conectando diferentes ecossistemas de maneira fluida.

Adaptação ao Crescimento do DeFi

O mercado de finanças descentralizadas (DeFi) está em rápido crescimento, e o Scrypto está posicionado para acompanhar essa expansão. À medida que novas necessidades e oportunidades surgem no DeFi, o Scrypto será adaptado para suportar essas mudanças. Isso inclui suporte a novas formas de ativos digitais, modelos de governança descentralizada, e mecanismos de consenso.

A introdução de **novas funcionalidades de segurança** será crucial para acompanhar as demandas do DeFi. Com o aumento das transações e do valor total bloqueado (TVL) em contratos inteligentes, a segurança continuará a ser uma prioridade máxima. O Scrypto já se destaca por suas características de segurança, mas o futuro promete ainda mais avanços, incluindo a implementação de métodos de verificação formal para contratos inteligentes.

Expansão da Comunidade

O futuro do Scrypto também está ligado ao crescimento de sua comunidade de desenvolvedores. Conforme mais pessoas adotam o Scrypto, a comunidade desempenha um papel fundamental na troca de conhecimento, criação de tutoriais, e desenvolvimento de soluções open-source que beneficiam a todos. A Radix está incentivando esse crescimento através de **programas de incentivo**, hackathons, e eventos que aproximam desenvolvedores de todo o mundo.

Uma comunidade ativa e engajada não só acelera o desenvolvimento de novos projetos, mas também fortalece a confiança no ecossistema Scrypto. Conforme mais desenvolvedores compartilham suas experiências e soluções, a linguagem se torna mais robusta e capaz de suportar uma vasta gama de aplicações.

Conclusão



Neste eBook, exploramos os aspectos essenciais do Scrypto, uma linguagem de programação poderosa projetada especificamente para o desenvolvimento de aplicativos descentralizados (dApps) no ecossistema Radix. Iniciamos com uma introdução ao que é o Scrypto, entendendo seu contexto e origem dentro do setor de finanças descentralizadas (DeFi). Discutimos a importância do Scrypto no DeFi, mostrando como ele facilita a criação de dApps mais seguros e eficientes.

Passamos pelos fundamentos da linguagem, explicando sua arquitetura baseada em Rust e os principais componentes, como blueprints, components, e resources. Demonstramos como configurar o ambiente de desenvolvimento, construir um projeto simples e realizar testes e debugging, fornecendo uma base sólida para qualquer desenvolvedor que deseje iniciar com Scrypto.

Exploramos também diversos casos de uso, desde a criação de mercados financeiros descentralizados até a implementação de tokens fungíveis e NFTs, destacando a flexibilidade e a versatilidade da linguagem. Finalizamos com uma visão sobre o futuro do Scrypto, abordando sua

evolução contínua, a expansão da comunidade de desenvolvedores, e as inúmeras oportunidades que se abrem para aqueles que desejam inovar no campo do DeFi.

Próximos Passos

Se você está interessado em se aprofundar mais no Scrypto e na programação para DeFi, aqui estão algumas sugestões de recursos adicionais:

- **Documentação Oficial do Scrypto:** A documentação oficial é um ponto de partida fundamental para aprender mais sobre as funcionalidades avançadas do Scrypto e entender os conceitos mais complexos. Ela oferece guias, exemplos de código e explicações detalhadas sobre a linguagem.
- **Tutoriais e Cursos Online:** Existem tutoriais e cursos dedicados ao Scrypto que oferecem uma abordagem prática para o aprendizado. Plataformas como YouTube, Udemy e Coursera podem ter cursos que ajudem a reforçar seus

conhecimentos.

- **Fóruns e Comunidades de Desenvolvedores:**
Participar de fóruns como o Stack Overflow e grupos de discussão em plataformas como o Reddit pode ser uma excelente maneira de interagir com outros desenvolvedores, tirar dúvidas e aprender com as experiências de outros na comunidade.
- **Hackathons e Competências de Programação:**
Participar de hackathons específicos para o ecossistema Radix ou competições de programação pode acelerar seu aprendizado e proporcionar uma experiência prática no desenvolvimento de dApps.

Chamada para Ação

Agora que você tem uma compreensão sólida dos fundamentos do Scrypto, é hora de colocar esse conhecimento em prática. Eu convido você a se engajar na **comunidade de desenvolvedores Scrypto**, onde você pode trocar ideias, colaborar em projetos e ajudar a

construir o futuro do DeFi.

Experimente desenvolver seu próprio dApp utilizando Scrypto, mesmo que seja um projeto simples. A melhor maneira de aprender é fazendo, e com o Scrypto, você tem todas as ferramentas à sua disposição para criar algo incrível. Compartilhe suas criações, colabore com outros desenvolvedores e contribua para o crescimento deste vibrante ecossistema.

O futuro da programação descentralizada está apenas começando, e você pode ser uma parte fundamental dessa revolução. Boa sorte, e boas criações!

Referências

RADIX DLT. *Scrypto Documentation*. Disponível em: <https://docs.radixdlt.com/main/scrypto/introduction.html>. Acesso em: 19 ago. 2024.

RADIX DLT. *Radix Engine Overview*. Disponível em: <https://docs.radixdlt.com/main/radix-engine/introduction.html>. Acesso em: 19 ago. 2024.

KLABNIK, Steve; NICHOLS, Carol. *The Rust Programming Language*. Disponível em: <https://doc.rust-lang.org/book/>. Acesso em: 19 ago. 2024.

DEFI PULSE. *DeFi Pulse*. Disponível em: <https://defipulse.com/>. Acesso em: 19 ago. 2024.

RADIX DLT. *Exploring Scrypto*. Disponível em: <https://www.radixdlt.com/blog>. Acesso em: 19 ago. 2024.

RADIX DLT. *Radix Discord Community*. Disponível em: <https://discord.com/invite/radix>. Acesso em: 19 ago. 2024.

RADIX DLT. *Radix DLT Hackathons*. Disponível em: <https://www.radixdlt.com/hackathon>. Acesso em: 19 ago. 2024.