



INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA (ISEL)

DEPARTAMENTO DE ENGENHARIA ELETRÓNICA E DE
TELECOMUNICAÇÕES E COMPUTADORES (DEETC)

LEIM

LICENCIATURA EM ENGENHARIA INFORMÁTICA E MULTIMÉDIA

UNIDADE CURRICULAR DE PROJETO

BookletNFT



Miguel Silvestre (45101)

Pedro Henriques (45415)

Orientador(es)

Professor Hélder Bastos

Professor Doutor Paulo Trigo

Setembro, 2022

Resumo

BookletNFT é uma aplicação web que tenta replicar as experiências de infância de colecionar, trocar e completar cadernetas de cromos.

Neste relatório descrevemos o processo de criação de uma aplicação descentralizada que permite simular cromos num ambiente digital. É possível realizar ações tais como criar, vender, comprar e abrir packs de cromos.

Esta aplicação foi realizada recorrendo a tecnologias emergentes tais como Blockchains, Non Fungible Tokens, aplicações descentralizadas. Abordamos cada um destes temas, introduzindo-os conforme necessário, e explicando as suas vantagens e desvantagens.

Este projeto foi realizado em tecnologias ainda pouco desenvolvidas, pelo que o seu desenvolvimento não foi sempre linear e por vezes um pouco caótico, sendo necessário recomeçar por diversas vezes devido a más configurações ou por erros das ferramentas utilizadas.

O sistema desenvolvido aproveita todas as vantagens da tecnologia blockchain nomeadamente no que respeita a questões de autenticação e segurança, mas também as sua desvantagens como falta de consolidação de tecnologia, levando a falhas operacionais. Apesar disto, conseguimos cumprir com a maioria dos objetivos propostos e explorar as tecnologias que pretendíamos, criando uma aplicação web capaz de servir de prova de conceito que poderá ser expandido no futuro.

Abstract

BookletNFT is a project that intends to develop a web app capable of replicating the childhood experiences of collecting, trading and finishing a trading card book.

In this report we describe the process of creation of a decentralized app that allows the replication of trading cards in a digital environment. We allow the user to perform actions such as creating, selling, buying and opening packs of cards.

To develop the application we used emerging technologies such as Blockchains, Non Fungible Tokens, DAPPs. We will approach each of these topics as needed, and explain the advantages and disadvantages, justifying the usage and presenting alternatives.

This project was made with unproven technologies, meaning the development process was not always linear and straightforward, which lead to its restart several times due to poor configs or mismatching tools.

The developed system makes use of all the advantages regarding authentication and security provided by the blockchain technology. However, some functional flaws still occur due to the operational status of the blockchain itself. Despite all those set backs we were able to deliver the proposed goals but most importantly to explore several technologies we wished to explore. We were able to create a web app capable of working as concept-proof that might be expanded in the future.

Agradecimentos

Agradecemos a todos os docentes, colegas e todos os contribuidores dos projetos Solana e Metaplex que auxiliaram, testaram e de qualquer forma contribuiram para o desenvolvimento deste projeto.

Além disso estendemos um especial agradecimento ao Professor Hélder Bastos pela sua orientação ao longo do semestre, e todo o conhecimento partilhado a nível de arquitetura de sistemas.

Gostaríamos também de agradecer especialmente às comunidades *Open-Source* do projeto Metaplex e Solana cujas ferramentas se mostraram indispensáveis à realização deste projeto.

Dedicamos este projeto a todos os colegas e docentes do ISEL, e aos developers da comunidade Solana.

Que através do seu trabalho tornaram este projeto possível.

Índice

Resumo	i
Abstract	iii
Agradecimentos	v
Índice	ix
Lista de Tabelas	xi
Lista de Figuras	xiii
1 Introdução	1
1.1 Motivação	2
1.2 Estrutura do documento	2
2 Trabalho Relacionado	3
2.1 Blockchain, NFTs e derivados	3
2.1.1 Blockchain	3
2.1.2 NFTs	6
2.1.3 Ethereum	6
2.1.4 Ripple	7
2.1.5 Terra	7
2.1.6 Solana	8
2.2 Produtos semelhantes	10
2.2.1 Cromos Panini	10
2.2.2 NBA Top Shot	11
2.2.3 Crypto Kitties	12

2.2.4	OpenSea.io	13
3	Modelo Proposto	15
3.1	Análise	15
3.1.1	Objetivos	15
3.1.2	Público Alvo	16
3.1.3	Metas a alcançar	16
3.1.4	Requisitos do sistema	17
3.1.5	Atributos do sistema	18
3.1.6	Casos de utilização	18
3.2	Protótipos Exploratórios	19
3.2.1	Ethereum	20
3.2.2	Ripple	21
3.2.3	Solana CLI, SPL-Token e Metaplex SDK	22
3.3	Opções tomadas	25
3.3.1	Solana & Metaplex Candy Machine	25
3.4	Abordagem	26
3.4.1	Processo de desenvolvimento	26
3.4.2	Arquitetura do sistema	26
3.4.3	Distribuição	27
4	Implementação do Modelo	31
4.1	Obtenção de ferramentas e configurações iniciais necessárias . .	31
4.2	Desenvolvimento inicial	33
4.2.1	Funcionalidades genéricas	33
4.2.2	Expansão de funcionalidades	36
5	Validação e Testes	39
5.1	Mint de NFT	39
5.2	Venda de um NFT	42
5.3	Criação e abertura de packs	43
6	Conclusões e Trabalho Futuro	47
	Bibliografia	49

Lista de Tabelas

2.1	Comparação de algumas blockchains	5
3.1	Requisitos do sistema	17
3.2	Atributos do sistema	18
4.1	Tabela descritiva dos principais atributos de um NFT	34

Listas de Figuras

2.1	Funcionamento da Blockchain Bitcoin [Nakamoto, 2009]	4
2.2	Everydays: the First 5000 Days by Beeple	6
2.3	Cromos da Panini by Inês Gomes Lourenço	10
2.4	NBA Top Shot Promo art	11
2.5	Crypto Kitties Promo art	12
2.6	OpenSea Promo art	13
3.1	Casos de utilização do ator utilizador	19
3.2	Casos de utilização para atores com permissões específicas	19
3.3	Diagrama <i>Swim Lane</i> do funcionamento do smart contract	24
3.4	Modelo representativo da estrutura do repositório	26
3.5	Modelo de pacotes proposto para a realização deste projeto.	27
3.7	Componentes do docker segundo [Berlatto, 2020]	28
3.6	Docker	28
4.1	Modelo representativo da interação dos componentes	32
4.2	Diagrama <i>Swim Lane</i> do funcionamento de compra de um NFT	35
4.3	Diagrama <i>Swim Lane</i> do processo de venda de um NFT	36
5.1	Selecionar o tipo de asset que vai ser o NFT	39
5.2	Fazer upload do ficheiro	40
5.3	Definir atributos do NFT	40
5.4	Quais os <i>royalties</i> a serem pagos	40
5.5	Confirmação e pagamento do lançamento	40
5.6	Página de sucesso	41
5.7	NFT na carteira	41
5.8	Tipo de venda	42

5.9	Selecionar o NFT que se pretende vender e o seu em preço em solanas	42
5.10	Review final da venda	42
5.11	Nft a ser listado para venda	43
5.12	NFT listado para venda	43
5.13	Seleção de NFT's que poderão sair no pack	43
5.14	Seleção do NFT's que servirá como voucher	44
5.15	Ajuste de quantidades e de probabilidades	44
5.16	Review final do pack	44
5.17	Pack criado e já aberto	45

Capítulo 1

Introdução

BookletNFT é um projeto que surge no âmbito da disciplina de Projeto, do curso Licenciatura em Engenharia Informática e Multimédia. Neste projeto temos como principal objetivo reproduzir em formato digital uma caderneta de cromos como aqueles comercializados pela Panini.

Este projeto faz uso de tecnologias emergentes Web3, tais como *Blockchains*, Cripto Moedas, *Non fungible Tokens* ou NFTs e aplicações descentralizadas ou DAPP. A plataforma é suportada por uma blockchain aproveitando as suas propriedades de segurança, autenticidade e autenticação.

BookletNFT é uma DAPP, hospedada estaticamente que disponibiliza uma plataforma que permite gerar e vender coleções de ficheiros multimédia, numa equiparação a cromos digitais. Para o efeito, a sua coleção e revenda, bem como a criação, disponibilização para venda e compra de packs com vários NFT por pack. A aplicação desenvolvida permite a criação de NFTs, com atributos definidos pelo seu criador.

1.1 Motivação

A grande motivação para este projeto foi o desejo de trabalhar com algo novo, desafiante e com metodologias ainda pouco exploradas. O surgimento do universo META, as menções à Web3 e o crescimento constante das Cripto Moedas e NFTs gerou um sentimento de curiosidade e de interesse na nossa parte.

Assim sendo, muito além de um projeto que simula uma caderneta de cromos, este projeto foi, na realidade, uma oportunidade para explorar as tecnologias até aqui mencionadas, aprofundar conhecimentos numa área em expansão e quem sabe obter bases para caminhos profissionais interessantes.

1.2 Estrutura do documento

Este relatório divide-se em 6 capítulos:

1. **Introdução:** Introduz o projeto e algumas palavras chaves
2. **Trabalho Relacionado:** Aborda temas fundamentais à preparação efetuada que permite implementar o projeto, introduzindo os tópicos teóricos e explorando aplicações já existentes.
3. **Modelo Proposto:** Demonstramos o modelo adotado para a realização do projeto.
4. **Implementação do Modelo:** Explicamos o processo através do qual concretizamos o modelo proposto no capítulo anterior.
5. **Validação e Testes:** Testamos a nossa aplicação, comparando a nossa aplicação com a que pretendíamos no inicio do projeto.
6. **Conclusões e Trabalho Futuro**

Capítulo 2

Trabalho Relacionado

O presente capítulo tem como objetivo abordar todas as partes auxiliares à realização do projeto, expondo ao leitor, técnicas, conceitos, tecnologias e produtos que serviram de auxílio à realização do projeto.

2.1 Blockchain, NFTs e derivados

Por ser um tema novo, não explorado durante o curso, e com tecnologias e conceitos muito diferentes dos que possuímos, foi necessário proceder a uma extensa investigação e estudar os temas associados a esta temática. Este secção visa introduzir o leitor nos temas necessários à compreensão do projeto.

2.1.1 Blockchain

Uma blockchain é uma base de dados digital, que apenas permite leitura e escrita. Cada transação ou instrução é armazenada num bloco que referencia o bloco anterior, através de uma assinatura digital ou *hash*. Estes blocos formam assim uma cadeia de blocos ou blockchain, verificável e imutável. A figura 2.1 mostra o comportamento esquematizado para uma transação na Blockchain Bitcoin.

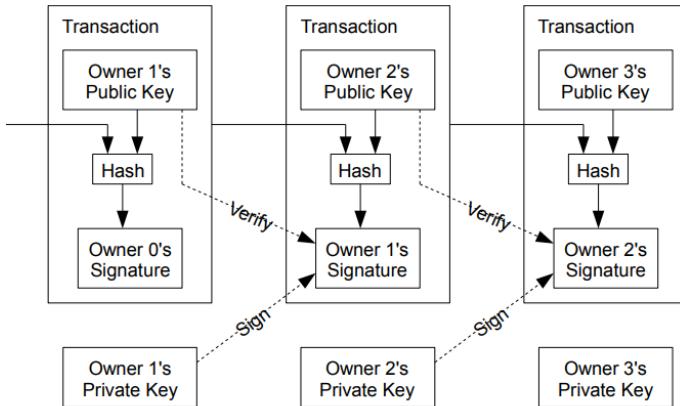


Figura 2.1: Funcionamento da Blockchain Bitcoin [Nakamoto, 2009]

Funcionamento de uma blockchain

Agora iremos explicar qual o funcionamento de uma blockchain, baseando-nos na explicação fornecida pela IBM [IBM, 2022], com referências também ao trabalho que popularizou a tecnologia, a Bitcoin em [Nakamoto, 2009].

Uma blockchain permite a descentralização da informação, ou seja, funciona como uma base de dados distribuída em que todos os participantes têm acesso a toda a informação presente na blockchain e podem validar e confirmar a mesma. Cada bloco numa blockchain pode armazenar qualquer tipo de informação (por exemplo, os NFTs). Qualquer erro num bloco não pode ser alterado sendo necessário a criação de um novo bloco que indique a correção. Ou seja, no contexto deste projeto, a **Blockchain é uma base de dados distribuída, imutável e segura, cuja segurança é garantida por todos os intervenientes.**

Existem diversas blockchains disponíveis como a Bitcoin e IBM Blockchain, passando por Ethereum, Solana, Hyperledger Fabric, Ripple ou Terra. Na tabela 2.1, podemos ver aquelas exploradas por nós e quais as vantagens e desvantagens de cada uma.

Vantagens

Segundo a IBM [IBM, 2022], a utilização da *blockchain* traz vantagens significativas relativamente a outras abordagens tradicionais, nomeadamente:

- **Segurança:** As *blockchains* são inherentemente seguras a nível de trans-

Blockchain	Documentação	Funcionalidades	Ferramentas	Estabilidade	Custo por transação	Escalabilidade	Notas
Ethereum Solana	Boa Boa	Boa Boa	Boa Muito boa	Boa Razoável	Elevados Muito baixo	Má Muito boa	Desenhada a pensar em escalabilidade e NFTs
Ripple	Boa	Razoável	Razoável	Boa	Baixo	Razoável	Muito focada em serviços financeiros
Terra	Boa	Razoável	Razoável	Boa	Baixo	Razoável	Valor da moeda atualmente próximo de 0

Tabela 2.1: Comparação de algumas blockchains

ações e validações devido à sua natureza descentralizada, as provas de trabalho realizadas por todos os intervenientes na Blockchain e o consenso entre estes.

Numa *blockchain* nenhum utilizador é dono da verdade e todos os intervenientes podem validar todas as transações, assim sendo, utilizadores mal intencionados têm de obter o controlo sobre a maioria dos outros intervenientes, tornando-a virtualmente impossível de atacar.

- **Transparência:** Todos os intervenientes têm acesso universal à *blockchain*. Assim sendo, todos os registo são auditados por todos, garantindo que não são realizadas escritas indevidas nesta, nem tão pouco a informação incluída pode ser escondida ou modificada.
- **Automação e “Smart Contracts”:** As transações numa blockchain podem ser automatizadas utilizando “Smart Contracts”. Estes “contratos inteligentes”, permitem aumentar a eficiência, velocidade e segurança das transações de negócios. Por exemplo, um *smart contract* garante a existência dos fundos necessários antes de cada transação ser realizada, seja a compra de um NFT numa data específica ou a criação de uma nova coleção.
- **Rastreabilidade:** Como toda a blockchain pode ser acedida por qualquer interveniente, é garantida a rastreabilidade de todas as transações. É possível verificar todos os atores que interagiram com um bloco em particular de modo a garantir autenticidade.

Ora estas vantagens resolvem instantemente, problemas relacionados com a segurança de dados e de transações sem qualquer custo computacional para

o sistema. No entanto, é sacrificado algum controlo sobre o sistema e até mesmo alguma escalabilidade.

2.1.2 NFTs

UM NFT é uma representação de um objeto, ou a sua referência, agregado a outros atributos, tais como, a informação do seu proprietário. Ora temos como exemplo o NFT, único, vendido pelo maior valor de sempre demonstrado na 2.2 “*Everydays: the First 5000 Days by Beeple*”

Um NFT é um ficheiro JSON armazenado numa blockchain com atributos: *name*, *type*, *imageUrl*, *description*, entre outros. A parte relativa à propriedade é verificada através do rastreio das transações desde o momento em que foi criado o token, vulgarmente conhecido por “*airdrop*”, até à ultima transação onde está presente o endereço da carteira que possui o token.

A associação do NFT a uma carteira constitui a propriedade do dono da carteira do NFT, tal como se se tratasse do livrete de um carro ou registo de propriedade de um imóvel.



Figura 2.2: *Everydays: the First 5000 Days* by Beeple

2.1.3 Ethereum

A Ethereum foi a primeira *blockchain*, bem sucedida no suporte em mais do que a transação de moedas. Com a Ethereum surgiram conceitos tais como “*smart-contracts*” e “*Non Fungible Tokens*”. A Ethereum possui a sua própria moeda, o **Ether**.

O Ethereum tem uma grande comunidade envolvente, com diversos projetos desenvolvidos na área dos NFTs, tais como o CryptoKitties e o OpenSea.

Esta blockchain é relativamente estável, tem diversos recursos online disponíveis, no entanto apresenta desvantagens relativamente ao *Block time* (tempo médio para realizar as transações), e às *gas fees* (custo por transação).

ETH standard ERC-721

Este é o standard mais frequentemente utilizado para a criação de NFTs em Ethereum. O ERC-721 introduz conceitos relevantes tais como tokenId, um identificador único e métodos para a validação de propriedade através de um contract address. Estes conceitos transitam diretamente para o programa SPL utilizado na Candy Machine da Solana.

2.1.4 Ripple

A Ripple é uma *blockchain*, criada a pensar em negócios. Foi desenhada com o objetivo de baixar o custo e aumentar a velocidade das trocas comerciais entre qualquer ponto do mundo, assim sendo, foi desenvolvido “*Ripple Transaction Protocol (RTXP)*” ou “*Ripple protocol*”.

XRP - CriptoMoeda

A CriptoMoeda do Ripple é o XRP e tem como objetivo servir como ponte entre moedas. Estas moedas podem ser “*fiat*” (Moeda fiduciária ou moeda emitida por um banco central, não garantida por metais preciosos, tais como Euro(€) ou Dollar(\$)) ou Cripto Moedas tais como Eth, Sol, etc....

Devido à Ripple ser focada em transações, esta *blockchain*, não suporta NFTs.

2.1.5 Terra

A Terra é uma blockchain de uso geral com a diferença de ser utilizada por moedas estáveis. Estas moedas são mantidas num valor relativamente estável da mesma forma que outras moedas fiduciárias, através da emissão e compra de moeda por uma entidade central.

Luna - CriptoMoeda

Em maio de 2022, a moeda **Luna** não conseguiu acompanhar a inflação do dólar e perdeu a ligação com esta. O que levou á perda de confiança na moeda e na blockchain pelo que o seu valor atingiu 0.

Os NFTs encontram-se num estado “conceptual” e não existem frameworks que suportem NFTs.

2.1.6 Solana

“Solana is a decentralized blockchain built to enable scalable, user-friendly apps for the world.”

solana.com

A Solana é uma Blockchain muito recente, criada com foco na escalabilidade e baixo custo de negócios. Foi primeiro descrita em 2018 por Anatoly Yakovenko no Artigo *“Solana: A new architecture for a high performance blockchain”*[Yakovenko, 2018]. Atualmente, ainda se encontra em Beta, existindo algumas falhas e bugs, notavelmente vulnerabilidade a DDOS e bugs na sua CLI.

Possui frameworks como o Metaplex que permitem a criação de aplicações Web3 com relativa facilidade. Além disso, a comunidade nas redes sociais (Discord), permite o esclarecimento de dúvidas e troca de ideias, ajudando à realização de projetos.

Metaplex

Segundo os autores [Metaplex-Foundation, 2022b]:

“Metaplex is a collection of tools, smart contracts, and more designed to make the process of creating and launching NFTs easier(…)”

Metaplex é uma framework escrita em Node JS, utilizando a CLI disponibilizada pela Solana. Esta framework é utilizada para a criação de aplicações web descentralizadas e coleções de NFTs.

O Metaplex disponibiliza um SDK (disponível em JS, Android e iOS) para a criação de aplicações web descentralizadas. Além disso, está disponível uma ferramenta chamada *Sugar: Candy Machine* que ainda facilita mais a criação de aplicações em JS, nomeadamente *Node JS* e *React JS*, disponibilizando uma framework para a criação de páginas web.

Solana Clusters

Na realidade, a Solana não mantém uma única blockchain. À data da realização do projeto são mantidas pelo menos 3 blockchains distintas: Devnet,

Testnet, Mainnet. Estas 3 blockchains estão distribuídas em “clusters” e correspondem a ambientes de desenvolvimento da Solana.

As blockchains presentes em cada um destes clusters são isoladas dos outros pelo que é impossível transitar Solanas ou NFTs de um cluster para outro.

Devnet A devnet corresponde a uma blockchain para desenvolvedores, ou aqueles que pretendam testar alguma das funcionalidades desta blockchain. Experimentando, por exemplo, a criação de contas usando a API RPC, ou assumindo o papel de validador da blockchain e verificando a integridade desta blockchain.

A principal desvantagem da utilização deste Cluster é que não é prometida a integridade dos dados aqui mantidos, pois a qualquer momento, os desenvolvedores Solana o podem reiniciar.

Outro facto a ter em atenção é que os tokens aqui presentes não possuem valor. Aliás, podem ser simplesmente criados sem qualquer validação previa através de *faucets* ou *airdrop*. É uma ferramenta bastante útil para a criação e desenvolvimento de aplicações reduzindo muito os custos associados aos tokens.

Testnet

A testnet é semelhante á devnet diferenciando-se por ser o cluster onde os desenvolvedores da Solana testam a própria blockchain, tendo este cluster a versão mais recente disponível.

É útil para realizar testes de carga e é frequentemente utilizada para tal.

Não apresenta vantagens em relação à devnet, podendo até ser mais instável.

Mainnet

Como o nome indica a Mainnet é o principal cluster da Solana. É onde as aplicações são lançadas e onde as Solanas têm valor real e não apenas imaginário. A execução de operações neste cluster implica sempre um custo em Solanas pelo que este cluster não deve ser utilizado para testar aplicações.

Este cluster não possui *faucets*, pelo que não é possível fazer *airdrop* de moedas.

2.2 Produtos semelhantes

2.2.1 Cromos Panini



Figura 2.3: Cromos da Panini by Inês Gomes Lourenço

Os cromos da Panini foram sem sombra de dúvida, a principal inspiração para este projeto; Aqueles livrinhos que alguns de nós levavam para o recreio para abrir saquetas, trocar, e completar a coleção. Estas coleções e as interações que elas criavam, foram o principal motivo para o desenvolvimento do projeto.

Numa era cada vez mais digital, com as informações a rumarem para o meio digital, os artigos colecionáveis também começam a migrar para o digital, acreditamos que os cromos não estão ”acabados”, no entanto têm de acompanhar a evolução.

2.2.2 NBA Top Shot



Figura 2.4: NBA Top Shot Promo art

NBA Top Shot [Dapper Labs, 2022] é o projeto que na nossa opinião se aproxima mais daquilo que pretendemos fazer. Neste momento, o NBA Top Shot implementa a quase totalidade dos casos de utilização que nós pretendemos implementar e demonstrar, faltando somente a abertura de packs com sorteios aleatórios.

Nesta aplicação web é possível fazer coleção de “*moments*”; Vídeos de alguns momentos seletos de jogos da NBA, seja pela sua relevância, teatralidade, ou interesse geral. Estes moments podem ser colecionados, trocados e vendidos.

Esta aplicação obteve um sucesso considerável, tendo cerca de 49 mil “*moments*” disponíveis para compra na sua plataforma, com mais de 35 mil já trocados, num total superior a 4.300 milhões de dólares transacionados.

Esta aplicação serviu de inspiração secundária para este projeto, sendo o ponto de referência para os casos de utilização e em conjunto com os “*developer diaries*”, publicados no próprio blog, um ponto de inspiração para a realização do nosso projeto e em certos aspectos um manual.

2.2.3 Crypto Kitties

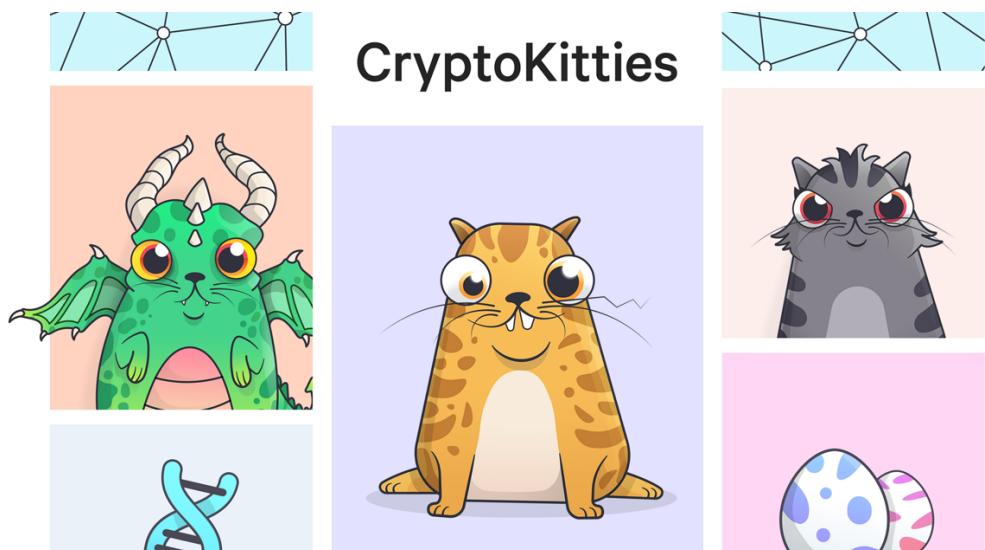


Figura 2.5: Crypto Kitties Promo art

Crypto Kitties é um projeto assente na tecnologia de blockchain em que os jogadores adotam, cuidam e reproduzem gatos virtuais. A propriedade de cada CryptoKittie é armazenada na blockchain Ethereum, e novos “kitties” são adicionados por reprodução de dois “kitties” já existentes, através de um algoritmo de combinação genética.

Este projeto foi pioneiro na área dos smart-contracts, sendo referência basilar sempre que o tema é referido.

2.2.4 OpenSea.io



Figura 2.6: OpenSea Promo art

OpenSea [Ozone Networks, 2022a] é o mais antigo e o maior mercado de NFTs existente aquando da realização deste projeto.

Neste mercado é possível descobrir, colecionar e vender NFTs com relativa facilidade. No entanto, ao contrário do nosso projeto, não é dado um foco a coleções e não é possível a compra de packs. Esta plataforma tem uma comunidade enorme de criadores e desenvolvedores em diversas Blockchains sobretudo Ethereum, mas mais recentemente com suporte para Solana [Ozone Networks, 2022b].

Durante a realização deste projeto foi mesmo pensado utilizar simplesmente a API de developer disponível e desenvolver o projeto como *wrapper* deste mercado.

Capítulo 3

Modelo Proposto

Neste capítulo iremos demonstrar qual o modelo adotado para a realização do projeto, começando por enumerar e explicar os requisitos do sistema, explicando os fundamentos técnicos e teóricos, recorrendo a diagramas quando possível ou mesmo a exemplos quando necessário. Neste sentido, será recolhida e exposta toda a informação necessária para a realização do projeto.

3.1 Análise

A análise de requisitos foi o ponto de partida para a realização do projeto. A partir deste ponto, elaboramos os casos de utilização e os diagramas de atividade e classes. Os requisitos devem ser o mais detalhados possível ao mesmo tempo que mantêm um nível grande abstração. Foram abordados os seguintes tópicos

- objetivos do projeto;
- público alvo;
- metas;
- atributos e funções do sistema;
- casos de utilização;

3.1.1 Objetivos

Pretende-se uma aplicação capaz de permitir a simulação de uma caderneta de cromos, como aquelas vendidas em cafés e papelarias por todo o país sob

os mais variados temas, desde personagens de desenhos animados, passando por jogadores de futebol, automobilismo, entre outros.

A plataforma cria um ambiente capaz de simular a compra, venda, abertura de packs e coleção destes cromos, gerando emoções como intriga, expectativa, frustrações.

Do ponto de vista de exploração e obtenção de conhecimento, este projeto é um caso de uso excelente para a utilização de NFTs e Blockchains, pelo que um objetivo será também a utilização destas tecnologias emergentes.

3.1.2 Público Alvo

O nosso público são colecionadores, pessoas de todas as idades, com algum rendimento disponível, interessadas em determinado tema.

Atendendo à utilização de NFTs para investidores profissionais que podem apostar nas tendências do mercado dos cromos para gerar lucros para si mesmo, e finalmente dum ponto de vista profissional, empresas poderão lançar as suas próprias cadernetas e cromos, como ações publicitárias.

3.1.3 Metas a alcançar

- Criar NFTs diretamente na aplicação
- Criar packs de NFTs
- Criar coleções de NFTs (corresponde a cadernetas), e associar NFTs (cromos) a essa caderneta.
- Conectar carteiras ao nosso sistema de modo a garantir que o utilizador pode realizar as ações que pretende.
- Vender cromos e packs
- Troca de NFT por NFT (objetivo desejável).
- Deployment da aplicação numa página web (objetivo desejável)

Além destas metas, existem muitos outros requisitos de sistema a ter em consideração, tais como requisitos de segurança e funcionalidades menores.

3.1.4 Requisitos do sistema

Tabela 3.1: Requisitos do sistema

Requisito	Função	Categoria	Agrupamento
R 1.1	Permitir a compra de cromos ou NFTs	Visível	
R 1.2	Permitir a compra de Packs de cromos	Visível	
R 1.3	Permitir gerar cromos NFTs	Visível	
R 1.4	Permitir gerar Packs	Visível	
R 1.5	Permitir a troca direta de NFT por NFT	Adorno	
R 1.6	Inserção dos cromos/NFTs numa blockchain	Invisível	NFT e Blockchain
R 1.7	Validação das transações acima mencionadas numa blockchain recorrendo a smart-contracts	Invisível	
R 1.8	Transferir NFTs entre carteiras	Invisível	
R 2.1	Conectar com uma carteira que suporte Solana	Visível	
R 2.2	Suporte a múltiplas carteiras	Adorno	Autenticação e autorização
R 2.3	Garantir a segurança das contas e carteiras	Invisível	
R 3.1	Permitir a consulta das cadernetas e cromos do utilizador autenticado	Visível	
R 3.2	Restringir a consulta de cadernetas consoante a vontade do seu dono	Adorno	Consultas
R 3.3	Permitir a troca direta de cromos	Visível	
R 3.4	Permitir a troca de cadernetas incluindo todos os cromos contidos	Adorno	

3.1.5 Atributos do sistema

Os atributos de sistema podem ser visualizados na tabela 3.2.

Atributo	Detalhe / Restrição Fronteira	Categoria
Resiliente	Situações inesperadas não devem deixar o sistema inutilizável	Desejável
Simples	O sistema deve ser fácil de usar	Desejável
Seguro	O sistema deve proteger os dados que lhe forem confiados, O sistema deve garantir a segurança de todas as transações O sistema deve ser seguro contra ataques informáticos	Obrigatório
Rápido	As operações devem ser realizadas num tempo razoável	Desejável
Escalável	As operações do sistema devem ser reproduzíveis independentemente do número de operações que o sistema deva realizar	Desejável
Plataforma	Solana	Desejável

Tabela 3.2: Atributos do sistema

3.1.6 Casos de utilização

Os casos de utilização surgem em sequência das metas e requisitos de sistema, tendo em conta os atributos definidos anteriormente. Assim, definimos dois tipos de atores distintos: o criador e o comprador ambos estendem o ator utilizador.

Qualquer utilizador terá capacidade de conectar uma carteira, e depois da carteira estar conectada, ver o seu conteúdo (independentemente se comprou ou criou os NFTs) e vender os conteúdos presentes da sua carteira.

O comprador para além dos casos de utilização de utilizador, também pode comprar packs ou NFTs. Já o criador, estende-os com ações relativas à criação de NFTs e ou packs.

Finalmente vale a pena notar que uma pessoa pode ser um criador e um comprador, mas nunca em simultâneo.

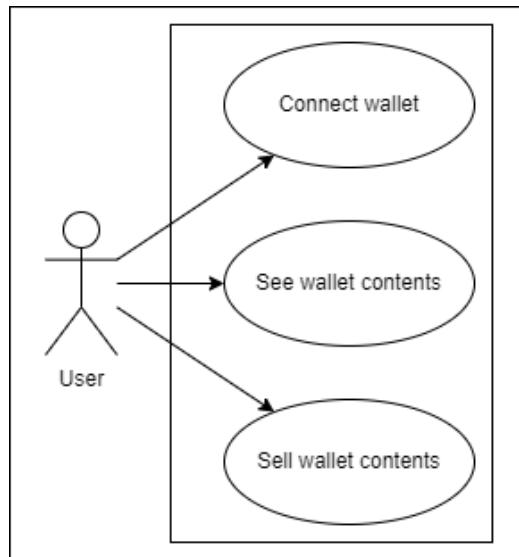


Figura 3.1: Casos de utilização do ator utilizador

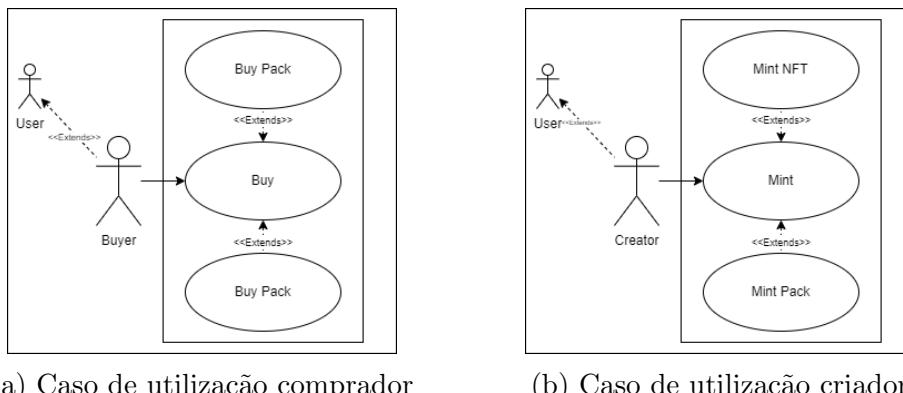


Figura 3.2: Casos de utilização para atores com permissões específicas

3.2 Protótipos Exploratórios

Com este capítulo pretendemos falar um pouco do trabalho exploratório realizado durante o projeto, com o intuito de pesquisar soluções para o nosso projeto, mas por um ou mais motivos acabamos por não implementar na versão final deste. Iremos falar de algumas vantagens e desvantagens das soluções apresentadas tendo sempre como base a nossa própria experiência.

3.2.1 Ethereum

O Ethereum foi a primeira blockchain explorada para a realização deste projeto e uma alternativa bastante viável devido comunidade envolvente, a quantidade de frameworks existentes, e sobretudo a quantidade de tutoriais e guias disponíveis online.

Vale também notar que a escolha da blockchain para a realização do projeto, foi seriamente ponderada entre Solana e Ethereum, sendo que alguns testes foram sendo realizados em paralelo nas duas blockchains.

Como ponto de partida utilizamos o tutorial disponível no próprio website da blockchain Ethereum [Mudgil, 2022] onde ensinam a desenvolver uma loja online para NFTs usando a framework Hardhat. Esta framework permite compilar, executar, testar e fazer o debug de aplicações em Ethereum.

Este tutorial é bastante completo e bastante útil para novos desenvolvedores, pois aborda a maioria dos conceitos necessários ao desenvolvimento de uma DAPP e auxilia bastante à realização de projetos semelhantes ao nosso.

Os capítulos que abordam o tema dos “*smart-contracts*”, foram os que geraram mais dúvidas, nomeadamente a escrita e execução do próprio smart-contract. A quantidade enorme de informação com diversos protocolos, paradigmas e interfaces a serem implementadas gerou uma sensação enorme de confusão e acabámos por decidir não explorar muito mais do que a criação de smart-contracts que implementam o standard ERC-721 [erc, 2022].

A criação de DAPP demonstra-se muito mais complexa em Ethereum do que em Solana. Apesar de os tutoriais serem mais completos e fáceis de seguir, os passos, não o são. Muitas das configurações presentes no tutorial não são as corretas para o SO utilizado por nós e não nos é fornecido comandos alternativos que poderíamos executar. Por este problema, acabamos por adotar uma prática semelhante àquela realizada para a Solana, a utilização de uma máquina virtual Linux, o WSL (que iremos explicar com mais detalhe no capítulo 4.1)

No entanto, algo que nós acreditámos ser vantajoso seria a facilidade com que se criam e se executam smart-contracts. Enquanto que com a Candy Machine da Solana, a estensão de funcionalidades recorrendo a smart-contracts era tido como algo pouco recomendado, em Ethereum a sua utilização era tida como fundamental e fortemente encorajada e ensinada. Acreditamos que se tivéssemos optado pela utilização desta blockchain, poderíamos ter

mais espaço para expandir as funcionalidades no futuro, nomeadamente cromos com funções especiais, expansão para jogos online como por exemplo Football Manager em NFT, aplicação para o universo Meta, entre outras.

A quantidade de sistemas envolvidos, nomeadamente IPFS, Hardhat e Alchemy, torna a utilização desta blockchain algo complexa, ainda que muito mais poderosa, que aquela que acabamos por adotar, mas sem dúvida o principal motivo para o abandono desta blockchain foram as *gas fees* incrivelmente em certas alturas. Estas *gas fees* não são passíveis de serem ignoradas quando pretendemos ter uma aplicação que gere lucros. Assim sendo, decidimos que seria melhor optar por uma blockchain cujas transações são mais baratas.

3.2.2 Ripple

O Ripple parecia uma blockchain bastante promissora para a realização deste projeto devido ao investimento muito elevado que os seus criadores estavam a realizar aquando da realização do projeto. Investimento este, que gerava uma sensação de confiança na própria blockchain e nos próprios projetos que poderiam vir a existir.

No entanto, a ausência de informação e comunidades online para a realização de projetos semelhantes deixou-nos reticentes em relação a esta blockchain. Ainda assim, decidimos realizar alguns testes com esta blockchain.

Seguindo o tutorial disponibilizado pelos próprios criadores [LEDGER, 2022] temos uma *framework* com um número bastante reduzido de funções. Enquanto que ao utilizar *frameworks* em Solana ou Ethereum teríamos mais funcionalidades que as necessárias, também teríamos muitas bastante uteis como conectar com uma Wallet num browser diretamente através do SDK, ou procurar Tokens por atributos tais como criador, ou data de criação. Nenhuma destas funcionalidades existem nativamente, sendo necessária a implementação da nossa parte destas funcionalidades, aumentando a complexidade da utilização desta blockchain.

Tentamos perguntar em grupos online, nomeadamente grupos de Discord, como realizar tais funções e sobre a existência de frameworks que poderíamos utilizar, no entanto, não obtivemos respostas de qualidade. Acabámos então por decidir contra prosseguir com o desenvolvimento nesta blockchain.

3.2.3 Solana CLI, SPL-Token e Metaplex SDK

As primeiras tentativas de executar algo em Solana foram através da utilização da sua Comand Line Interface abreviada para CLI. A CLI possui um conjunto de métodos bastante alargado e poderoso, nomeadamente funções para a geração de tokens, airdrop (não disponível na MainNet) e transferência de tokens.

Utilizamos a CLI para executar testes rápidos tais como gerar Tokens sem imagens associadas, trocar tokens rapidamente entre carteiras e em certos pontos do desenvolvimento verificar se as transações na DAPP estavam a ser devidamente realizadas devidamente.

Existe também uma ferramenta chamada SPL-token. Esta ferramenta é notavelmente semelhante á CLI sendo mesmo em alguns casos apenas um *wrapper* desta. A diferença assenta no facto do programa SPL-token, ser feita para usar o contrato *SeaLevel* enquanto a CLI trabalha diretamente em cima da Solana. A utilização semelhante, comandos semelhantes e comportamento semelhantes gerou alguma confusão e alguns problemas nomeadamente a incompatibilidade de tokens, e por vezes comportamentos inesperados. Ainda assim a SPL-token é uma ferramenta incrivelmente útil e foi também utilizada em conjunto com a CLI para validar alguns comportamentos, nomeadamente se os tokens estavam a ser criados com o conteúdo correto e se a transferência de tokens foi bem-sucedida.

No entanto sabíamos perfeitamente que não seria viável realizar a aplicação somente através da utilização aplicações executadas apenas em linha de comando. Assim depressa partimos para a utilização da API para JavaScript existente na documentação oficial. A utilização desta ferramenta demonstrou-se tão complexa como a utilização da própria Ethereum, sendo a quantidade de dados incrivelmente alta. Ainda assim, realizamos alguns testes, tendo mesmo criado uma pequena página web, utilizando HTML e JS, capaz de criar uma carteira para um utilizador, e mostrar o endereço dos tokens que este possuía. Foi durante a realização destes pequenos testes que nos foi apresentada a Metaplex - Candy Machine.

A utilização da API e da CLI permite-nos compreender o funcionamento interno da Candy Machine, e mesmo que não a tenhamos utilizado diretamente na criação DAPP, tentamos utilizar para a expansão das funcionalidades, nomeadamente na criação dos packs.

O SDK desenvolvido pela Metaplex Foundation permite a criação de aplicações de raiz, executando operações na blockchain chamando diretamente os métodos disponibilizados. Com esta SDK a Metaplex tenta criar uma framework que pode ser utilizada para realizar alguns casos de utilização comuns tais como a criação de tokens e sua venda, procurar na blockchain NFTs através de algum atributo tais como o seu criador ou dono. É uma ferramenta bastante útil, no entanto ainda está em desenvolvimento ativo e á altura do desenvolvimento do relatório não trazia vantagens em relação á Candy Machine, pelo que iria aumentar a complexidade do projeto, sem qualquer vantagem.

Packs e Smart Contract

Como mencionado nos objetivos, uma das funções pretendidas seria a implementação de packs com sorteio aleatório. De modo a realizar esta função é necessária a expensão das funcionalidades fornecidas pela framework Metaplex.

De modo a realizar este objetivo, em primeiro lugar tentamos implementar e fazer deploy de um *smart contract*. Assim escrevemos um algoritmo em Rust, que nos permitiria atribuir pesos, entre 0 e 1 (sendo 0, impossível de sair e 1, sai sempre) para um conjunto de masters. Este smart contract iria ser realizado sempre que se procedesse á abertura de um Pack, realizando a destruição ou “*burn*” do token correspondente ao pack e o *minting* e troca automática dos NFTs criados. Tal processo pode ser esquematizado na figura 3.3

Como podemos verificar no diagrama, a blockchain verifica as condições do utilizador antes de iniciar o smart-contract, pois o smart contract é instanciado pela própria blockchain e não diretamente pelo utilizador. Denota-se que a atribuição dos NFTs é feita a um endereço de uma carteira dentro da blockchain, e não ao utilizador diretamente. Essa carteira por sua vez é propriedade do utilizador. E o utilizador, por transitividade, é dono dos NFTs presente na sua carteira.

Outra consideração a ter em conta é a destruição do NFT ou token do pack. Assim sendo, a destruição do token, foi implementada por nós, como uma transferência para uma carteira ”lixo”. Esta carteira, após ser gerada, é lhe imediatamente apagada as credenciais privadas pelo que estas nunca

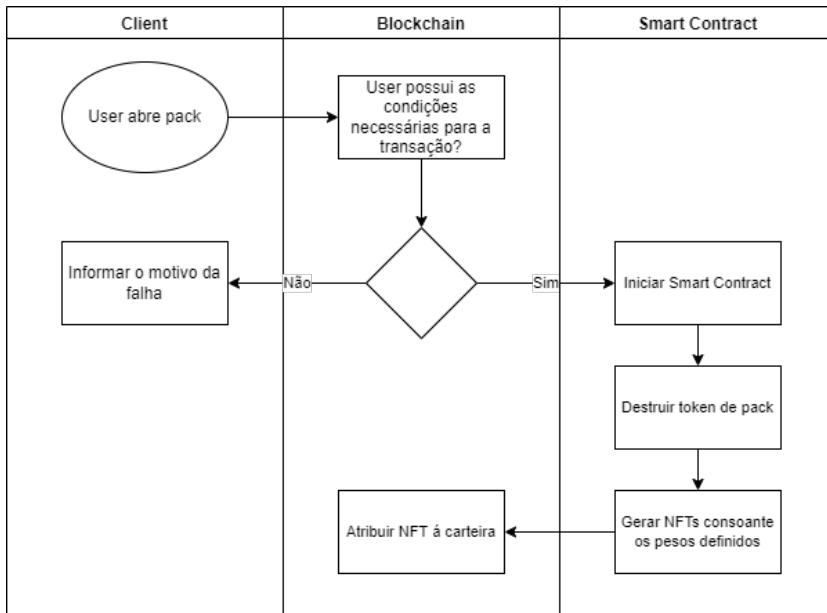


Figura 3.3: Diagrama *Swim Lane* do funcionamento do smart contract

poderão ser acedidas por utilizadores mal intencionados. Além disso, também seria possível incluir no próprio smart-contract uma verificação pelo número do token (que é único, gerado a quando da sua criação), garantindo que o mesmo pack não é aberto mais que uma vez.

Finalmente o grande problema desta abordagem. A geração de NFTs consoante o peso.

Supondo que o criador dos packs pretende criar um pack com 3 cromos no seu interior de um universo de escolha com N elementos, sendo que cada elemento é equiprovável na sua distribuição. A escolha de 3 elementos além de apresentar a possibilidade de levar a elementos repetidos também exige para 3 elementos $3+M$ transações. Assim sendo os custos de transação (“gas fee”) aumentam linearmente consoante a indisponibilidade da blockchain e existe inclusive a possibilidade dos fundos do utilizador se esgotarem sem que o smart contract seja cumprido, impossibilitando a conclusão deste e deixando a execução “pendurada”, possivelmente até que o utilizador introduza mais fundos na carteira.

Devido a este último problema acabamos por riscar esta abordagem acabando por utilizar uma implementação da própria Metaplex, que curiosamente sofre do mesmo problema. Esta implementação pode ser vista na

secção seguinte 4.2.2

Nota: À data da realização do projeto, esta funcionalidade era incrivelmente instável na Candy Machine pelos problemas relatados em cima. No entanto no novo SDK, parece que muitos destes problemas foram ultrapassados e o feedback em chats online tem sido bastante positivo pelo que certamente será um comportamento a modificar no futuro, implementando uma versão melhorada dos packs. Mais informação pode ser obtida através do repositório [Metaplex-Foundation, 2022c].

3.3 Opções tomadas

Com este capítulo tentamos explicar algumas das escolhas que tomamos durante a realização deste projeto.

3.3.1 Solana & Metaplex Candy Machine

A escolha da Solana advém de diversos fatores, notavelmente ao baixo custo de transação e velocidade com que estas são realizadas.

A Solana desenvolveu por iniciativa própria a fundação Metaplex que disponibiliza um diverso leque de ferramentas que auxiliam à criação de aplicações, reduzindo a complexidade necessária à execução do projeto.

Além destes motivos, a comunidade envolvente que, o contacto direto com os desenvolvedores da blockchain, e a velocidade com que nos respondiam foram vantagens inigualáveis que não era fornecido em mais nenhuma blockchain.

Em resumo os motivos que levaram à nossa escolha foram:

1. As *gas fees* reduzidas.
2. A comunidade envolvente
3. A facilidade de utilização das ferramentas
4. O foco da própria blockchain em NFT

3.4 Abordagem

3.4.1 Processo de desenvolvimento

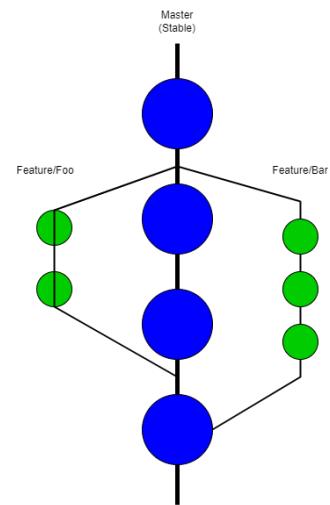
Controlo de versões

De forma a garantir o desenvolvimento paralelo em equipa do projeto, enquanto se gerem as versões do projeto realizado, foi adotada a utilização de software de controlo de versões, no nosso caso o **GIT**, com o Github como repositório.

A nível de *branchs* tentamos manter o esquema o mais simples possível, mantendo geralmente apenas 1 branch o *master*. Na realidade, por existir alturas em que ambos trabalhávamos em paralelo, foi necessária por vezes, a criação de outros *branchs* sempre no esquema de *feature/<<feature-name>>*, onde feature-name corresponde ao nome da questão a ser trabalhada. A representação do nosso controlo de versões encontra-se na figura 3.4.

A utilização deste sistema não é possível sem alguns problemas, nomeadamente *merge conflicts*. Por vezes o retorno ou *merge* com o master não gerava conflito como é o caso do branch *feature/Foo*, sendo a sua junção a simples união dos dois *branchs*, outras vezes existia um *merge conflict* que era resolvido caso a caso, como no branch *feature/Bar*, selecionando os autores quais as partes de quais os *branchs* em conflito a manter.

Figura 3.4: Modelo representativo da estrutura do repositório



3.4.2 Arquitetura do sistema

Tendo em conta as potencialidades da blockchain escolhida para a realização do projeto o modelo que nós propusemos, considera a **Blockchain como a própria base de dados e backend**. Assim sendo, o nosso projeto apenas manipula informação na blockchain e faz uso das potencialidades disponíveis nesta.

O modo como o sistema funciona pode ser representado pelo esquema presente na imagem 3.5

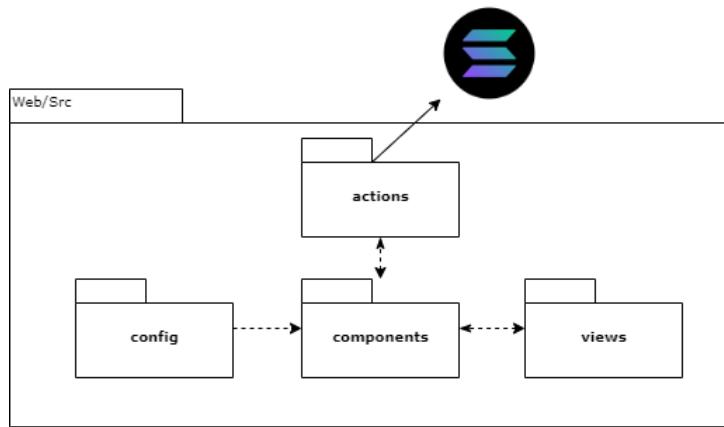


Figura 3.5: Modelo de pacotes proposto para a realização deste projeto.

Neste esquema podemos reparar que o sistema no seu todo é composto por 4 packages distintos que interagem entre si.

1. O 1º package **config** que possui todas as configurações que não devem ser incluídas no código, como endereços de carteiras, repositórios, e urls onde o programa será hospedado.
2. O 2º package **components** contem á lógica da nossa aplicação. Nomedamente é contida toda a parte relativa ao processamento das informações introduzidas pelo utilizador.
3. O 3º package **actions** corresponde á comunicação com a blockchain. Neste package são efetuadas as chamadas à CLI da Solana. As respostas obtidas são posteriormente enviadas para o package components onde irão ser transformadas e exibidas ao utilizador.
4. O 4º package **views** corresponde á parte visível do website. Onde os components são recebidos, os estilos são lhes aplicados e os eventos provocados pelo utilizador são capturados.

Este modelo é o padrão da framework Metaplex.

3.4.3 Distribuição

Docker

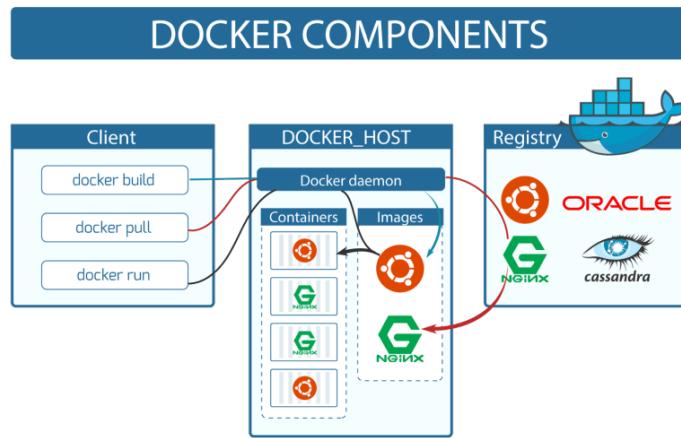


Figura 3.7: Componentes do docker segundo [Berlatto, 2020]

A instalação e execução deste projeto demonstrou-se muito problemática. Desde documentação incorreta que instala versões incompatíveis com outros componentes do projeto passando por inconsistências entre as máquinas dos autores. De modo a resolver esta questão introduzimos o **Docker**.

O Docker permite criar uma camada uniformizadora entre as várias plataformas de desenvolvimento. Em conjunto com a aplicação envia também todo o sistema operativo, instala dependências e executa a aplicação, criando uma **imagem**. Desta forma temos uma aplicação executada de forma consistente e previsível entre máquinas distintas.

De uma maneira simplificada o funcionamento do Docker é dividido em 3 componentes esquematizado na figura 3.7:

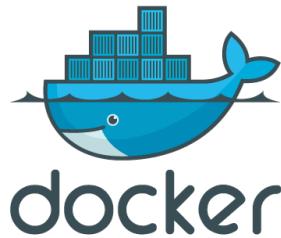


Figura 3.6: Docker

1. O repositório ou “*registry*”. É o local na cloud onde as imagens das aplicações são armazenadas. Assim sendo o nosso projeto na realidade possui dois repositórios distintos que devem funcionar em conjunto para a execução da aplicação.
2. O cliente. O cliente executa um conjunto de ações que disponibiliza para posterior execução pelo daemon. Podem ser executados comandos

para criar a imagem localmente, armazenar no repositório e obter as imagens presentes neste, respetivamente “*build*”, “*pull*”, “*run*”.

3. O daemon. Corresponde ao cérebro do docker. É onde são controlados os **containers** que executam as imagens. Um container executa uma única imagem, no entanto uma imagem pode fazer uso de diversas outras imagens para executar.

Neste projeto, iremos utilizar o docker como uma forma para distribuir a nossa aplicação. Desta forma, esperamos nós obter uma aplicação que execute de uma maneira fiável e consistente independentemente da máquina em que seja executado, reduzindo a execução deste a 3 passos. Instalar o docker, fazer pull da imagem, abrir a página web.

Capítulo 4

Implementação do Modelo

4.1 Obtenção de ferramentas e configurações iniciais necessárias

A grande dificuldade na implementação dos modelos propostos no capítulo anterior, não foi a escrita do código, mas sim a obtenção, configuração e execução das ferramentas necessárias à sua realização.

O primeiro problema surgiu aquando da instalação do SDK da Blockchain. Começando pelas suas dependências, nomeadamente, **Node JS**, **Yarn** e **Type Script**.

Apesar de ser possível instalar o NodeJS sem problemas, já a instalação do Yarn e TypeScript revelaram diversos problemas.

A instalação do yarn usando o gestor de pacotes do Node JS, o NPM, resulta numa versão não suportada pelos outros componentes do projeto. Foi necessário recorrer a fóruns para identificar qual a versão compatível a instalar, removendo a versão incompatível. Por outro lado, a versão mais recente do Type Script, ou seja, aquela disponibilizada para o Windows, não é compatível com os outros componentes do projeto. Assim sendo, pensamos em executar e instalar através do código fonte. Tal não se demonstrou viável, pois ia ser necessária a instalação de ainda mais componentes e compiladores externos. O resultado final depende das versões NodeJS 16.14.2, Yarn 1.22.18 e Type Script 10.4.0

Face a estas dificuldades decidimos utilizar o OS, para o qual as ferramentas foram criadas, O Ubuntu. De modo a facilitar a utilização desta ferramenta, optamos por usar as potencialidades do Windows nomeadamente

o **WSL, Windows Subsystem for Linux**. Através da utilização desta “*mini-máquina virtual*” somos capazes de executar a maioria dos comandos disponíveis em Linux, mais especificamente Ubuntu 20.04.

Assim sendo, o nosso projeto foi realizado em máquinas windows, recorrendo a uma máquina virtual linux, utilizando ferramentas desenvolvidas pela Metaplex Foundation, que utiliza o SDK da Solana, trabalhando diretamente em cima desta blockchain. Este processo pode ser esquematizado na imagem 4.1.

Figura 4.1: Modelo representativo da interação dos componentes



Outras ferramentas uteis ao desenvolvimento foram:

- **VS Code** - Editor de texto com uma enorme gama de ferramentas e extensões disponíveis para auxiliar o desenvolvimento nas mais diversas linguagens. No nosso caso utilizamos extensões de modo a facilitar a criação de programas em Node JS e React. Além disso, possui também extensões para Latex pelo que o próprio relatório acabou por ser escrito neste editor.
- **React Devtools** - Uma extensão para o browser que permite explorar e visualizar os componentes escritos em React. Desta forma conseguimos explorar e identificar quais os componentes que já existiam, o que eles faziam e o que era preciso alterar.
- **Phantom** - Uma carteira para CriptoMoedas. Foi necessário a sua utilização de modo a armazenar os tokens criados e fundos necessários á criação
- **Discord e StackOverflow.com** - Plataformas de comunicação. Apesar de não terem impacto direto no desenvolvimento, foram plataformas bastante uteis para a troca ideias com a comunidade em torno da blockchain e das frameworks utilizadas por nós.

4.2 Desenvolvimento inicial

Inicialmente o desenvolvimento seguiu o processo descrito na documentação oficial disponibilizada em [Metaplex-Foundation, 2022a]. De notar que a ferramenta Candy Machine V2 utilizada por nós encontra-se *deprecada* desde o final de Agosto de 2022, estando o seu substituto, o Sugar presente no mesmo domínio online. Dada a proximidade da entrega optamos por não reimplementar o projeto na nova versão, notando apenas a possibilidade de o fazer.

Vale a pena notar que todo o processo de desenvolvimento foi realizada rede DEVNET da Solana. Esta rede fornece uma blockchain isolada da blockchain principal, ainda que com os mesmos protocolos e funcionalidades. Esta rede é fornecida gratuitamente pelo que existem falhas consideráveis de *uptime* ou tempo útil. Apesar disto, os custos muito reduzidos permitem-nos realizar operações com tokens e moedas reais, ainda que sem valor monetário e sem a promessa de persistência.

Após a instalação das dependências necessárias para a realização do projeto, clonamos um repositório do git que serve de template à criação de uma loja online. O repositório contém diversos componentes escritos em React que fornecem algumas funcionalidades genéricas.

4.2.1 Funcionalidades genéricas

Conexão de carteiras

Aproveitámos a componente de conexão de carteiras do Metaplex, apenas alterando o estilo.

A conexão das carteiras funciona através da API da Solana em conjunto com a API disponibilizada por uma extensão do browser. Essencialmente, a função procura no browser qual o endereço da carteira ativa e guarda o endereço público desta numa variável de sistema no browser (*localStorage*). Ao conectar a sua carteira, o utilizador está também a autenticar-se, pelo que, esta função de login é relativamente mais simples que outras por combinação de password e senha, sendo apenas necessário a utilização de uma carteira que já possui outros métodos de verificação e controlo, alienando assim a responsabilidade da nossa aplicação destes critérios.

Minting

A criação de NFTs também chamado de “*minting*” é uma funcionalidade que vem inclusa na framework. O NFT criado tem diversos atributos que podem ser visualizados na tabela 4.1

Nome do atributo	Formato	Descrição
Creators	Array	Array de objetos que identifica os criadores do token
Name	String	
Symbol	String	String representativa do token
isItParent-Collection	Boolean	Se o token for capa de uma coleção este atributo é tido como verdadeiro
Collection		Identifica qual a coleção em que o NFT se inclui.
SellerFee	Float $\in [0, 1]$	Valor pago em percentagem ao criador do NFT, durante cada transação.
Description	String	
Data	Object	Conteúdo do NFT. Obrigatório este atributo ter valor, se o uri for vazio.
uri	Binary String	Url com a localização do conteúdo binário do NFT. Obrigatório este atributo ter valor, se o Data for vazio

Tabela 4.1: Tabela descritiva dos principais atributos de um NFT

O criador destes NFTs consegue definir diversos atributos tais como nome, símbolo, se é o token principal da coleção e se lhe é pago “*royalties*” através da seller fee. Existem diversos atributos, sendo que alguns deles são compostos por outros objetos JSON.

Salientamos os atributos Data e o URI. O conteúdo associado à propriedade do NFT pode ser armazenado de uma de duas formas:

1. **Data:** Com o conteúdo binário. Possui gas fees mais elevadas e dá um maior controlo sobre o conteúdo.
2. **URI:** Com um apontador. Gas fees mais reduzidas, mas dependente de serviços externos para o armazenamento dos dados. É o inverso do Data.

O criador pode pesar estas duas opções e escolher a que mais benefícios lhe traga. Na opinião dos autores é preferível incluir todo o conteúdo binário no NFT ainda que as *gas fees* sejam maiores.

Compra e Venda de NFTs

Existem dois tipos de venda de NFT's nomeadamente a venda direta e o leilão. Neste projeto, só trabalhamos com a venda direta uma vez que considerámos que a venda por leilões não se enquadrava com a finalidade do projeto.

A compra de um NFT, no contexto da nossa aplicação apenas considera a venda através de *marketplace*, assim sendo, para comprar um NFT o utilizador acede a um *vault* onde estão listados todos os NFTs que todos os utilizadores pretendem vender. O primeiro utilizador a comprar o token pelo preço definido pelo vendedor fica com o NFT.

A venda de um NFT segue um processo semelhante. O proprietário do NFT define o cromo a vender, define o preço e pode inclusive definir uma data futura em que o token pode ficar disponível. A venda do token assenta num *smart contract* em que a venda é processada através de uma carteira intermediária. O processo é melhor visualizado nas figuras 4.3 e 4.2.

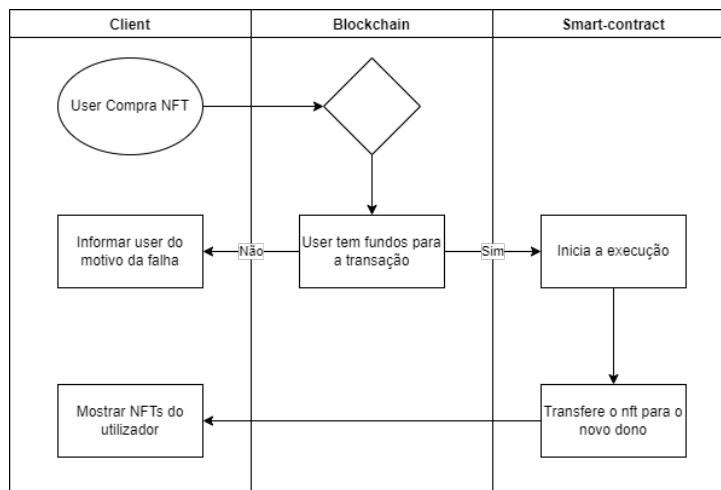


Figura 4.2: Diagrama *Swim Lane* do funcionamento de compra de um NFT

Segundo os diagramas podemos ver as responsabilidades de cada intervingente na realização das transações. O utilizador interage com o *client* que

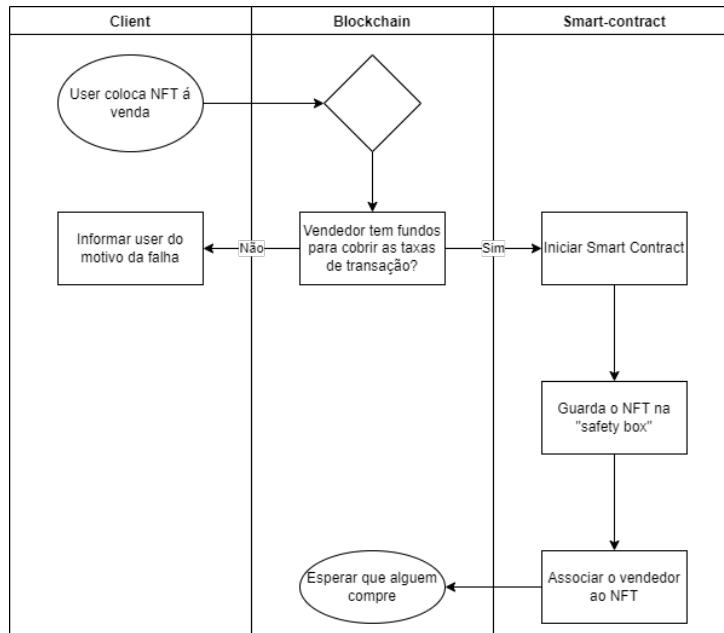


Figura 4.3: Diagrama *Swim Lane* do processo de venda de um NFT

pode ser um browser em qualquer dispositivo suportado pela aplicação, o client comunica com a blockchain através do SDK e a própria blockchain verifica as condições da transação. O smart contract serve de fiel depositário dos NFTs em questão. A sua execução implica que o token seja armazenado no vault aquando da sua disponibilização para venda, e no momento da compra, transfere os fundos recebidos para o vendedor.

4.2.2 Expansão de funcionalidades

Implementação Packs

A implementação dos packs recorreu a uma implementação presente no próprio repositório. Esta componente não se encontra ativa por defeito.

Em 1º lugar gostaríamos de mencionar que o funcionamento desta *feature* se deve em grande parte à comunidade de apoio em grupos online.

O processo de desenvolvimento iniciou-se com a criação do “path” que iria mostrar os packs, criamos uma rota capaz de indicar a página e anunciamos a disponibilização desta rota no ficheiro correspondente.

Em seguida, fizemos enable da funcionalidade, modificando uma variável de ambiente permitindo a criação e comercialização de packs. Tendo esta

feature sido *enabled*, usamos o esqueleto de pagina web já desenvolvida para o efeito e aplicámos os estilos pretendidos.

A venda de Packs foi realizada da mesma maneira que a venda de NFTs normais, sendo a sua principal diferença o conteúdo do NFT. O conteúdo deste NFT é nada mais que um dicionário que mapeia referencias a NFTs com a probabilidade destes serem atribuídos. Além deste dicionário existe também o próprio voucher que irá corresponder à capa do pack.

Concluindo, um pack corresponde a um NFT que pode ser redimido pelo utilizador que o possua, sendo contido no seu interior todas as informações necessárias à geração de novos NFTs ou atribuição de NFTs já criados presentes num *vault*.

De qualquer das formas obtemos um NFT que não conhece quais os cromos contidos no seu interior antes da sua abertura. Aumentando ao máximo a equidade na atribuição dos cromos. E ainda assim as probabilidades da atribuição de cada cromo podem ser verificadas por qualquer utilizador da blockchain, garantindo assim a transparência do sorteio.

Capítulo 5

Validação e Testes

Após concluir o desenvolvimento do projeto, decidimos testar e validar as funcionalidades deste. Com este capítulo tentamos então responder a questões como “Quais os objetivos cumpridos?”, “Os casos de utilização foram seguidos?” etc....

Dada a natureza relativamente simples do projeto, optamos por realizar testes manuais e não implementar testes automáticos com ferramentas como *Selenium* ou *Cypress*. A utilização desta ferramenta traria valor de um ponto de vista de desenvolvimento de software, no entanto não nos foi possível implementar a tempo todos os casos de utilização pelo que optamos por não incluir na entrega..

5.1 Mint de NFT

Começamos por testar o mint de um NFT como se pode ver pelas figuras seguintes.

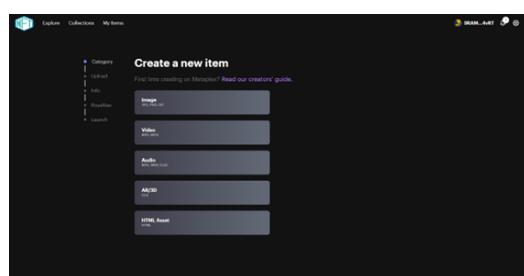


Figura 5.1: Selecionar o tipo de asset que vai ser o NFT

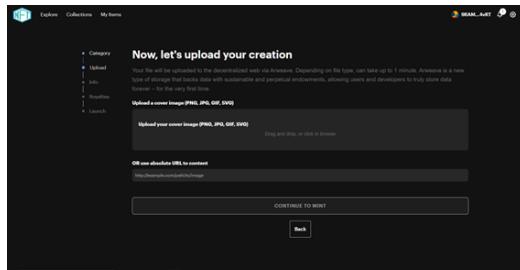


Figura 5.2: Fazer upload do ficheiro

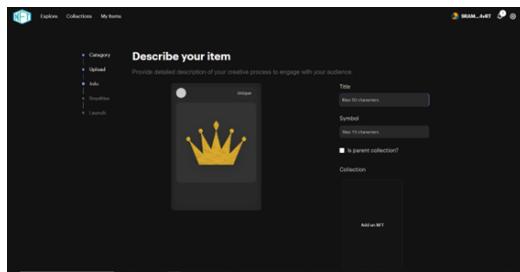


Figura 5.3: Definir atributos do NFT

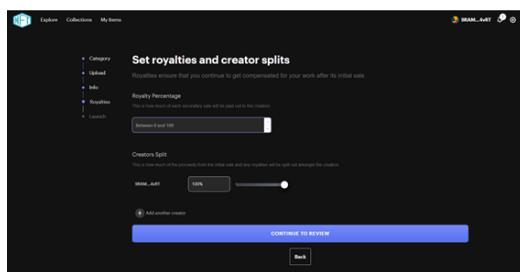


Figura 5.4: Quais os *royalties* a serem pagos

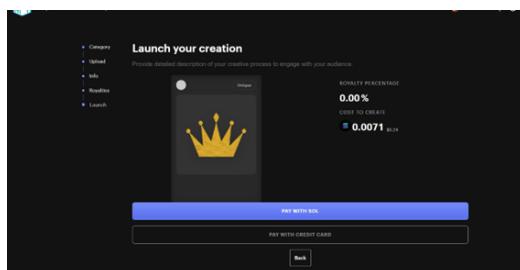


Figura 5.5: Confirmação e pagamento do lançamento

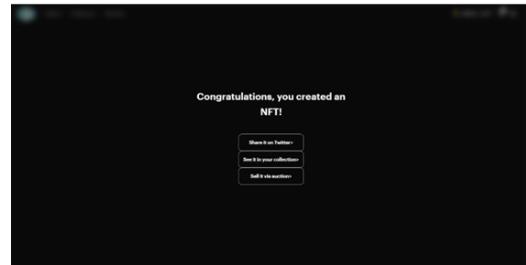


Figura 5.6: Página de sucesso

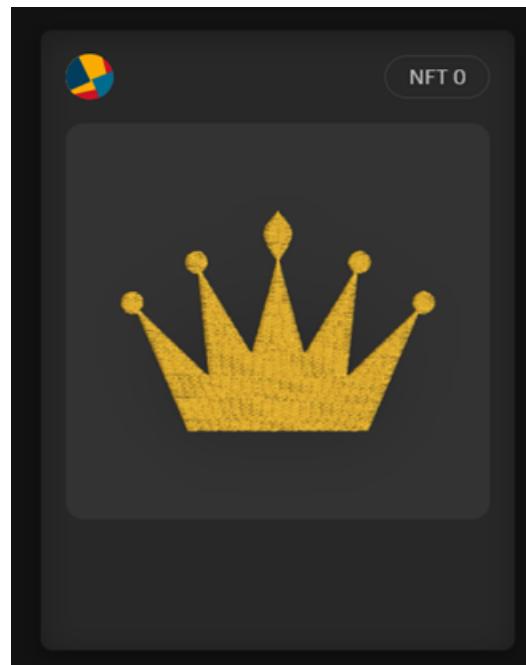


Figura 5.7: NFT na carteira

5.2 Venda de um NFT

De seguida testamos se conseguimos pôr um NFT à venda as seguintes figuras representam os passos necessários para colocar um NFT à venda.

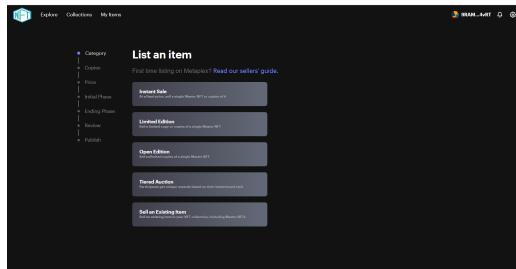


Figura 5.8: Tipo de venda

Como foi mencionado anteriormente só foi usado a venda direta uma vez que se achou que os leilões não se adequavam com a finalidade do projeto.

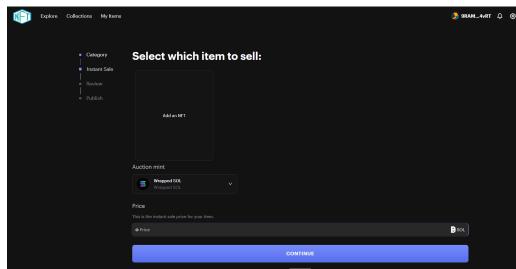


Figura 5.9: Selecionar o NFT que se pretende vender e o seu em preço em solanas

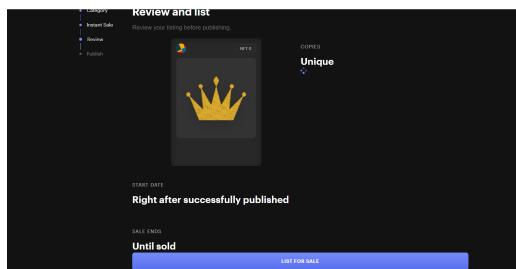


Figura 5.10: Review final da venda

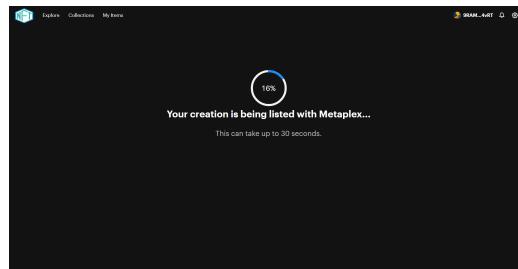


Figura 5.11: Nft a ser listado para venda

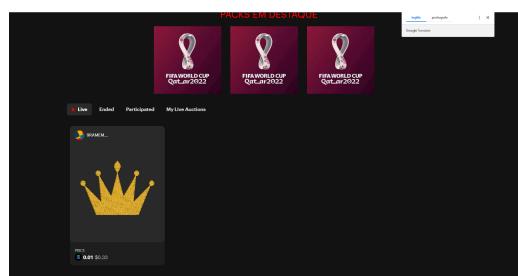


Figura 5.12: NFT listado para venda

5.3 Criação e abertura de packs

De seguida foi testada a criação e a abertura dos packs, como foi referido anteriormente os packs correspondem a um NFT que se chama de voucher e quem possuir uma cópia desse determinado NFT poderá abrir o pack.

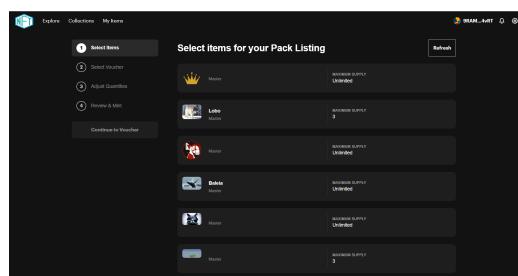


Figura 5.13: Seleção de NFT's que poderão sair no pack

Com estes testes podemos concluir que conseguimos entregar a maioria dos casos de utilização propostos, nomeadamente a criação, troca, venda, de NFTs e a comercialização de packs que geram os seus próprios NFTs.

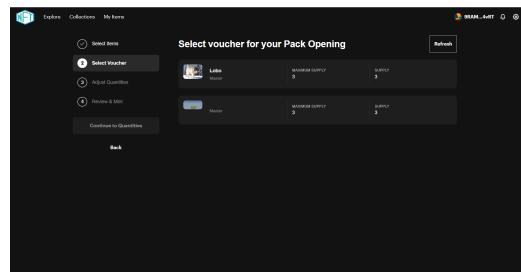


Figura 5.14: Seleção do NFT's que servirá como voucher

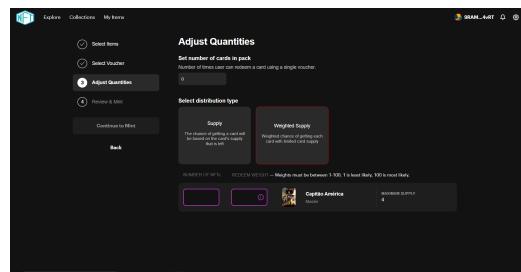


Figura 5.15: Ajuste de quantidades e de probabilidades

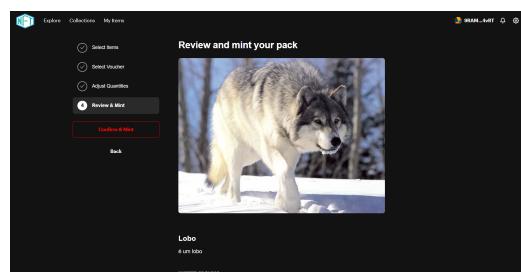


Figura 5.16: Review final do pack

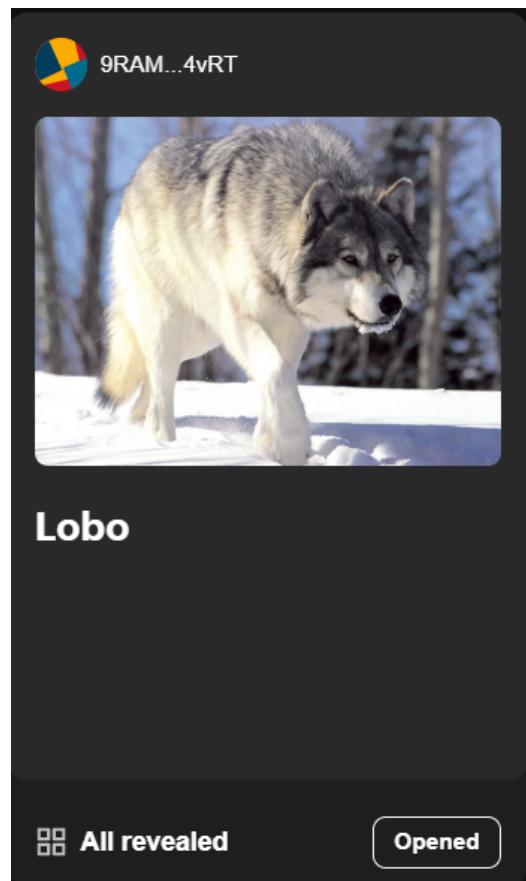


Figura 5.17: Pack criado e já aberto

Capítulo 6

Conclusões e Trabalho Futuro

Este trabalho permitiu-nos explorar e aprender largamente sobre criptomoedas, blockchains e NFTs. Apesar de não ter sido possível cumprir com todos os objetivos propostos, acreditamos que o próprio projeto é capaz de servir de fundação os restantes objetivos.

A partir do contacto direto com a blockchain, a pesquisa numa área recente com propensão a mudar muito rapidamente, aprendemos a adaptar-nos, mudar os comportamentos até ali desenvolvidos e mudar mentalidades para a melhor execução do projeto.

Apesar de ter conseguido cumprir com a maioria dos objetivos propostos, a existência de bugs na própria blockchain, o facto de não termos sido nós a criar a totalidade do código e estarmos a adaptar algo já existente, a nossa inaptidão com React e Node levou à entrega de um projeto com bugs não desejáveis, mas que ainda assim efetua uma prova de conceito que poderá ser expandida no futuro.

No entanto, partindo de um ponto de vista não puramente informático, este projeto provocou em nós uma reflexão sobre estas tecnologias, dando origem a ideias para trabalhos futuros, em diversas áreas, de diversas formas. Por exemplo a utilização de blockchain para novos projetos, nomeadamente rastreio de cadeias logísticas ou até possivelmente integrar com IoT e tentar criar um sistema de vendas de bilhetes para concertos, aumentando a confiança na troca destes, entre muitas outras ideias que possam surgir.

As Blockchains e as CriptoMoedas são definitivamente, mais que uma moda passageira cujo potencial ainda não foi, nem de perto, completamente explorado. Exemplos de trabalho futuro nesta área pode ser além daqueles

já mencionados em cima, rastreio de produtos por cadeias de produção e distribuição, emissão de documentos online, rastreio de atos médicos, venda de bilhetes para eventos, etc ...

Esperamos com este projeto ter demonstrado as potencialidades da blockchain, criptomoedas e NFTs além de conhecimento obtido nesta área e de sistemas informáticos no seu geral.

Bibliografia

- [erc, 2022] (2022). Erc-721 non-fungible token standard.
- [Berlatto, 2020] Berlatto, L. (2020). Primeiros passos com docker: Conceitos básicos e criação da sua primeira imagem.
- [Dapper Labs, 2022] Dapper Labs, I. (2022). Nba top shot. <https://nbatopshot.com/about/>.
- [IBM, 2022] IBM, I. (2022). Benefits of blockchain - ibm blockchain. <https://www.ibm.com/topics/benefits-of-blockchain>.
- [LEDGER, 2022] LEDGER, X. (2022). Nftoken tester tutorial.
- [Metaplex-Foundation, 2022a] Metaplex-Foundation (2022a). Candy machine js cli - introduction.
- [Metaplex-Foundation, 2022b] Metaplex-Foundation (2022b). Metaplex docs.
- [Metaplex-Foundation, 2022c] Metaplex-Foundation (2022c). Metaplex nft packs solana program.
- [Mudgil, 2022] Mudgil, S. (2022). How to write & deploy an nft.
- [Nakamoto, 2009] Nakamoto, S. (2009). Bitcoin: A peer-to-peer electronic cash system. *Whitepaper*.
- [Ozone Networks, 2022a] Ozone Networks, I. (2022a). Opensea.io. <https://opensea.io/>.
- [Ozone Networks, 2022b] Ozone Networks, I. (2022b). Opensea.io beta supports solana. <https://opensea.io/explore-solana>.

[Yakovenko, 2018] Yakovenko, A. (2018). Solana: A new architecture for a high performance blockchain v0. 8.13. *Whitepaper*.