



INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA (ISEL)

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

LEIM

LICENCIATURA EM ENGENHARIA INFORMÁTICA E MULTIMÉDIA

UNIDADE CURRICULAR DE PROJETO

BookletNFT

(eventual) imagem ilustrativa do trabalho – *dimensão*: até 13cm x 4cm

Miguel Silvestre (45101)

Pedro Henriques (45415)

Orientador(es)

Professor Hélder Bastos

Professor Doutor Paulo Trigo

Julho, 2022

Resumo

BookletNFT é um projeto que pretende criar uma aplicação web capaz de reproduzir as experiencias de infância de colecionar, trocar e completar cadernetas de cromos.

Se for bem sucedido este projeto irá ser capaz de descrever o processo de criação de uma aplicação distribuída que permita efetuar ações nos cromos tais como criar, vender, comprar e possivelmente até a troca direta de NFTs.

Neste relatório iremos abordar temas tais como Blockchains, Non Fungible Tokens e aplicações descentralizadas (*DAPP*), explicando quais as suas vantagens e desvantagens e justificando a utilização destas tecnologias em prol de outras.

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

O sistema desenvolvido é altamente seguro devido á utilização da tecnologia blockchain, no entanto pode apresentar falhas funcionais devido ao estado operacional da mesma blockchain. 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

Write here an overview of your work ...

Motivation, most relevant ideas, main contributions, evaluations and brief conclusions.

Short sentences. Succinct paragraphs. Top-down approach.

Agradecimentos

Escrever aqui eventuais agradecimentos . . .

Eventual texto de dedicatória . . .

. . . mais texto,

. . . e o fim do texto.

Índice

Resumo	i
Abstract	iii
Agradecimentos	v
Índice	ix
Lista de Tabelas	xi
Lista de Figuras	xiii
1 Introdução	1
1.1 Motivação	2
2 Trabalho Relacionado	3
2.1 Cromos Panini	3
2.2 NBA Top Shot	4
2.3 Crypto Kitties	5
2.4 OpenSea.io	6
3 Modelo Proposto	7
3.1 Requisitos	7
3.1.1 Objetivos	8
3.1.2 Público Alvo	8
3.1.3 Metas a alcançar	8
3.1.4 Atributos do sistema	9
3.1.5 Casos de utilização	10
3.2 Fundamentos	11

3.2.1	Blockchain	11
3.3	Abordagem	17
3.3.1	Processo de desenvolvimento	17
3.3.2	Arquitetura do sistema	18
3.3.3	Distribuição	19
4	Implementação do Modelo	21
4.1	Obtenção de ferramentas e configurações iniciais necessárias .	21
4.2	Desenvolvimento inicial	23
4.2.1	Funcionalidades genéricas	23
4.2.2	Expansão de funcionalidades	26
4.2.3	Distribuição	30
5	Validação e Testes	33
5.1	Mint de NFT	33
5.2	Venda de um NFT	36
5.3	Criação e abertura de packs	37
6	Conclusões e Trabalho Futuro	41
A	Detalhe Requisitos	43
B	Acrónimos	45
	Bibliografia	47

Lista de Tabelas

3.1	Comparação de algumas blockchains	14
4.1	Tabela descritiva do conteúdo de um NFT	25
4.2	Tabela descritiva do conteúdo de um NFT com packs	29
A.1	Requisitos do sistema	43

Lista de Figuras

2.1	Cromos da Panini by Inês Gomes Lourenço	3
2.2	NBA Top Shot Promo art	4
2.3	Crypto Kitties Promo art	5
2.4	OpenSea Promo art	6
3.1	Casos de utilização utilizador	10
3.2	Casos de utilização para atores com permissões específicas . . .	11
3.3	Funcionamento da Blockchain Bitcoin, como descrito em [Nakamoto, 2009] .	13
3.4	Everydays: the First 5000 Days by Beeple	15
3.5	Modelo representativo da estrutura do repositório	17
3.6	Modelo de pacotes proposto para a realização deste projeto. .	18
3.7	Docker	19
3.8	Componentes do docker segundo [Berlatto, 2020]	20
4.1	Modelo representativo da interação dos componentes	22
4.2	Diagrama <i>Swim Lane</i> do funcionamento de compra de um NFT	26
4.3	Diagrama <i>Swim Lane</i> do processo de venda de um NFT . . .	27
4.4	Diagrama <i>Swim Lane</i> do funcionamento do smart contract . .	28
5.1	Selecionar o tipo de asset que vai ser o NFT	33
5.2	Fazer upload do ficheiro	34
5.3	Definir atributos do NFT	34
5.4	Quais os <i>royalties</i> a serem pagos	34
5.5	Confirmação e pagamento do lançamento	34
5.6	Página de sucesso	35
5.7	NFT na carteira	35
5.8	Tipo de venda	36

5.9	Selecionar o NFT que se pretende vender e o seu em preço em solanas	36
5.10	Review final da venda	36
5.11	Nft a ser listado para venda	37
5.12	NFT listado para venda	37
5.13	Seleção de NFT's que poderão sair no pack	37
5.14	Seleção do NFT's que servirá como voucher	38
5.15	Ajuste de quantidades e de probabilidades	38
5.16	Review final do pack	38
5.17	Pack criado e já aberto	39

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*, CriptoMoedas, *Non Fungible Token (NFT)* (daqui em diante, chamados de NFTs) e Aplicações Distribuídas (DAPPs) (daqui em diante, DAPPs). Esta aplicação irá fornecer uma DAPP, hospedada de maneira estática, capaz de receber um ficheiro multimédia (video, audio, imagem, etc), que irão corresponder a cromos digitais. A plataforma é suportada pela blockchain que irá garantir a propriedade e segurança dos cromos e a sua autenticidade. O factor de segurança foi o principal motivo para a realização deste projeto recorrendo a Blockchains em oposição a plataformas digitais convencionais.

Um dos principais problemas foi a diferença de tecnologias utilizadas neste projeto para aquelas lecionadas durante o curso. Chegamos mesmo a dizer que a única semelhança foi a utilização de uma framework de Java Script, o React, sendo de resto em tudo diferente. Ou seja, desde a blockchain que serve de base de dados e backend para o nosso projeto, até ao front end em React tudo foi diferente, inclusive foi dado um grande foco a programação assíncrona através de promessas.

Para realizar este projeto foi necessário uma pesquisa bastante extensa, a utilização de ferramentas pouco consolidadas e por vezes com comportamento inconstante pelo que os resultados obtidos em datas muito posteriores podem não ser os aqui demonstrados.

A aplicação desenvolvida permite a criação de NFTs, a definição de atributos para cada NFT a quando da sua criação, a criação de coleções, a venda desses NFTs em packs, a coleção de NFTs e a venda desses NFTs em segunda mão. Muitas das funções descritas aqui foram criadas diretamente em cima da blockchain, as restantes resultam da composição de outras funções mas apenas ao nível da front-end.

1.1 Motivação

A grande motivação para este projeto foi o desejo de trabalhar com algo novo, algo desafiante e por métodos ainda pouco explorados. O surgimento do universo META, as menções á Web3 e o crescimento constante das CriptoMoedas e NFTs gerou um sentimento de curiosidade e de interesse na nossa parte. Pelo que quando nos foi oferecida a oportunidade de criar algo nesta área decidimos aproveitar e agarrar a oportunidade.

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.

Capítulo 2

Trabalho Relacionado

2.1 Cromos Panini



Figura 2.1: 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 saquitas e completar a coleção. Estas coleções e as interações que elas criavam foram o principal motivo para o desenvolvimento do projeto.

Assim sendo, vale a pena referir, mesmo que de um ponto de vista técnico seja irrelevante para este projeto.

2.2 NBA Top Shot



Figura 2.2: 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.

Nesta aplicação web é possível fazer coleção do que eles chamam de “*moments*” que nada mais são que vídeos de alguns momentos seletos, seja pela sua relevância, teatralidade, ou interesse geral. O utilizador pode colecionar, trocar e vender os seus “*moments*”.

Esta aplicação obteve um sucesso considerável, tendo cerca de 49 mil “*moments*” disponíveis para compra no marketplace, com mais de 35 mil já trocados, num total superior a 4,300 milhões de dólares transacionados.

Assim sendo esta aplicação serviu de inspiração principal para este projeto, sendo o ponto de referência para os casos de utilização e em conjunto com os developer diaries um ponto de inspiração para a realização do nosso projeto e em certos aspectos um manual.

2.3 Crypto Kitties



Figura 2.3: 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 introduziu-nos ao conceito de smart-contracts, que tentamos

usar no nosso projeto mas que acabou por se demonstrar desnecessário.

2.4 OpenSea.io



Figura 2.4: OpenSea Promo art

OpenSea [Ozone Networks, 2022a] é o mais antigo e o maior mercado de NFTs existente a quando 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 Requisitos

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. Assim sendo os requisitos devem ser o mais detalhados possível ao mesmo tempo que mantendo um nível grande abstração. Assim sendo iremos abordar neste capítulo os seguintes tópicos

- Os objetivos do projeto.
- Público Alvo
- Metas
- Casos de utilização.
- Atributos e funções do sistema
- Concretização dos casos de utilização com base nos pontos anteriores.
- Agenda dos objetivos

3.1.1 Objetivos

Neste projeto será desenvolvida 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, etc....

Pretende-se criar um ambiente capaz de simular a compra, venda, abertura de packs e coleção destes cromos, gerando emoções como intriga, expectativa, frustrações, etc....

Este projeto é um caso de uso excelente para a utilização de NFTs e Blockchains, pelo que um objetivo será também a pesquisa, e obtenção de conhecimento sobre estas tecnologias emergentes.

3.1.2 Pùblico Alvo

O nosso pùblico alvo poderá variar, consoante a quem interesse a coleção de determinado tema. Segundo estritamente por estereótipo, um tópico que interessa mais a meninas serão princesas e a meninos monstros, ou a pessoas mais velhas, as mulheres possivelmente interessam-se por personagens de novelas e os homens por futebol.

Assim sendo definimos o nosso principal pùblico alvo, como pessoas de todas as idades, com algum rendimento disponível, interessadas em determinado tema.

Alem dessas pessoas poderíamos também vender cromos e packs a traders profissionais que apostariam nas tendências do mercado dos cromos para gerar lucros para si mesmo, e finalmente dum ponto de vista professional, empresas que pagariam por ações publicitárias lançando as suas próprias cadernetas e cromos.

3.1.3 Metas a alcançar

- Criar NFTs diretamente na nossa 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)

Alem destas metas existem muitos outros requisitos de sistema a ter em consideração, tais como requisitos de segurança e funcionalidades menores. De modo a não obter uma lista demasiado extensa o leitor poder verificar o apêndice A.1 com a lista completa dos requisitos do sistema.

3.1.4 Atributos do sistema

De modo a realçar e melhorar o sistema ele deve possuir alguns atributos, alguns obrigatórios, outros opcionais, de qualquer forma estes atributos foram tidos em consideração a quando da realização do relatório.

Atributo	Detalhe / Restrição Fronteira	Categoría
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

3.1.5 Casos de utilização

Os casos de utilização surgem em sequencia das metas e requisitos de sistema, tendo em conta os atributos definidos anteriormente. Assim definimos dois tipos de atores, o criador e o comprador ambos estes utilizadores.

Em primeiro lugar surgem os casos de utilização de todos os utilizadores. Nestes casos de utilização podemos ver que qualquer utilizador terá capacidade de conectar uma carteira, e depois da carteira estar conectada, ver o conteúdo da sua carteira (independentemente se comprou ou criou os NFTs) e vender os conteúdos presentes na sua carteira.

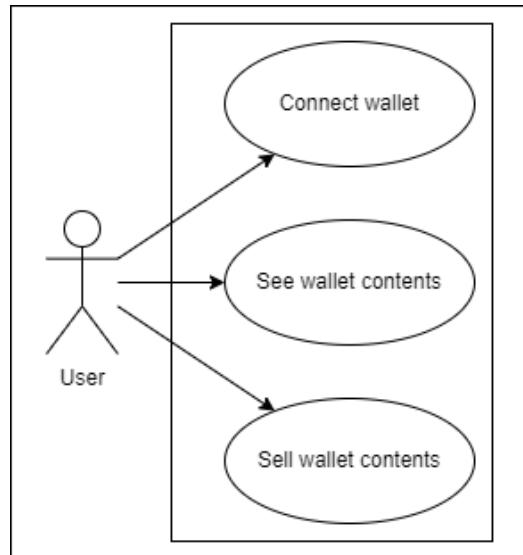


Figura 3.1: Casos de utilização utilizador

Os casos de utilização específicos para o comprador, são aqueles presentes anteriormente mais a opção de comprar. Os casos de utilização específicos para o criador, são aqueles disponíveis para o utilizador, mais a opção de criar.

Finalmente vale a pena notar que um mesmo utilizador pode ser um criador e um comprador em simultâneo, apesar de poder ser somente um desses papéis.

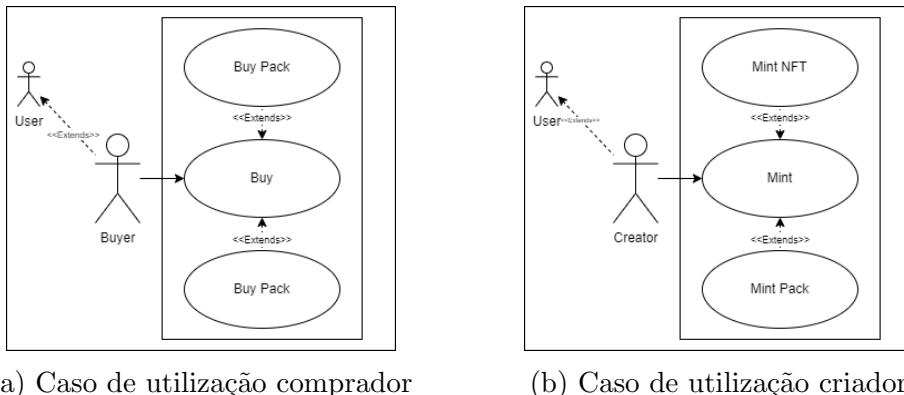


Figura 3.2: Casos de utilização para atores com permissões específicas

3.2 Fundamentos

Nesta secção iremos introduzir a teoria dos tópicos utilizados para realizar o projeto.

3.2.1 Blockchain

Como dito anteriormente este projeto será realizado em cima de uma blockchain. Nesta secção iremos explicar o funcionamento da blockchain de um ponto de vista geral e quais as suas vantagens.

Vantagens

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

- **Segurança:** As blockchains são inherentemente seguras a nível de transações e validações devido à sua natureza descentralizada, as provas de trabalho realizadas por todos os intervenientes na Blockchain e o consenso entre estes intervenientes.

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, portanto apesar de não ser impossível atacar, é certamente mais difícil que outros sistemas informáticos.

Este motivo foi o fator decisivo de utilização deste sistema em prol de outros.

- **Transparéncia:** Todos os intervenientes na blockchain possuem acesso a uma única base de dados, e nenhum destes é dono da verdade. Assim sendo todas as transações são registadas, validadas e acedidas a partir de uma única base de dados universal, garantindo que não são realizadas escritas indevidas desta
- **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. No exemplo do nosso projeto o 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 e outros problemas associados.

Ora estas vantagens resolvem por nós todos os atributos de sistema relativos a segurança de dados e de transações sem qualquer custo computacional para o nosso sistema.

Funcionamento de uma blockchain

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

Resumidamente uma blockchain é uma base de dados que apenas permite leitura e escrita. Cada transação ou instrução é armazenada num bloco e ao criar um novo bloco com outra transação ou instrução é colocada uma referência ao bloco anterior. Estes blocos formam assim uma cadeia de blocos ou blockchain, verificável e imutável. Comportamento esquematizado na

figura 3.3, ainda que apenas mencione transações o funcionamento para dados e instruções é idêntico.

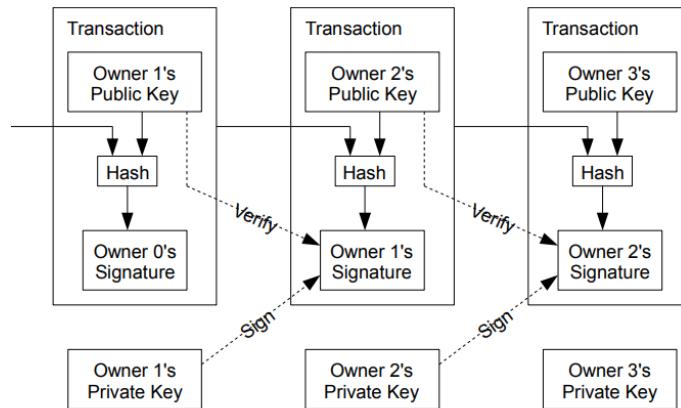


Figura 3.3: Funcionamento da Blockchain Bitcoin, como descrito em [Nakamoto, 2009]

Com um detalhe maior, podemos ver uma blockchain 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. A cada transação/instrução é colocado um novo bloco de informação na base de dados. Estes blocos correspondem a qualquer pedaço de informação e nos casos dos NFTs, contém dados que correspondem ao registo de propriedade de outros pedaços de informação digital. A cada bloco adicionado é colocado uma referência ao hash do bloco anterior e é calculado o seu hash que posteriormente será usado no bloco seguinte. Estas referências nunca poderão ser alteradas mantendo sempre a mesma informação. Caso exista algum erro como preço ou itens incorretos, essa informação não pode ser alterada sendo necessário a criação de um novo bloco que indique a correção a ser realizada. 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 no mercado, desde aquelas já mencionadas em cima Bitcoin e IBM Blockchain, passando por Ethereum, Solana, Hyperledger Fabric, Ripple, Terra etc..., na tabela 3.1, podemos ver aquelas exploradas por nós e quais as vantagens e desvantagens de cada uma. Através desta Comparaçāo decidimos que devido aos baixos custos por

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 3.1: Comparação de algumas blockchains

transação e á existênciade diversas ferramentas a blockchain mais vantajosa para o nosso trabalho seria a **Solana**.

Ethereum

A Ethereum foi a primeira blockchain, a obter sucesso, capaz de suportar mais do que a transação de moedas. Com a Ethereum surgiram conceitos tais como “*smart-contracts*” e “*Non Fungible Tokens*”, essenciais á realização deste projeto. A Ethereum possui a sua própria moeda o **Ether**.

O Ethereum tem uma grande comunidade em seu redor com diversos projetos desenvolvidos tais como o CryptoKitties e o OpenSea mencionados em cima.

NFTs

Tal como mencionado em cima, a grande inovação desta blockchain foi a possibilidade de se incorporar nos seus blocos, informação adicional, informação esta que levaria á criação de NFTs. Desta forma podemos armazenar representações de objetos ou referencias a estes num bloco, armazenando na blockchain esta referencias e garantindo a sua propriedade. Ora temhamos como exemplo o NFT, único, vendido pelo maior valor de sempre. “*3.4 Everydays: the First 5000 Days by Beeple*”

Como podemos ver na figura 3.4 á direita, este NFT representa nada mais nada menos que uma imagem ou obra de arte. Esta peça encontra-se armazenada no IPFS e é imutável no seu interior. A composição do NFT é um ficheiro JSON com um atributo *imageUrl* que indica onde está armazenada esta imagem. Toda a informação sobre este NFT pode ser acedida através do link na seguinte citação [Beeple, 2021].

De um ponto de vista de sistemas informáticos, este NFT é um ficheiro JSON armazenado numa blockchain com atributos tais como *name*, *type*, *imageUrl*, *description*, etc. A parte relativa á propriedade é verificada através do rastreio das transações desde o momento em que foi criado o token, vulgarmente conhecido por “*airdrop*”, até á ultima transacção onde está presente o endereço da carteira que possui o token.

O importante a lembrar é que um NFT é nada mais que um bloco de informação que representa um objeto. Além disso o dono desta representação é também o dono do objeto em si, tal como se se tratasse do livrete de um carro ou registo de propriedade de um imóvel.

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 ás *gas fees* (custo por transacção).

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 CriptoMoedas tais como Eth, Sol, etc....

Devido ao Ripple ser focada em transações, e não ter suporte explícito a NFTs, esta blockchain não é adequada á realização deste projeto.

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

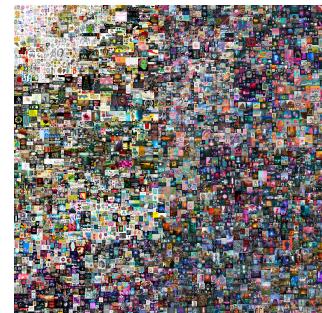


Figura 3.4: Everydays:
the First 5000 Days by
Beeple

da mesma forma que outras moedas fiduciárias, através da emissão e compra de moeda por uma entidade central. A

Em Maio de 2022 a moeda **Luna** não conseguiu acompanhar a inflação do dólar e perdeu a ligação com esta. Assim sendo perdeu-se a confiança na moeda e na blockchain pelo que o seu valor atingiu o 0.

Na altura da escolha, não sabíamos que iria acontecer a queda abrupta no valor da moeda, no entanto o facto dos NFTs nesta blockchain estarem apenas num estado ”conceptual” e não existirem frameworks para realizar o projeto acabamos por decidir contra a utilização desta.

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 primeiramente 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 em redes sociais, notavelmente o Discord, permite o esclarecimento de duvidas e troca de ideias, ajudando a realização do projeto.

Assim sendo, os baixos custos de transação, as ferramentas disponíveis e a comunidade em torno levam a que esta seja a Blockchain escolhida por nós para desenvolver este projeto.

Metaplex

Segundo os autores:

“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 Websites e coleções de NFTs.

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

Assim sendo esta ferramenta parece ser bastante útil para a criação de aplicações web que trabalhem com a Solana.

3.3 Abordagem

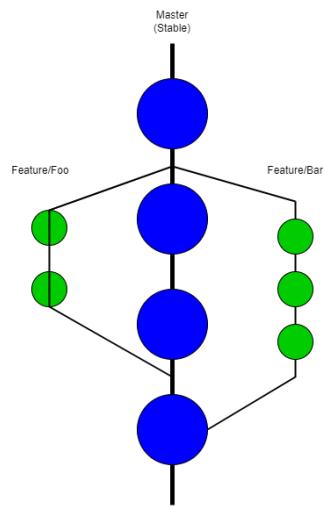
3.3.1 Processo de desenvolvimento

Controlo de versões

Em projetos realizados noutras unidades curriculares do curso enviar emails com as alterações realizadas não se demonstra um grande entrave. A quantidade de ficheiros é pequena, as alterações são curtas e é fácil de as enumerar. No entanto com o progredir da licenciatura, depressa este método de partilha de conteúdos se tornou demasiado moroso e propenso a falhas. Assim sendo ao longo do curso, e neste mesmo projeto, foi adotada a utilização de software de controlo de versões, no nosso caso o **GIT**.

Neste projeto usamos o git, como sistema de controlo de versões, e o Github como repositório. A nível de *branches* 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 *branches* sempre no esquema de *feature/<<feature-name>>*, onde feature-name corresponde ao nome da questão a ser trabalhada. Naquele momento. A representação do nosso controlo de versões pode ser representada na figura 3.5.

Figura 3.5: Modelo representativo da estrutura do repositório



Por vezes o retorno ou *merge* com o master não gerava conflito como é o caso do branch feature/Foo, outras vezes existia um *merge conflict* que era resolvido caso a caso, como no branch feature/Bar.

3.3.2 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 3.6

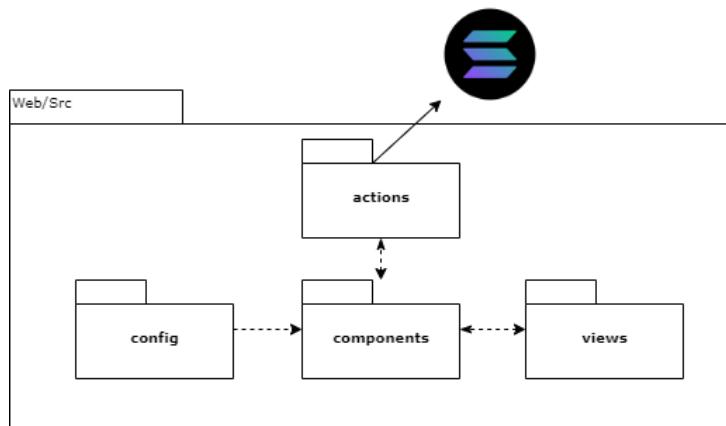


Figura 3.6: Modelo de pacotes proposto para a realização deste projeto.

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 irá possuir todas as configurações que não devem ser incluídas no código, como endereços de carteiras, repositórios, e urls onde o programa será hospedado, etc...
2. O 2º package **components** irá corresponder á lógica da nossa aplicação. Neste package será contido toda a parte relativa ao processamento das informações introduzidas pelo utilizador.

3. O 3º package **actions** irá corresponder á comunicação com a blockchain. Neste package são efetuadas as chamadas á 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 não foi desenvolvido por nós, tendo sido adotado devido á utilização de uma framework. Através da analise deste obtivemos o diagrama em cima na figura 3.6.

3.3.3 Distribuição

Docker

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, originando um medo que os próprios arguentes sejam incapazes de executar o projeto levando a uma avaliação injustas. De modo a resolver esta questão introduzimos o **Docker**.

O Docker elimina a desculpa do “*não sei porque o programa não funciona na tua, mas ele está a funcionar na minha máquina*” de uma maneira bastante simples. 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 3.8:

1. O repositório ou “*registry*”. É o local na cloud onde as imagens das aplicações são armazenadas. Assim sendo o nosso projeto na realidade

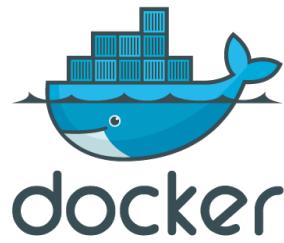


Figura 3.7: Docker

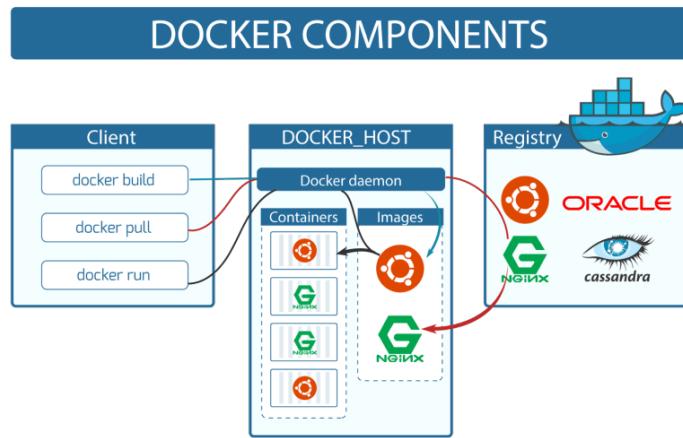


Figura 3.8: Componentes do docker segundo [Berlatto, 2020]

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, respectivamente “*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. Um exemplo disto é no nosso projeto a dependências á imagem do Node.

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**.

O Node JS serve como plataforma de execução do resto do projeto pelo que foi instalado primeiro sem grandes conflitos.

De seguida a instalação do yarn mostrou-se problemática, em primeiro lugar a instalação “simples” usando o gestor de pacotes do Node JS, o NPM, instala uma versão não suportada pelos outros componentes do projeto. Pelo que obrigou a realizar debug e a questionar em grupos online para o efeito o que se estaria a passar. Resolvido este problema e instalada a versão correta, a versão antiga não foi eliminada e aliás, continuava a ser tida como a versão utilizada, levando a confusão e frustração na nossa parte. Finalmente resolvemos essa questão eliminando das nossas máquinas a versão não compatível.

Noutro ponto a instalação de Type Script. Ora esta mostrou-se a questão mais problemática de todas. A versão mais recente do Type Script ou seja aquela disponibilizada para o sistema operativo que executamos não é de todo compatível com os outros componentes do projeto. Assim sendo pensa-

mos em executar e instalar através do código fonte. Tal não nos foi possível realizar pois ia ser necessário a instalação de ainda mais componentes e compiladores externos.

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, facilitando muito a execução da maioria dos comandos presentes na documentação e tutoriais online.

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 úteis 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.

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 *deprecated* 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, mas sem valor monetário real 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 á criação de uma loja online. O repositório contem diversos componentes escritos em React que fornecem algumas funcionalidades genéricas.

4.2.1 Funcionalidades genéricas

Conexão de carteiras

A conexão de carteiras não foi implementada por nós. E os próprios componentes em questão não foram modificados além de lhes terem sido aplicados

estilos. Assim sendo apenas iremos explicar o funcionamento desta função tendo em consideração que a sua grande maioria não foi de nossa autoria.

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 publico desta numa variável de sistema no browser *localStorage*. 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 veio inclusa com a primeira instalação do projeto. Ao clonar o projeto e após realizar as configurações necessárias como definir uma carteira e um endereço para a loja somos capazes de executar a criação de NFTs diretamente na própria blockchain. O NFT criado tem diversos atributos que podem ser visualizados na tabela 4.1

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. Existe diversos atributos, sendo que alguns deles são compostos por outros objetos JSON.

Entre todos os atributos , de longe os mais importantes de todos é o Data ou o URI onde está armazenada o conteúdo binário correspondente ao NFT. A utilização do data com os dados em bruto implica a transmissão de todos os dados do NFT, levando a *gas fees* maiores. A utilização do uri implica a perda de alguma autonomia do utilizador em relação ao cromo que este comprou. 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.

A utilização

Compra e Venda de NFTs

A venda de NFT's é outra funcionalidade que veio incluída com a instalação do projeto. Existem dois tipos de venda de NFT's nomeadamente a venda

Nome do atributo	Formato	Obrigatório	Descrição
Creators	Array	Yes	Array de objetos que identifica os criadores do token
Name	String	No	
Symbol	String	No	String representativa do token
isItParent-Collection	Boolean	No	Se o token for capa de uma coleção este atributo é tido como verdadeiro
Collection		No	Identifica qual a coleção em que o NFT se inclui.
SellerFee	Float $\in [0, 1]$	No	Valor pago em percentagem ao criador do NFT, durante cada transação.
Description	String	No	
Data	Object	No	Conteúdo do NFT. Obrigatório este atributo ter valor, se o uri for vazio.
uri	Binary String	No	Url com a localização do conteúdo binário do NFT. Obrigatório este atributo ter valor, se o Data for vazio

Tabela 4.1: Tabela descritiva do conteúdo de um NFT

direta e o leilão. Neste projeto só trabalhamos com a venda direta uma vez que se achou que leilões não se enquadrava com a finalidade do projeto.

A venda de um NFT, no contexto da nossa aplicação apenas considera a venda através de *marketplace*, assim sendo, para colocar um NFT à venda foi criado um vault onde são listados os NFT's 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.

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

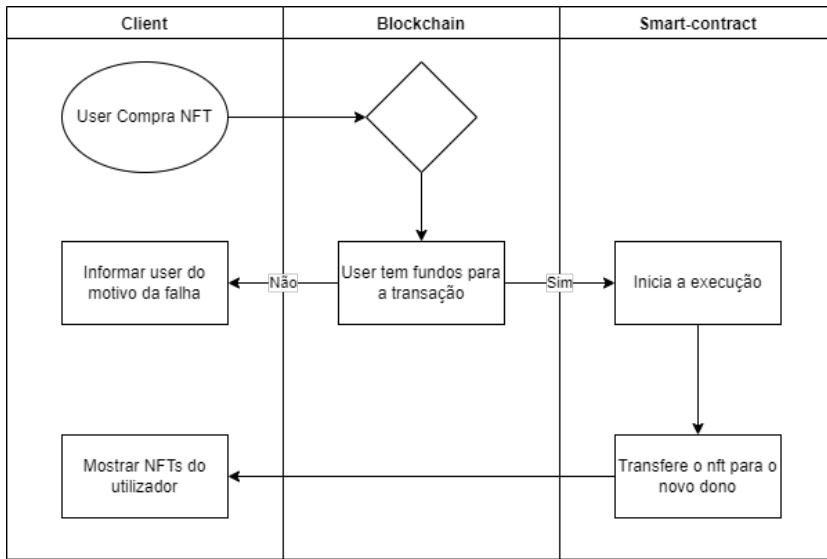


Figura 4.2: Diagrama *Swim Lane* do funcionamento de compra de um NFT

dos NFTs em questão. A sua execução implica que o token seja armazenado no vault e a quando da compra, transfere os fundos recebidos para o vendedor.

4.2.2 Expansão de funcionalidades

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 foi necessária a expansão das funcionalidades fornecidas pela framework Metaplex. Assim sendo, vamos descrever o processo através do qual implementamos esta funcionalidade.

Começando por algo que não obteve sucesso, 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 à 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 4.4

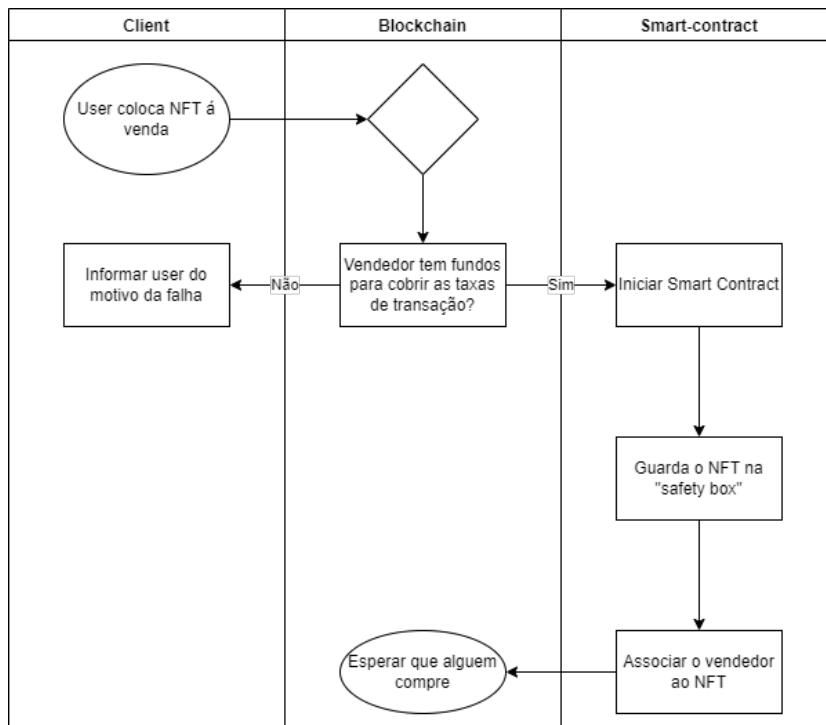


Figura 4.3: Diagrama *Swim Lane* do processo de venda de um NFT

Algumas notas sobre o diagrama. Em 1º lugar temos que a blockchain verifica as condições do utilizador antes de iniciar o smart-contract. Tal não é um erro, pois o smart contract é instanciado pela propria blockchain e não diretamente pelo utilizador. Da mesma forma que a atribuição dos NFT, apesar de indicar que é ao utilizador, não o é. É atribuído a um endereço de uma carteira dentro da blockchain. Essa carteira por sua vez é propriedade do utilizador. Este utilizador por transitividade é dono dos NFTs presente na sua carteira. No entanto, de modo a obter um diagrama rigoroso, não colocamos a transição diretamente para o utilizador.

Outra consideração a ter em conta é a destruição do NFT ou token do pack. Ora como dito previamente tal é impossível pois a blockchain não permite modificar ou eliminar dados. No entanto o que permite é a transferência sem qualquer tipo de verificação para qualquer carteira. Assim sendo na realidade, 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 po-

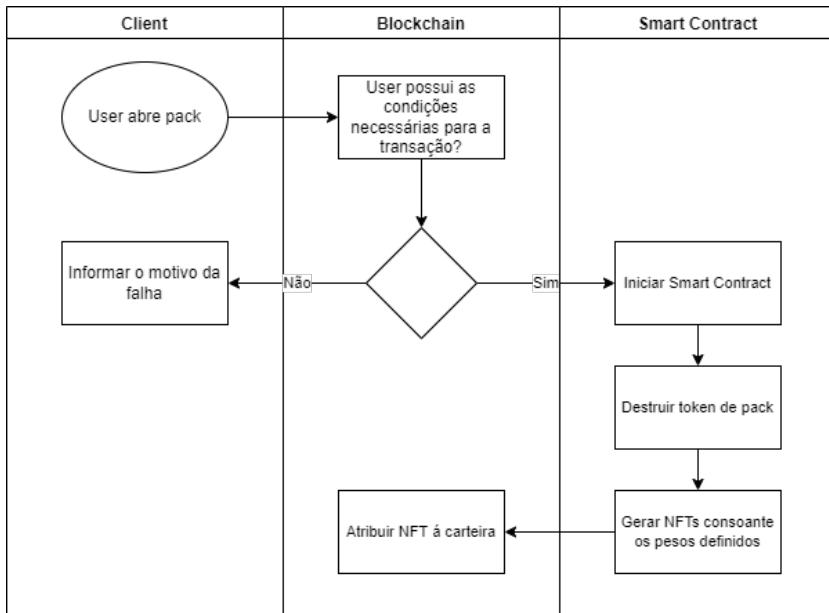


Figura 4.4: Diagrama *Swim Lane* do funcionamento do smart contract

derão 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 $\in [A, B, C, D, E]$, sendo que cada elemento é equiprovável na sua distribuição. A escolha de 3 elementos além de apresentar a possibilidade de levar a elementos repetidos tem uma falha crítica, a distribuição de 3 elementos corresponde a $3+N$ 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 como que pendurada, possivelmente até que o utilizador introduza mais fundos na carteira.

Devido a este ultimo problema acabamos por riscar esta abordagem acabando por utilizar uma implementação da própria Metaplex, que curiosamente sofre exatamente 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, 2022b].

Implementação Packs

Como já mencionado anteriormente, a implementação dos packs recorreu a uma implementação presente no próprio repositório. A parte do código relativa aos packs não se encontra ativa por defeito. Vamos agora explicar o processo de desenvolvimento através do qual ativamos esta funcionalidade.

Em 1º lugar gostaríamos de mencionar que o funcionamento desta *feature* se deve em grande parte à comunidade de apoio em grupos online. Assim sendo fomos capazes de realizar um desenvolvimento muito mais rápido do que seria expectável, ainda que possivelmente sacrificando alguma da pesquisa e obtenção de conhecimentos que viriam com isso.

Assim sendo o processo de desenvolvimento iniciou-se com a criação do “path” que iria mostrar os packs. Assim sendo 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 enable, usamos esqueleto de pagina web já desenvolvida para o efeito, 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 que contem os atributos presentes na tabela 4.2

Nome do atributo	Formato dos dados	Obrigatório?
------------------	-------------------	--------------

Tabela 4.2: Tabela descritiva do conteúdo de um NFT com packs

Assim sendo um pack corresponde a um NFT que pode ser redimido pelo

utilizador que o possua, sendo contido no seu interior todas as informações necessárias á geração de novos NFTs ou atribuição de NFTs já criados presentes num *vault*.

4.2.3 Distribuição

O processo de preparação do projeto para distribuição tem como principal objetivo tornar o mais fácil possível o armazenamento e distribuição de uma versão compilada do código. Assim sendo optamos por usar um modelo de distribuição em “*containers*”, tal como já referido no capítulo 3.3.3, usando o Docker como ferramenta de distribuição.

Ambiente de produção

Este projeto sofreu alguns contratempas na sua confeção que levou á criação de pastas, ficheiros e programas de uso temporário. De modo a não enviar para a avaliação estes elementos com valor nulo, decidimos proceder á criação de um ambiente destinado á produção onde apenas incluímos o produto estável e tudo o que é necessário ao seu funcionamento.

Assim sendo criamos uma diretoria nova no disco, onde incluímos apenas os ficheiros necessários e executamos o programa em modo de produção.

Uma alternativa a esta abordagem seria a utilização do git, criando um branch para este ambiente ou até reservando o master para o efeito. Acabamos por não optar por esta abordagem devido á necessidade de incluir outros ficheiros como o relatório e seu ficheiros auxiliares na entrega seguindo uma estrutura de diretórias rígida que não tínhamos implementado até então.

Criação da imagem e sua execução

Não nos iremos alongar muito na criação da imagem com a aplicação pois o cerne da questão é o Dockerfile, que já vinha incluído com o repositório sendo apenas necessária a mudança de algumas linhas referentes a configurações específicas, nomeadamente o endereço de carteira do proprietário da loja.

Ainda assim vale a pena mencionar que o Docker, executa um conjunto de passos de modo a criar um ambiente de execução fiável e reproduzível, independente da máquina em que é executado.

Lendo o ficheiro Dockerfile podemos verificar o ambiente de execução, nomeadamente uma imagem de Node JS, a instalação de dependências necessárias á execução do projeto no interior do container, a definição de variáveis de ambiente e a abertura da porta necessária á comunicação com o exterior. Vale notar que esta porta é apenas para uso no interior do *container*, sendo mapeada pelo docker para outra qualquer porta no *host*.

A execução do projeto consiste então em instalar o docker, fazer “*pull*” da imagem do repositório, e executa o container. Acedendo então ao projeto através de qualquer browser através de *localhost* na porta definida pelo utilizador aquando da execução da imagem.

Nota: A execução deste projeto implica a instalação e execução do Docker. Se o SO do utilizador for o Windows, o docker tem um peso computacional significativo pois necessita de executar uma VM Linux para a execução das imagens. Ainda assim consideramos que a vantagem da facilidade da distribuição e a possibilidade de ser replicado vantagens que não devem ser ignoradas.

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 a sua utilização.

5.1 Mint de NFT

Começamos primeiramente por testar o mint de um NFT as imagens que se seguem mostram os passos necessários para fazer mint de um NFT.

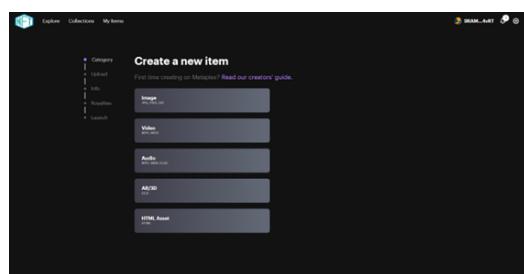


Figura 5.1: Selecionar o tipo de asset que vai ser o NFT

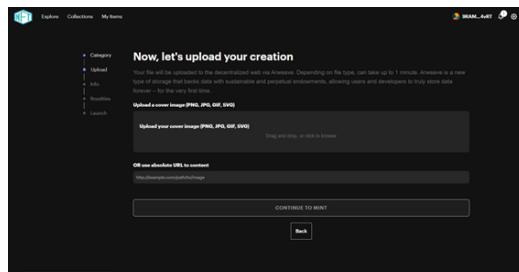


Figura 5.2: Fazer upload do ficheiro

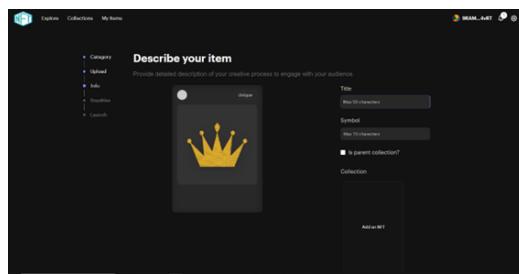


Figura 5.3: Definir atributos do NFT

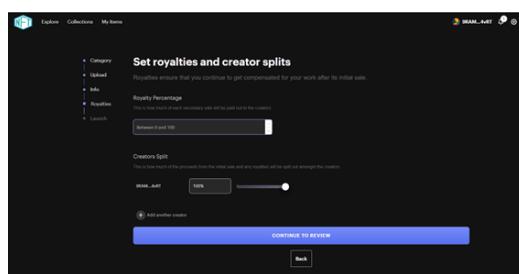
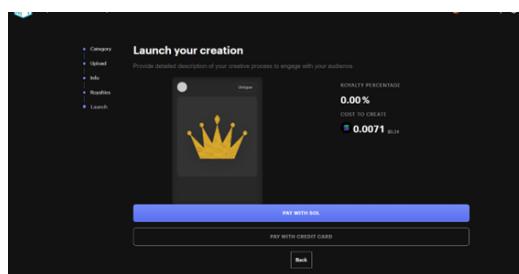
Figura 5.4: Quais os *royalties* a serem pagos

Figura 5.5: Confirmação e pagamento do lançamento

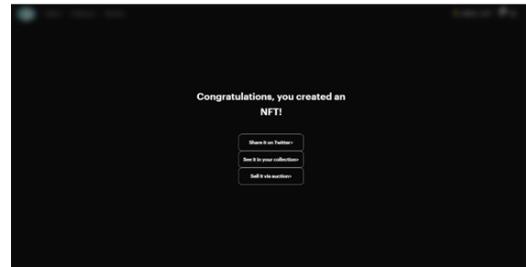


Figura 5.6: Página de sucesso

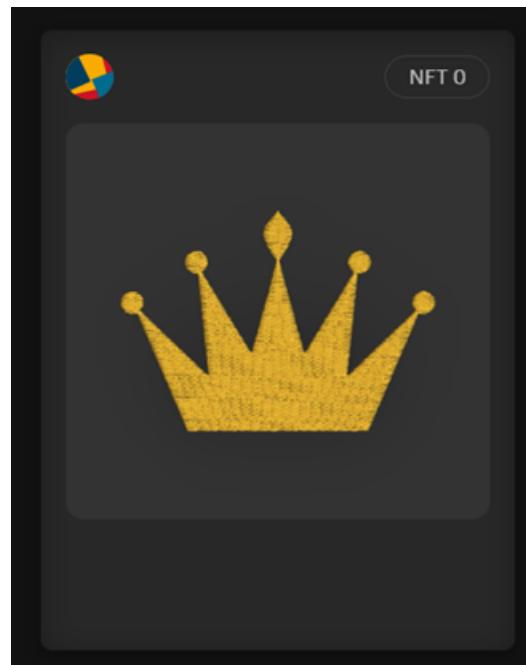


Figura 5.7: NFT na carteira

5.2 Venda de um NFT

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

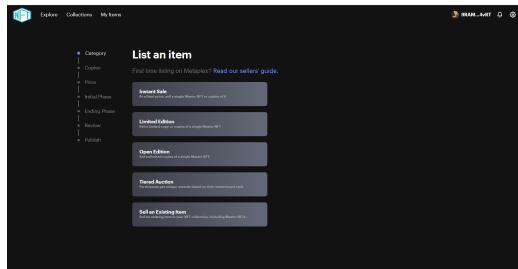


Figura 5.8: Tipo de venda

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.

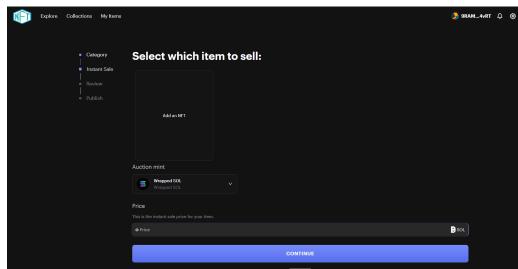


Figura 5.9: Selecionar o NFT que se pretende vender e o seu em preço em solanas

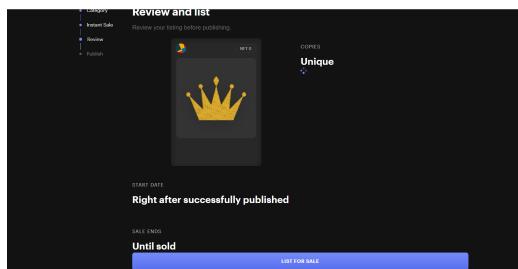


Figura 5.10: Review final da venda

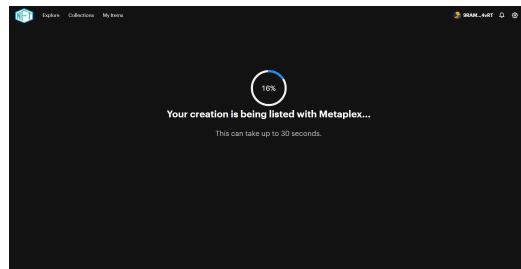


Figura 5.11: Nft a ser listado para venda

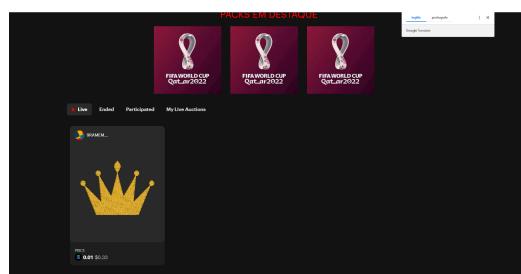


Figura 5.12: NFT listado para venda

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.

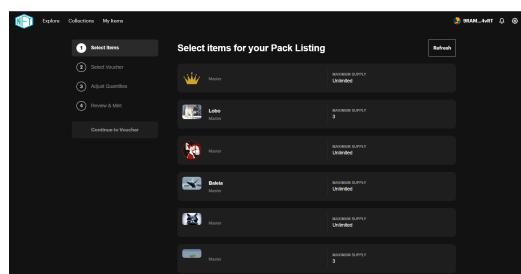


Figura 5.13: Seleção de NFT's que poderão sair no pack

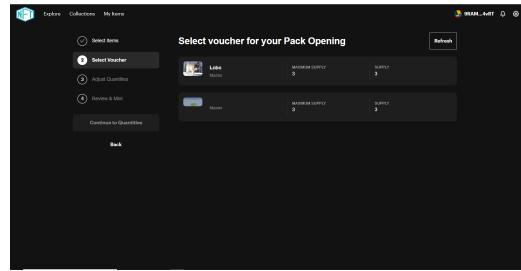


Figura 5.14: Seleção do NFT's que servirá como voucher

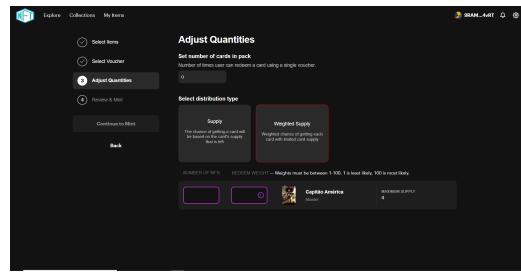


Figura 5.15: Ajuste de quantidades e de probabilidades

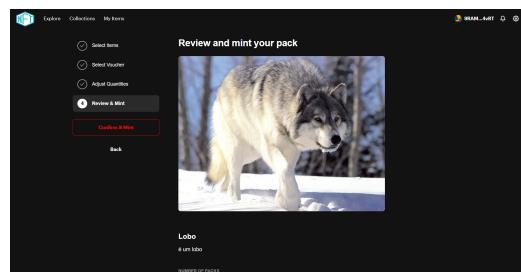


Figura 5.16: Review final do pack

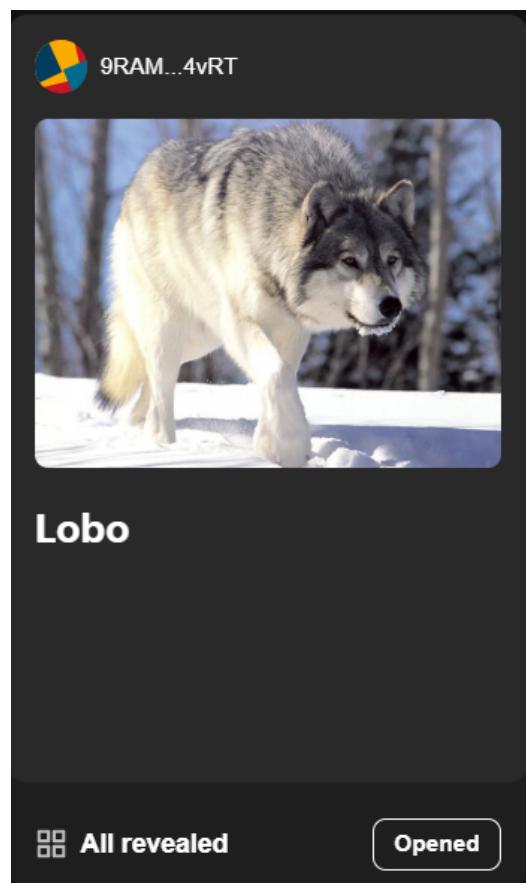


Figura 5.17: Pack criado e já aberto

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 para todos os objetivos propostos.

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 grande 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.

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 block-

chain, cripto moedas e nfts além de conhecimento obtido nesta área e de sistemas informáticos no seu geral.

Apêndice A

Detalhe Requisitos

Este apêndice pretende demonstrar com maior detalhe quais os requisitos do sistema anteriormente no relatório.

Tabela A.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	

Continua na próxima página

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	
R 3.3	Permitir a troca direta de cromos	Visível	Consultas
R 3.4	Permitir a troca de cadernetas incluindo todos os cromos contidos	Adorno	

Apêndice B

Acrónimos

NFT Non Fungible Token

DAPP Aplicação Distribuída

Bibliografia

- [Beeple, 2021] Beeple (2021). Everydays: The first 5000 days.
- [Berlatto, 2020] Berlatto, L. (2020). Primeiros passos com docker: Conceitos básicos à criação de 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>.
- [Metaplex-Foundation, 2022a] Metaplex-Foundation (2022a). Candy machine js cli - introduction.
- [Metaplex-Foundation, 2022b] Metaplex-Foundation (2022b). Metaplex nft packs solana program.
- [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*.