Instituto Superior
de Contabilidade
e Administração
Politécnico de Coimbra

Aldair Conceição Francisco Chaves

Segurança de dados em aplicações baseadas em Blockchain

Coimbra, novembro de 2022

Segurança de dados em aplicações baseadas em Blockchain
Aldair Conceição Francisco Chaves
ISCAC | 2022

Instituto Superior de Contabilidade e Administração
Politécnico de Coimbra
COIMBRA BUSINESS SCHOOL
ISCAC.pt

Aldair Conceição Francisco Chaves

Segurança de dados em aplicações baseadas em Blockchain

Trabalho de projeto submetido ao Instituto Superior de Contabilidade e Administração de Coimbra para cumprimento dos requisitos necessários à obtenção do grau de **Mestre em Sistemas de Informação de Gestão**, realizado sob a orientação do Professor Doutor António Trigo e do Dr. Francisco Lavrador Pires.

Coimbra, novembro de 2022

TERMO DE RESPONSABILIDADE

Declaro ser o autor deste projeto, que constitui um trabalho original e inédito, que nunca foi submetido a outra Instituição de ensino superior para obtenção de um grau académico ou outra habilitação. Atesto ainda que todas as citações estão devidamente identificadas e que tenho consciência de que o plágio constitui uma grave falta de ética, que poderá resultar na anulação do presente projeto.

Coimbra Business School | ISCAC
Quinta Agrícola – Bencanta, 3045-601 Coimbra
Tel +351 239 802 000; E-mail: secretariado@iscac.pt; www.iscac.pt

AGRADECIMENTOS

A entrega de um projeto que decorreu durante aproximadamente um ano, representa diversas batalhas pessoais e acadêmicas. As várias fases passam por descobertas de novos conhecimentos, desafios, cansaço, frustrações, falta de tempo e persistência, mas a representatividade maior da entrega deste trabalho é o significado de dever cumprido e de vitória. Não se pode mensurar as dificuldades encontradas pelo caminho, mas é possível sim mensurar a satisfação de se chegar ao fim do que foi proposto, com muito trabalho e sacrifícios.

Meu primeiro agradecimento vai para minha família pelo suporte de sempre e por representarem uma base para que as coisas pudessem acontecer. Em segundo lugar aos meus amigos mais próximos, que sempre têm de aturar todas as minhas inconstâncias, decisões controversas, ideias mirabolantes e ainda assim me apoiam nas decisões e momentos mais complicados.

E por último gostaria de agradecer aos meus orientadores, António Trigo e Francisco Pires, que compraram a minha ideia e não me deixaram sair dos objetivos propostos durante este longo processo.

Este momento não é só simbólico, pois recompensa os esforços e reafirma a coerência nas minhas convicções.

Coimbra Business School | ISCAC
Quinta Agrícola – Bencanta, 3045-601 Coimbra
Tel +351 239 802 000; E-mail: secretariado@iscac.pt; www.iscac.pt

RESUMO

Ataques cibernéticos têm se tornado cada vez mais frequentes no cotidiano de empresas e as técnicas utilizadas estão cada vez mais bem elaboradas. Ataques como *phishing* têm como objetivo roubar informações sensíveis e causam prejuízos financeiros cada vez mais frequentes e difíceis de combater, do ponto de vista legal e do ponto de vista corporativo. Empresas que utilizam documentos digitais para representar ou validar operações financeiras, encontram dificuldades para manter o nível de confiabilidade e segurança de tais documentos. Atualmente as opções para contornar este tipo de situação e ter um ambiente mais seguro, envolvem soluções ligadas à certificação digital ou ambientes centralizados para garantir a autenticidade e segurança de documentos.

Diante do tema segurança e da segurança dos dados, o presente trabalho tem como objetivo o desenvolvimento de uma aplicação para armazenamento de documentos em Blockchain. Com a crescente utilização da tecnologia Blockchain, suas características como descentralização e imutabilidade, trazem os atributos necessários para um ambiente mais seguro. Por outro lado, a popularização e facilidade de desenvolvimento de aplicações descentralizadas alertam para um outro lado da segurança. A falta de padronização no desenvolvimento podem levar a exploração de brechas de segurança que já existem em ambientes Blockchain ou em brechas criadas durante o processo de desenvolvimento. Para reduzir esses riscos, o desenvolvimento do protótipo deste projeto será acompanhado com a utilização de boas práticas e análise utilizando ferramentas específicas para exemplificar procedimentos que podem mitigar brechas de segurança.

Como resultado deste projeto apresenta-se a aplicação descentralizada desenvolvida (https://github.com/aldairchaves/enviadocs_v4.git) e o processo de desenvolvimento que constitui uma primeira abordagem para um processo de desenvolvimento de aplicações descentralizadas, com o foco na segurança dos dados do ponto de vista da solução e na construção da aplicação.

Palavras-chave: Blockchain; Aplicações descentralizadas; DApp; Segurança de dados; Segurança em aplicações descentralizadas; Processo de desenvolvimento de software.

ABSTRACT

Cyber-attacks have become more and more frequent in the daily life of companies and the techniques used are becoming more and more elaborate. Attacks such as phishing aim to steal sensitive information and cause increasingly frequent and difficult to combat financial losses, both from a legal and corporate point of view. Companies that use digital documents to represent or validate financial transactions find it difficult to maintain the level of reliability and security of such documents. Today the options to get around this type of situation and have a more secure environment involve solutions linked to digital certification or centralized environments to ensure the authenticity and security of documents.

Facing the security theme and data security, the present work aims to develop an application for storing documents in Blockchain. With the increasing use of Blockchain technology, its characteristics such as decentralization and immutability, bring the necessary attributes for a more secure environment. On the other hand, the popularization and ease of development of decentralized applications warns of another side to security. Lack of standardization in development can lead to exploitation of security holes that already exist in Blockchain environments or security breaches created during the development process. To reduce these risks, the development of the prototype of this project will be followed with the use of best practices and analysis using specific tools to exemplify procedures that can mitigate security breaches.

As outcomes of this project, the decentralized application developed (https://github.com/aldairchaves/enviadocs_v4.git) and the development process that constitutes a first approach for a decentralized application development process is presented, focusing on data security from the solution point of view and the application construction.

Keywords: Blockchain; Decentralized applications; DApp; Data security; Security in decentralized applications; Software development process

ÍNDICE GERAL

Coimbra Business School | ISCAC
Quinta Agrícola – Bencanta, 3045-601 Coimbra
Tel +351 239 802 000; E-mail: secretariado@iscac.pt; www.iscac.pt

Coimbra Business School | ISCAC
Quinta Agrícola – Bencanta, 3045-601 Coimbra
Tel +351 239 802 000; E-mail: secretariado@iscac.pt; www.iscac.pt

ÍNDICE DE FIGURAS

Coimbra Business School | ISCAC
Quinta Agrícola – Bencanta, 3045-601 Coimbra
Tel +351 239 802 000; E-mail: secretariado@iscac.pt; www.iscac.pt

Coimbra Business School | ISCAC
Quinta Agrícola – Bencanta, 3045-601 Coimbra
Tel +351 239 802 000; E-mail: secretariado@iscac.pt; www.iscac.pt

Coimbra Business School | ISCAC
Quinta Agrícola – Bencanta, 3045-601 Coimbra
Tel +351 239 802 000; E-mail: secretariado@iscac.pt; www.iscac.pt

LISTA DE ABREVIATURAS, ACRÓNIMOS E SIGLAS

DApp – *Decentralized application*

DDOS – *Distributed denial-of-service*

DoS – *Denial of Service*

DPoS – *Delegated Proof of Stake*

DSR – *Design Science Research*

ERC – *Ethereum Request for Comment*

GDPR – *General Data Protection Regulation*

HTTP – *Hypertext Transfer Protocol*

IoT – *Internet of Things*

OWASP – *Open Web Application Security Project*

P2P – *Peer to Peer*

PoS – *Proof of Stake*

Pow – *Proof of Work*

SDLC – *Software Development Life Cycle*

SWC – *Smart contract Weakness Classification*

UML – *Unified Modelling Language*

WSTG – *Web Application Security Testing Guide*

Coimbra Business School | ISCAC
Quinta Agrícola – Bencanta, 3045-601 Coimbra
Tel +351 239 802 000; E-mail: secretariado@iscac.pt; www.iscac.pt

INTRODUÇÃO

Com as facilidades trazidas pelos avanços tecnológicos, a ampla disposição de materiais e conhecimento, fica relativamente acessível para qualquer pessoa com o mínimo de conhecimento adentrar no mundo de aplicações em Blockchain. As diversas vantagens e potencialidades da tecnologia trazem perspetivas promissoras no que diz respeito a fatores chave e de extrema relevância, que estão ligados à segurança de dados em uma determinada rede e aplicações, a auditabilidade possível por natureza e o aspeto de contribuição. Entretanto, embora haja facilidades para o desenvolvimento de aplicações descentralizadas, falhas graves podem comprometer de diversas formas uma aplicação e quanto mais tempo se leva para descobrir e corrigir as falhas, mais prejuízos são contabilizados.

Desta forma, é necessário ter atenção no desenvolvimento de aplicações e seguir as boas práticas desde o início do processo para reduzir as possibilidades de falhas que podem ser evitadas com o acompanhamento necessário. A decisão de utilização de aplicações em Blockchain necessita de uma série de pontos a serem seguidos para um melhor aproveitamento. Primeiramente é necessário um conhecimento geral prévio sobre Blockchain, o funcionamento e qual rede será utilizada para implementação. Em seguida, é necessário perceber as limitações da tecnologia para poder dimensionar as reais funcionalidades da aplicação e o que será armazenado. Por fim a definição da arquitetura da aplicação, importante para definição de como estará disposta a aplicação, em um ambiente completamente descentralizado ou em ambiente misto.

É preciso perceber que, mesmo com as características apresentadas em Blockchain como imutabilidade dos dados, a auditabilidade e criptografia de alto nível não é possível garantir que os dados estarão seguros somente pelo facto de estarem em Blockchain. Existem diversas vulnerabilidades catalogadas por entidades e que já causaram perdas significativas à empresas, de forma que o acompanhamento desde o início do projeto pode guiar para uma aplicação livre de tais vulnerabilidades. Além da utilização de *frameworks* de segurança disponibilizados, diversas aplicações podem e devem ser utilizadas para mitigar erros e melhorar o código da aplicação descentralizada, uma vez que se torna mais complexo pelo desenvolvimento de um *front-end* e de um *back-end*.

Coimbra Business School | ISCAC
Quinta Agrícola – Bencanta, 3045-601 Coimbra
Tel +351 239 802 000; E-mail: secretariado@iscac.pt; www.iscac.pt

Um outro ponto que torna fundamental a aplicação de práticas de segurança em ambientes descentralizados está no facto de que, uma vez que os *smart contracts* estão em uma Blockchain, já não podem ser alterados e qualquer tipo de correção posterior de falhas representa processos custosos do ponto de vista financeiro e tecnológico.

O presente projeto tem por objetivo demonstrar a exequibilidade da construção de aplicações descentralizadas para armazenamento de dados, neste caso em concreto de dados relativos a orçamentos, que garantem um ambiente mais seguro para armazenamento dos mesmos, pois gozam das características de imutabilidade e descentralização das Blockchains. Associada à temática da segurança o presente projeto também avalia as vulnerabilidades da própria aplicação descentralizada desenvolvida, com vista à utilização de boas práticas no desenvolvimento de aplicações baseadas em Blockchain, desde a sua conceção.

O trabalho apresentado neste relatório possui quatro capítulos, para além dos capítulos da Introdução e da Conclusão.

O primeiro capítulo é o da revisão de literatura em que se apresentam os principais conceitos de suporte ao desenvolvimento do trabalho, como os conceitos de Blockchain, *smart contracts* e aplicações descentralizadas e se apresentam também as principais áreas de aplicação do Blockchain e a segurança em Blockchain, incluindo algumas vulnerabilidades e ataques mais comuns neste domínio.

No segundo capítulo apresentam-se os passos da metodologia de utilizada para a realização do presente trabalho.

No terceiro capítulo apresenta-se a aplicação descentralizada desenvolvida que teve duas versões dadas as limitações encontradas após o desenvolvimento da primeira versão da mesma, descritas neste capítulo.

No capítulo quarto é feita uma análise de vulnerabilidades à aplicação desenvolvida da qual resultaram ações de melhoria da mesma.

Coimbra Business School | ISCAC
Quinta Agrícola – Bencanta, 3045-601 Coimbra
Tel +351 239 802 000; E-mail: secretariado@iscac.pt; www.iscac.pt

1 REVISÃO LITERATURA

Embora tenha se popularizado no final dos anos 2000, é necessário voltar para a década de 90 para falar da origem da tecnologia disruptiva que é Blockchain. Em 1991 Haber & Scott Stornetta (1991) propuseram um modelo baseado em criptografia, numa rede em cadeia que garantiria a procedência e unicidade de produtos digitais. Neste trabalho, os autores propõem a solução baseada nos princípios de registo único na criação e a premissa de que este registo não pode ser alterado. Neste modelo, encontram-se alguns conceitos fundamentais que foram colocados em prática anos a frente. A utilização de uma formulação matemática chamada *hash*, para gerar um conjunto de caracteres de forma aleatória, com o objetivo da criação de uma chave de identificação única. A outra característica marcante, que também é base da Blockchain é o facto de todas as informações serem armazenadas em um espaço de forma interligada, assim haveria um registo sequencial de cada documento ou formato digital adicionado à estrutura. A utilização de um validador de *hashes*, que garantiria a autenticidade de quem está tentando adicionar um documento à estrutura.

Todas essas características e técnicas deram origem ao que conhecemos hoje como Blockchain. Essa estrutura de blocos interligados utiliza conceitos de criptografia para garantir segurança, imutabilidade e privacidade e começou a ganhar notoriedade de facto em 2008. Nesse ano Nakamoto (2008) ficou conhecido com o projeto que conceptualizou a primeira Blockchain em seu sistema de transações eletrónicas chamado *Bitcoin*. No artigo publicado por Nakamoto (2008) é explicada a estrutura que é utilizada em *Bitcoin*, ficando óbvio os conceitos abordados por Haber & Scott Stornetta (1991) dezassete anos antes, referenciados no artigo de Nakamoto (2008).

Desde então, as potencialidades de Blockchain vêm sendo exploradas e criando diversas perspetivas de avanço. Ao passar dos tempos, outras plataformas de transações eletrónicas foram criadas como *Ethereum*, *Dogecoin*, *Litecoin* e tantas outras. No entanto, é importante tratar *Ethereum* com um pouco mais de atenção. Criada em 2013, por Vitalik Buterin (Buterin, 2014), como alternativa ao *Bitcoin*, a Blockchain pública *Ehtereum* apresentou novas funcionalidades, sendo a mais importante a possibilidade de utilização de *smart contract*s (em português contratos inteligentes), o que possibilitou a interação

Coimbra Business School | ISCAC
Quinta Agrícola – Bencanta, 3045-601 Coimbra
Tel +351 239 802 000; E-mail: secretariado@iscac.pt; www.iscac.pt

com códigos e a execução de diferentes funções dentro da plataforma, além de emergir com uma ideia de colaboração junto à comunidade de desenvolvedores que tem por objetivo contribuir com ambientes mais confiáveis e estáveis.

1.1 Blockchain

A tecnologia utilizada na estrutura de aplicações de cripto moedas vem se tornando cada vez mais popular e ganhando espaço nas mais diversas áreas de atuação. Blockchain pode ser definido de forma simples como uma estrutura de dados descentralizados armazenados em blocos e que tem como principais características a imutabilidade e rastreabilidade (Guo et al., 2020). Estes blocos possuem informações sobre as transações gravadas em uma determinada operação e a cada novo conjunto de registo que forma um bloco é adicionado na cadeia, criando uma estrutura de dados interconectados, que possuem referências dos blocos anteriores e utilizam mecanismos de consenso para confirmar e adicionar novos blocos à cadeia de forma segura (Sanka & Cheung, 2021).

Popularizada a partir de 2009 com sua utilização em *Bitcoins*, a estrutura de dados elimina a necessidade de uma terceira parte para armazenamento tornando as transações mais transparentes e seguras, uma vez que todos os integrantes da rede possuem uma cópia da informação e seu conteúdo garantido por criptografia de alto nível. Além dessas características, a comunicação dentro de uma Blockchain é feita ponto a ponto, aumentando assim a confiança das transações, uma vez que a conexão com a segunda parte é feita de forma direta com a devida identificação das partes utilizando formas de criptografia (Bhutta et al., 2021).

Estudos recentes vêm mostrando que a usabilidade do Blockchain vai muito além do sector financeiro e seu potencial tecnológico pode contribuir de forma significativa para preencher lacunas que foram abertas com os avanços tecnológicos dos últimos anos, nomeadamente em relação à segurança e privacidade dos dados. Entusiastas do potencial do Blockchain enxergam a forma descentralizada e com vários mecanismos de segurança e confiança que são características básicas da tecnologia como a solução ideal para garantir segurança e privacidade em ambientes *Internet of Things* (IoT) (Hassan et al., 2019).

Coimbra Business School | ISCAC
Quinta Agrícola – Bencanta, 3045-601 Coimbra
Tel +351 239 802 000; E-mail: secretariado@iscac.pt; www.iscac.pt

Embora seja uma tecnologia que está no início de sua exploração para além das cripto moedas, ainda possui uma série de questões relacionadas à perspetiva de funcionamento e escalabilidade computacional (Gao et al., 2018). Entretanto, é percetível que soluções baseadas em Blockchain podem apresentar respostas para questões que envolvem segurança, transparência e maior confiabilidade no que diz respeito a dados e ou transações.

1.1.1 Estrutura da Blockchain

A estrutura de uma Blockchain se assemelha a uma base de dados comum, só que completamente descentralizada. Esta estrutura, possui como principais componentes os nós e os registos de dados, como é visto em Salman et al. (2018). Cada nó de uma rede Blockchain funciona como cliente e servidor, podendo fazer ou receber requisições diretamente de outros nós existentes na rede (Bhutta et al., 2021).

Para compreender melhor, enumera-se de seguida os principais componentes e conceitos de uma Blockchain (Liu et al., 2019; Monrat et al., 2019):

- *Transactions* – são os dados propriamente dito e armazenados entre os nós da rede;
- *Nodes* – podemos entender como nós como todos os dispositivos ou computadores que estão ligados em uma Blockchain*;*
- *Blocks* – são blocos que compõem uma Blockchain que armazenam informações e transações. A estrutura de um bloco é composta por alguns elementos, a saber:
  - *Hash* – Trata-se de um conjunto de números gerados automaticamente por um algoritmo e é utilizado para identificar um bloco;
  - *Previous hash* – Este é o identificador do bloco anterior de cada bloco;
  - *Merkle root* – É onde estão guardadas as informações referentes a cada *hash* dos blocos em uma Blockchain;
- *Consensus mecanism* – podem ser entendidos como protocolos estabelecidos para que haja um acordo de aceitação sobre novas transações, juntamente com validação de novos blocos;
- *Miners* – os mineradores podem ser vistos como atores, que criam ou validam transações.

Coimbra Business School | ISCAC
Quinta Agrícola – Bencanta, 3045-601 Coimbra
Tel +351 239 802 000; E-mail: secretariado@iscac.pt; www.iscac.pt

As transações dentro de uma Blockchain são armazenadas em blocos sequenciais e o primeiro bloco dentro de uma rede Blockchain é chamado genesis (Monrat et al., 2019). Além de um conjunto de transações, um bloco contém um cabeçalho com informações de referência que são utilizadas para identificar a sequência dos blocos (Liu et al., 2019).

Dentro de uma Blockchain, os utilizadores que também podem ser vistos como nós dentro da rede e possuem uma cópia dos dados integral ou parcial. A criptografia é utilizada para manter a segurança de comunicação entre os nós (Bhutta et al., 2021). Para que uma transação seja adicionada a um bloco, é necessário que todos os outros nós estejam de acordo com a procedência e veracidade das transações. Os mecanismos de consenso fazem este papel e complementam a camada de segurança, fazendo com que os dados tenham a integridade garantida dentro da rede descentralizada (Ali Syed et al., 2019).

O processo de adicionar uma nova transação a um bloco é chamado de mineração. Para que um minerador tenha o direito de adicionar uma transação, ele deve se submeter às regras e satisfazer as condições impostas pelos mecanismos de consenso, que pode ser visto como um processo eletivo para determinação do escolhido (Belotti et al., 2019).

1.1.2 Tipos de Blockchain

Conceitualmente existem três tipos de Blockchain; públicas, privadas e consórcio. Uma Blockchain pública é uma estrutura de dados descentralizados onde qualquer um pode se candidatar a ser um nó da cadeia de blocos, desde que atenda aos requisitos dos mecanismos de consenso (Hassan et al., 2019). Seus participantes são incentivados financeiramente para executar o processo de mineração dos blocos.

Blockchain de consórcio ou híbridas são estruturas que possuem um pouco da privada e pública. Nestes ambientes, organizações fazem parte de um grupo e cada uma delas pode ter sua própria Blockchain trabalhando dentro de uma rede descentralizada maior. Um conceito de uma rede colaborativa, onde cada organização possui sua rede e dados, quer compartilhar informações com outras organizações. Apesar de ser um ambiente com este viés colaborativo, é preciso reiterar que os dados são controlados e tornados públicos dentro da rede somente por autorização (Chen et al., 2020).

Coimbra Business School | ISCAC
Quinta Agrícola – Bencanta, 3045-601 Coimbra
Tel +351 239 802 000; E-mail: secretariado@iscac.pt; www.iscac.pt

E por último a Blockchain privada, que como o próprio nome indica, é uma rede fechada para participantes em um ambiente restrito e em que o conceito de colaboração entre os membros da comunidade não é a essência de funcionamento (Gomathi et al., 2021). Apesar de manter a mesma estrutura descentralizada com nós, a Blockchain privada está sob o controlo de uma organização que mantém sua estrutura de dados e esta organização ou autoridade é quem determina fatores como quem irá minerar os blocos (Ahmed & Pathan, 2020). Embora esta estrutura seja um pouco contrária ao conceito e entendimento básico de Blockchain, que idealiza uma estrutura de dados onde todos possam ter acesso e não haja interferência de terceiros, não deixa de ser uma solução para empresas que não querem ou não podem, por motivos culturais corporativos ou por contrato, expor seus dados de uma forma pública, ainda que os mesmos estejam criptografados.

1.1.3 Relação de confiança dentro da Blockchain

Para uma melhor compreensão de como é estabelecida a confiança entres os nós dentro de uma rede Blockchain, deve-se levar em consideração alguns conceitos que envolvem o aspeto comportamental dos participantes. Para que este comportamento seja explicado, é necessário abordar os conceitos do problema dos generais Bizantinos e a teoria de jogos.

O problema dos generais Bizantinos é um conceito amplamente utilizado em diversas áreas e tenta resolver o problema da confiabilidade dentro de um sistema descentralizado. Neste problema, imagina-se que existam vários generais em localidades diferentes que têm por objetivo atacar um inimigo comum, necessitando, para ter sucesso, agir ao mesmo tempo e com a mesma decisão (Lloyd, 1994). Como estão em locais diferentes, precisam enviar mensageiros para combinarem o tempo exato do ataque. O problema acontece exatamente por conta desse acordo, uma vez que não garantido que os mensageiros não serão capturados, corrompidos ou que um ou mais generais ajam de forma maliciosa. O consenso dentro de uma rede Blockchain é fundamental, uma vez que os nós são independentes e precisam garantir a consistência das informações na Blockchain (Monrat et al., 2019).

A teoria dos jogos é um estudo criado em economia com o objetivo de analisar as interações comportamentais e racionalidade entre tomadores de decisões, como pode ser descrito em (Song et al., 2019). Este conceito é utilizado amplamente para analisar o

Coimbra Business School | ISCAC
Quinta Agrícola – Bencanta, 3045-601 Coimbra
Tel +351 239 802 000; E-mail: secretariado@iscac.pt; www.iscac.pt

aspeto comportamento em Blockchain, onde os decisores podem ser vistos como os mineradores ou os utilizadores (Liu et al., 2019).

Um modelo bem utilizado para este estudo dentro da Blockchain é o dilema dos prisioneiros. Neste dilema, dois prisioneiros estão sob custódia da polícia e são interrogados separadamente e suas decisões entre se declararem culpados ou inocentes impactam diretamente no resultado de suas sentenças, que poderiam variar desde a soltura de um quanto em penas severas. No entanto, dilema mostra que caso os dois acusados estabeleçam uma estratégia e se declarem culpados, a sentença que os dois cumprirão se torna a mais racional dentre as possibilidades (Kong & Yue, 2020).

Uma rede em Blockchain possui uma estrutura que resiste ao problema dos generais Bizantino através dos mecanismos de consenso, uma vez que unindo suas propriedades baseadas em criptografia e as validações, mesmo havendo nós com comportamento malicioso, ainda é possível manter a integridade da rede, como é visto em Zhang et al. (2019).

A teoria de jogos aplicada em Blockchain, explica como numa rede descentralizada onde os participantes não se conhecem e onde não há como confiar uns nos outros, é mantida a colaboração. De uma forma racional, para os participantes de uma Blockchain agir de forma honesta traz benefícios a todos pelo tanto que é investido para manter o sistema económico do que tentar obter vantagens de forma desonesta (Li et al., 2020). Isto porque os mecanismos de consenso garantem que os participantes dediquem muito tempo e esforço computacional para o processo de mineração, tornando ilógico um comportamento errático.

1.1.4 Mecanismos de consenso

Sendo uma estrutura de dados descentralizada, onde seus participantes agem de uma forma cooperativa e não havendo nenhum tipo de entidade ou ponto central para validar as transações, torna-se necessário a existência de protocolos para proceder com estas validações dentro de uma Blockchain. Um dos grandes feitos em Blockchain é a capacidade de estabelecer esta confiança nas transações sem uma terceira parte para intermediar (Saxena et al., 2021). Para atingir um acordo de validação de transações

Coimbra Business School | ISCAC
Quinta Agrícola – Bencanta, 3045-601 Coimbra
Tel +351 239 802 000; E-mail: secretariado@iscac.pt; www.iscac.pt

dentro da Blockchain são utilizados os chamados mecanismos de consenso, que tem como objetivo garantir a consistência e procedência das transações.

Para a criação de um novo bloco, um utilizador que faz parte da Blockchain se propõe a resolver um desafio matemático e ganhar o direito de adicionar e oferece recompensas financeiras como incentivo e estímulo para mineração. Existem atualmente diversos tipos de mecanismos de consenso e cada um possui suas implicações, vantagens e desvantagens (Bamakan et al., 2020). Seguem abaixo alguns dos mecanismos de consenso, a título de exemplo (Bouraga, 2021; Han et al., 2022):

- *Proof-of-Work* (PoW) – Este mecanismo é utilizado para que cada utilizador dentro de uma Blockchain faça um esforço computacional para resolver um enigma de combinação de números para ter direito a adicionar uma informação de blocos. O grau de dificuldade é variado, mas normalmente obriga aos computadores processarem milhões e milhões de combinações, o que leva um tempo considerável para resolver.
- *Proof-of-Stake* (PoS) – Este mecanismo é baseado em um sorteio para definir qual o nó será escolhido para criar um novo bloco, levando em consideração alguns aspetos do participante, como a quantidade de dinheiro que possui. Este mecanismo é menos custoso do que o PoW, do ponto de vista de utilização de tempo.
- *Delegated-Proof-of-Stake* (DPoS) – Este mecanismo é semelhante ao *Proof-of-Stake*, mas com a diferença que neste caso os escolhidos para adicionar um novo bloco são definidos por um grupo de forma aleatória. Por ser um mecanismo com número de participantes limitado, os blocos podem ser criados de uma forma mais rápida em relação aos outros.

1.2 *Smart contracts*

Os *smart contract*s ou apenas contratos inteligentes, são códigos que servem para estabelecer um contrato digital entre duas ou mais partes. Os *smart contract*s podem ser vistos como uma forma de cumprir acordos pré-definidos, estabelecendo um registo

Coimbra Business School | ISCAC
Quinta Agrícola – Bencanta, 3045-601 Coimbra
Tel +351 239 802 000; E-mail: secretariado@iscac.pt; www.iscac.pt

definitivo na Blockchain e sem a necessidade de uma terceira parte para garantir a confiabilidade do acordo, não existindo possibilidade de editá-lo (Bhushan et al., 2021).

Para compreender o funcionamento dos contratos inteligentes, podemos tomar como exemplo uma situação em que duas pessoas pretendem negociar a venda de um carro. Quando os termos de venda são estabelecidos e acordados entre ambas as partes, é feito o registo através do contrato inteligente e o mesmo é enviado, neste caso usando a Blockchain. A partir do momento que o contrato é enviado, um registo único é feito e o comprador tem a confiança de que os dados não serão alterados, o que garante que não haverá tentativa de alterar os termos. Em seguida, o comprador pode efetuar o pagamento, sem a necessidade de uma entidade para intermediar a negociação.

1.3 Áreas de atuação

Embora tenha um forte apelo na área de finanças, Blockchain está sendo utilizado em diversas áreas diferentes e o potencial da tecnologia amplia ainda mais as possibilidades de aplicação e utilização.

Atualmente podemos encontrar já em estudos e aplicação avançados, a utilização de Blockchain em áreas como Internet das Coisas (IoT), Big Data, Computação nas nuvens, Gestão de identidades, Contratos Inteligentes, Soluções de negócios, Gestão de Informação médica, Comércio e Indústria (Gao et al., 2018). Para além das áreas citadas acima (Taveira, 2020) também acrescenta importantes domínios de utilização atual da Blockchain, como Governo, Privacidade e Segurança, área de Educação e Cadeia de Abastecimento.

Em todos as áreas citadas acima, as características da Blockchain como imutabilidade, rastreabilidade e transparência, surgem como solução em potencial para problemas encontrados em ambientes que têm por característica a descentralização ou ambientes que necessitam garantir um nível de segurança e transparência no tratamento de dados e transações sensíveis.

Coimbra Business School | ISCAC
Quinta Agrícola – Bencanta, 3045-601 Coimbra
Tel +351 239 802 000; E-mail: secretariado@iscac.pt; www.iscac.pt

1.4 Aplicações descentralizadas

As *Decentralized Application (*DApp*)*, surgem num contexto em que Blockchain vem ganhando maior notoriedade e importância. As DApp são aplicações que se dividem em duas partes, um *front-end* que tem uma interface de comunicação com o utilizador e um *back-end*, que tem uma aplicação que integra com uma Blockchain (Wohrer et al., 2021). Neste contexto, os dados são armazenados numa Blockchain, utilizando os *smarts contracts* para compor e manter a conexão.

Uma DApp pode ser completamente descentralizada ou ter parte da aplicação em ambientes centralizados, como é visto em Duran et al. (2020). Neste caso, parte dos dados que se referem a dados sensíveis podem ser armazenados em um ambiente centralizado por conta de privacidade, enquanto as transações permanecem em ambiente descentralizado.

O desenvolvimento e concessão de DApp exige o conhecimento e inserção de bibliotecas e componentes. No desenvolvimento das interfaces de utilizador, é importante levar em consideração a utilização de biblioteca de códigos para interação com os *smart contracts* e Blockchain, como por exemplo *Web3.js* (*Ethereum* Revision, 2022b) e o *Ether.js* (Moore, 2021). Outro componente fundamental numa DApp é a utilização de uma carteira digital, para gerir os *tokens* e as identidades dos utilizadores, como é visto em Wohrer et al. (2021).

O desenvolvimento de uma DApp possui passos semelhantes ao de uma aplicação *web* comum. Em Trojanowska et al. (2020), é apresentada a utilização de uma metodologia em que inclui os mesmos passos de aplicação comum, como design, implementação, verificações de segurança e manutenção.

1.5 Segurança e Blockchain

A estrutura e mecanismos de segurança utilizados em redes Blockchain proporcionam um ambiente nativamente mais robusto e com respostas eficazes em relação à segurança dos dados. Entretanto, é importante ressaltar que há ainda assim a necessidade de estabelecer métodos para manter as redes Blockchain seguras.

Coimbra Business School | ISCAC
Quinta Agrícola – Bencanta, 3045-601 Coimbra
Tel +351 239 802 000; E-mail: secretariado@iscac.pt; www.iscac.pt

O crescimento constante da preocupação com a segurança em ambientes que utilizam a tecnologia Blockchain é altamente justificável e requer maior colaboração para tentar criar ambientes com maior confiabilidade em diversos aspetos. Conforme mencionado em Wang et al. (2021), os prejuízos causados por hackers em provedores de cripto moedas tem crescido exponencialmente com a popularização e gerado perdas consideráveis para diversas empresas e particulares. Para os autores, apesar de existirem mecanismos que por natureza dificultem as ameaças, não é fácil conter os cibercriminosos.

Em Wen et al. (2021) é feito um estudo sobre segurança em Blockchain, levando em consideração seis camadas que os autores utilizam para análise. As camadas são a de dados, redes, camada de consenso, incentivo, contrato e aplicação. Primeiramente é feito uma abordagem teórica a cada uma dessas camadas, explicando do que trata cada uma e seus componentes. Em seguida é feita uma estruturação de uma série ataques de segurança identificados e estudados pelos autores, segmentados de acordo com cada camada especificada, trazendo possíveis medidas para resolução ou minimização dos problemas encontrados.

Já em Bhushan et al. (2021), para além de uma caracterização estrutural da Blockchain e um estudo sobre ameaças de segurança estudadas abordando também medidas propostas para ultrapassar o problema, trazem também uma visão geral da arquitetura da Blockchain e suas áreas de aplicação. Os autores também reforçam que quatro fatores integram uma Blockchain e são pontos fortes para garantir alguns aspetos de segurança no ambiente, como a rede ponto a ponto, contratos inteligentes, encriptação assimétrica e base de dados distribuída. Por fim os autores enumeram privacidade e anonimato, mineração dos dados, compatibilidade e a consulta de dados armazenados em blocos como sendo os desafios enfrentados em ambientes com Blockchain.

Em Gao et al. (2018) é defendido que aplicações baseadas em Blockchain precisam também ser bem estruturadas desde a sua conceção, visto que qualquer falha na implementação ou no código podem potencializar brechas de segurança.

É de suma importância que haja um estudo para que as brechas de segurança sejam minimizadas na implementação ou utilização de uma estrutura em Blockchain. Em

Coimbra Business School | ISCAC
Quinta Agrícola – Bencanta, 3045-601 Coimbra
Tel +351 239 802 000; E-mail: secretariado@iscac.pt; www.iscac.pt

Homoliak et al. (2021), podemos ver que as potenciais vulnerabilidades dentro de uma rede Blockchain podem estar espalhadas nas diversas camadas existentes.

1.6 Vulnerabilidades e ataques

Embora Blockchain possua características como imutabilidade, rastreabilidade e utilização de criptografia de alto nível, não há como garantir a segurança total e evitar a exploração de vulnerabilidades em sistemas, como é visto em Ahmed & Pathan (2020).

A identificação das vulnerabilidades e os possíveis ataques que podem ocorrer dentro de um ambiente descentralizado corrobora com necessidade de constante melhoria e evolução no que diz respeito à segurança. Seguem abaixo alguns tipos de ataques baseados em vulnerabilidades conhecidas (Anita, 2019; Jonathan & Sari, 2019; Moubarak et al., 2018; Wang et al., 2021; Wen et al., 2021):

- *Timehijacking* – Este ataque baseia-se em outra forma de isolar um nó ou o nó e seus sucessores do restante da rede. Neste caso, os *hackers* precisam alterar o contador interno do nó, fazendo com que o mesmo exceda o tempo em que um bloco é rejeitado, fazendo com que ele seja separado da restante Blockchain.
- *Sybil attacks* – Os ataques desta natureza têm como objetivo tomar controlo de múltiplos nós dentro da rede utilizando diferentes identidades, mas controladas por um só agente malicioso.
- *Selfish mining attack* – Este ataque tem como objetivo o aumento dos ganhos por parte de um *hacker*, que basicamente redireciona recompensas para si. Utilizando esta estratégia, o atacante numa rede, após decifrar o problema matemático proposto no mecanismo de consenso, não anuncia seu feito e deixa os outros participantes continuar a resolver o problema, revelando em um momento propício para ele. Neste meio tempo, o atacante prossegue com a mineração de outro bloco e "esconde" seus feitos em uma Blockchain paralela. Quando um outro participante da rede original consegue resolver o problema anterior, o atacante então revela sua rede escondida. Isto faz com que ele se beneficie do facto que, numa Blockchain que utiliza PoW por exemplo, a rede reconhecida pelos blocos que empregam mais esforços computacionais. Isso faz com que a

Coimbra Business School | ISCAC
Quinta Agrícola – Bencanta, 3045-601 Coimbra
Tel +351 239 802 000; E-mail: secretariado@iscac.pt; www.iscac.pt

rede paralela do atacante seja reconhecida e ele consiga obter todas as recompensas dos participantes.

- *Double spending attack* – Este ataque caracteriza-se quando um criminoso tenta utilizar o mesmo crédito digital por mais de uma vez. Nesta situação, o atacante aproveita o tempo que uma transação leva para ser confirmada e utiliza os mesmos fundos para efetuar uma nova transação.
- *Reentrancy* – Neste ataque, os criminosos utilizam um contrato externo para chamar uma função de outro contrato e, enquanto uma transação está em processo de validação e aceitação, os criminosos continuam a retirar o dinheiro enquanto houver saldo.
- *Denial of Service* (DoS) – Neste tipo de ataque o *hacker* tem como objetivo deixar o nó ou nós que estão sob ataque indisponíveis, através de uma quantidade de requisições que incapacita os atacados.
- *Quantum attack* – Baseia-se em atacantes que utilizam de computadores quânticos para simplificar os problemas matemáticos gerados pelos mecanismos de consenso. Como já é de conhecimento, os computadores que utilizam tecnologia quântica têm poder de processamento muito maior do que os computadores convencionais, o que poderia se tornar uma ameaça para a mineração dentro de uma Blockchain.
- Engenharia Social – Esta técnica de ataque está muito mais ligada à interação entre os indivíduos. Nesta situação, os criminosos tentam obter informações sensíveis como senhas e informações da conta como abordagem. A exploração da fragilidade das pessoas, falta de treinamento, conhecimento sobre o que se pode ser informado estão entre fatores essenciais neste tipo de ataque.

1.7 Segurança em aplicações descentralizadas

No que diz respeito à segurança em aplicações descentralizadas, para além das práticas comuns em aplicações *web*, é preciso ter em consideração as vulnerabilidades relacionadas com a Blockchain e os *smart contracts*. Em Ji et al. (2021) é feito um estudo das vulnerabilidades encontradas nos *smart contracts*, como o *Reentrancy attack*, *DoS*,

Coimbra Business School | ISCAC
Quinta Agrícola – Bencanta, 3045-601 Coimbra
Tel +351 239 802 000; E-mail: secretariado@iscac.pt; www.iscac.pt

entre outras. Para além das brechas apresentadas, também são sugeridas formas de prevenir tais vulnerabilidades.

Para garantir a segurança do *smart contract*, é necessário ter em conta diferentes aspetos, como a utilização de padrões e bibliotecas seguras, técnicas e recomendações de segurança, bem como utilizar ferramentas de análise, desenvolvidas com o objetivo de identificar diferentes tipos de vulnerabilidades nos contratos (Ren et al., 2021). Conforme é visto em (Kim & Lee, 2020), diversas ferramentas de análise como Manticore, Mythril, Mythx, Securify e Smartcheck podem ser utilizadas para detetar vulnerabilidades e falhas de segurança. Não obstante, a melhor forma de ter um código seguro é iniciar a programação obedecendo as boas práticas de segurança para que as vulnerabilidades sejam evitadas.

As ferramentas de análise de vulnerabilidades de *smart contracts* podem efetuar análises estáticas ou dinâmicas, como é o caso da MythX's, sendo que algumas só realizam um tipo de análise, como é o caso da Slither que só faz análises estáticas. Enquanto que as ferramentas de análise estática executam testes sem executar o *smart contract* e analisam as possibilidades de comportamento do código, as ferramentas de análise dinâmica utilizam um método onde o código da aplicação é executado, simulando um ataque e as possibilidade de exploração das vulnerabilidades (Praitheeshan et al., 2019). Por vezes é necessário utilizar mais que uma ferramenta para garantir uma análise geral mais rigorosa (Zhdarkin & Anureev, 2021).

No âmbito das aplicações *web*, a ação prévia para deteção de vulnerabilidades segue a mesma ideia. Em Guaman et al. (2017) é visto que o processo de seguimento de normas e boas práticas de segurança regidas por entidades como a *Open Source Foundation for Application Security* (OWASP) são fundamentais para mitigar as brechas de segurança. Para além disto, os autores utilizam ferramentas como OWASP ZAP, Vega e Wireshark para as análises apresentadas (Guaman et al., 2017). Já em Devi (2020) são utilizadas ferramentas como OWASP ZAP e Nikto para analisar as vulnerabilidades de aplicações *web*, alertando os autores para a importância dos testes e tentando explicar em seu trabalho como a maioria dos ataques ocorrem, com exemplos de algumas ferramentas como Sparta, Zenmap e Nmap para rastrear as vulnerabilidades de uma aplicação *web*.

Coimbra Business School | ISCAC
Quinta Agrícola – Bencanta, 3045-601 Coimbra
Tel +351 239 802 000; E-mail: secretariado@iscac.pt; www.iscac.pt

No trabalho de Kunda & Alsmadi (2022), são apresentadas um conjunto de ferramentas de código aberto, pagas ou no modelo misto como OWASP ZAP, Metasploit, SoapUI e Vega para analisar vulnerabilidades. Os autores utilizam as ferramentas de código aberto, nomeadamente OWASP ZAP e a SoapUI para detetar as vulnerabilidades em um *website*, classificá-las e apresentar soluções de acordo com o método estabelecido, que tem como objetivo identificar os objetivos, procurar ferramentas de análise, executar testes, validar os resultados, documentar os resultados e repetir os testes até que os objetivos estejam alcançados.

1.8 Análise de segurança em aplicações descentralizadas

Para a análise de segurança, serão utilizadas uma série de boas práticas e ferramentas recomendadas pela OWASP para o desenvolvimento e testes gerais do componentes e boas práticas recomendadas pela *Ethereum* (Consensys, 2022; Trojanowska et al., 2020).

Assim, baseado no modelo sugerido na OWASP, três técnicas de análise de segurança foram identificadas para utilizar no escopo do projeto, passando pelo código da DApp, o código do contrato inteligente e os componentes de cada um que podem representar brechas de segurança.

Para o *front-end*, será feita a utilização da base do *framework* de testes fornecido pela OWASP, seguindo as etapas abaixo:

- *Threat Modeling* – A modelagem de ameaças é um detalhamento das possíveis brechas do projeto que está a ser desenvolvido. Isto permite que o desenvolvedor possa minimizar os riscos a partir do momento que os pontos de falha são identificados ainda no desenvolvimento.
- *Code Review* – A revisão do código fonte das aplicações utilizadas. Esta revisão tem como objetivo evitar falhas comuns em funções, operações matemáticas, etc.
- *Penetration Testing* – O teste de penetração permite ter uma visão prática da exploração das vulnerabilidades de uma aplicação. Estes testes permitem ainda obter respostas práticas em caso de correções do código e também outras possíveis vulnerabilidades não catalogadas.

Coimbra Business School | ISCAC
Quinta Agrícola – Bencanta, 3045-601 Coimbra
Tel +351 239 802 000; E-mail: secretariado@iscac.pt; www.iscac.pt

Para auxiliar nas análises de segurança, algumas ferramentas identificadas na literatura, como Mythx e OWASP ZAP serão utilizadas com o propósito de facilitar a identificação das vulnerabilidades e segurança.

Em relação aos contratos inteligentes, serão seguidas as boas práticas de desenvolvimento abordadas e recomendadas pela *Ethereum* (Trojanowska et al., 2020). Esse conjunto de boas práticas abordam aspetos básicos que podem acontecer no desenvolvimento de um contrato e com o tempo, podem se tornar difíceis de perceber e gerar problemas futuros ainda maiores.

Coimbra Business School | ISCAC
Quinta Agrícola – Bencanta, 3045-601 Coimbra
Tel +351 239 802 000; E-mail: secretariado@iscac.pt; www.iscac.pt

2 METODOLOGIA

O presente projeto, como referido no capítulo da Introdução, tem por objetivo o desenvolvimento de uma aplicação descentralizada e avaliação da segurança da mesma recorrendo a ferramentas de análise de vulnerabilidades. Dito isto, o que será apresentado é um cenário onde há a identificação de um problema, a contextualização do mesmo, a avaliação dos dados e sugestão de soluções para o problema inicial. Estas etapas descrevem bem a metodologia que será utilizada neste projeto, que será a *Design Science Research (DSR)*.

A metodologia DSR é um processo de avaliação evolutiva, que tem por objetivo identificar um problema, avaliar e propor solução de forma interativa (vom Brocke et al., 2020). Essa metodologia é largamente utilizada na área dos Sistemas de Informação e permite aos envolvidos em um projeto terem uma noção exata das etapas e processos utilizados para resolução de um problema e seus contributos.

Para que o desenvolvimento de uma aplicação esteja enquadrado na metodologia DSR, segundo Hevner et al. (1996), é necessário que se trate de um problema que ainda não tenha sido resolvido ou uma forma mais eficiente de se resolver. No desenvolvimento da aplicação BudgetDApp, a metodologia DSR irá auxiliar no entendimento de uma forma que tem por objetivo ser mais eficiente na minimização dos problemas relacionados à segurança em aplicações que utilizam Blockchain como base.

A utilização da metodologia DSR fez-se com recurso à grelha DSR (ver Figura 2.1), um quadro proposto por Vom Brocke et al. (2020) que pretende simplificar a utilização da mesma, colocando num único quadro todos os elementos para o planeamento, execução e comunicação da investigação utilizando esta metodologia.

Coimbra Business School | ISCAC
Quinta Agrícola – Bencanta, 3045-601 Coimbra
Tel +351 239 802 000; E-mail: secretariado@iscac.pt; www.iscac.pt

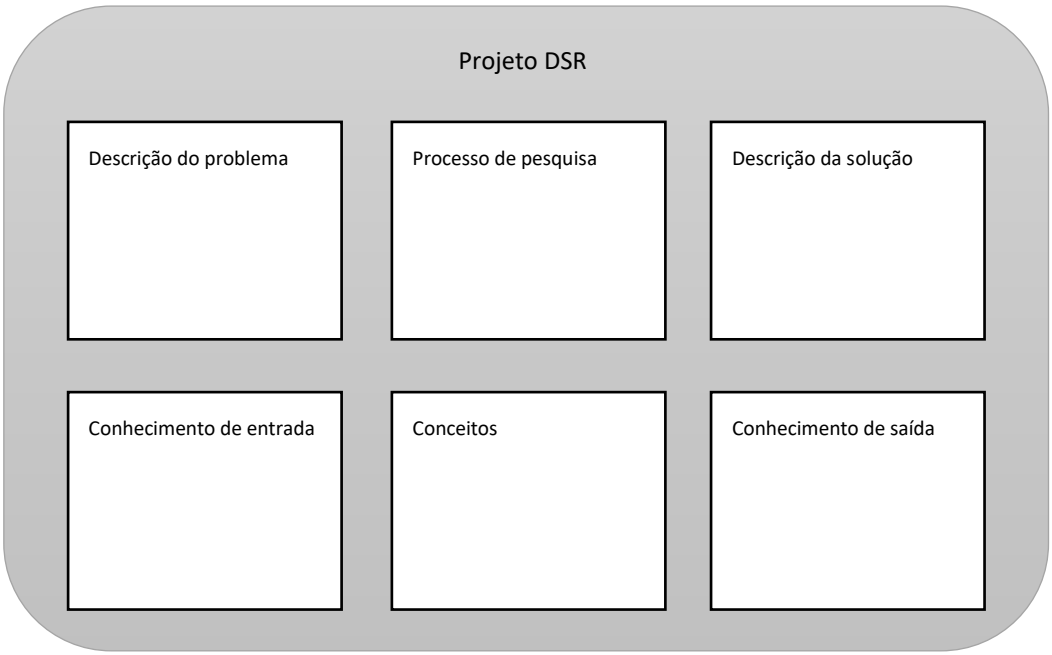


Figura 2.1. Grelha DSR

Adaptado de vom Brocke et al. (2020)

De seguida descrevem-se cada um dos quadros da grelha DSR (ver Figura 2.1) indicando para cada um o que foi feito no presente projeto:

- Descrição do problema

  Neste quadro faz-se a contextualização do problema em questão. No caso do projeto proposto, a problemática está focada em duas situações com a mesma temática, a da segurança. Primeiramente, baseado em uma situação real onde empresas ou pessoas que utilizam meios digitais para troca de e-mails com informações sensíveis, sofrem ataques criminosos através de técnicas como *phishing* para roubar dados e informações vitais. Com este cenário, é proposto o desenvolvimento de uma aplicação que possa ajudar a colmatar os riscos e brechas de segurança nesses ambientes. A segunda situação está também ligada à segurança, mas com a preocupação de aplicar práticas de segurança no desenvolvimento de aplicações descentralizadas, como forma de não desvincular a segurança da informação e dados tanto no desenvolvimento quanto na solução propriamente dita.

Coimbra Business School | ISCAC
Quinta Agrícola – Bencanta, 3045-601 Coimbra
Tel +351 239 802 000; E-mail: secretariado@iscac.pt; www.iscac.pt

- Conhecimento de entrada

  Os conhecimentos de entrada neste caso estão relacionados ao que é necessário para desenvolvimento da solução. Para desenvolver a aplicação BudgetDApp é necessário um conhecimento sobre Blockchain e aplicações descentralizadas. Estes conceitos são fundamentais para que a solução seja desenhada da forma adequada e para que adequações possam ocorrer sem grandes impactos. Outro conhecimento imprescindível para o projeto está ligado à segurança de dados, de várias perspetivas. Da perspetiva do que a aplicação pode oferecer como solução segura, da forma com que os dados sensíveis são tratados dentro da aplicação descentralizada e do ponto de vista do desenvolvimento e dos códigos dos componentes que serão utilizados.

- Processo de pesquisa

  O processo de pesquisa envolve toda a temática que é proposta para a aplicação, a segurança dos dados e da própria DApp. Para se chegar ao desenvolvimento da aplicação proposta, pesquisas voltadas para a área de segurança envolvendo estudos de vulnerabilidades, a utilização de boas práticas e *frameworks* existentes para composição do projeto foram realizados, como se pode verificar no capítulo da Revisão de Literatura. A partir do material reunido, é possível definir como serão aplicadas as boas práticas e técnicas na aplicação. Para colocar em prática, um esquema com um conjunto de etapas é descrito na Figura 2.2 e mostra o modelo completo que será descrito no decorrer do projeto.

Coimbra Business School | ISCAC
Quinta Agrícola – Bencanta, 3045-601 Coimbra
Tel +351 239 802 000; E-mail: secretariado@iscac.pt; www.iscac.pt

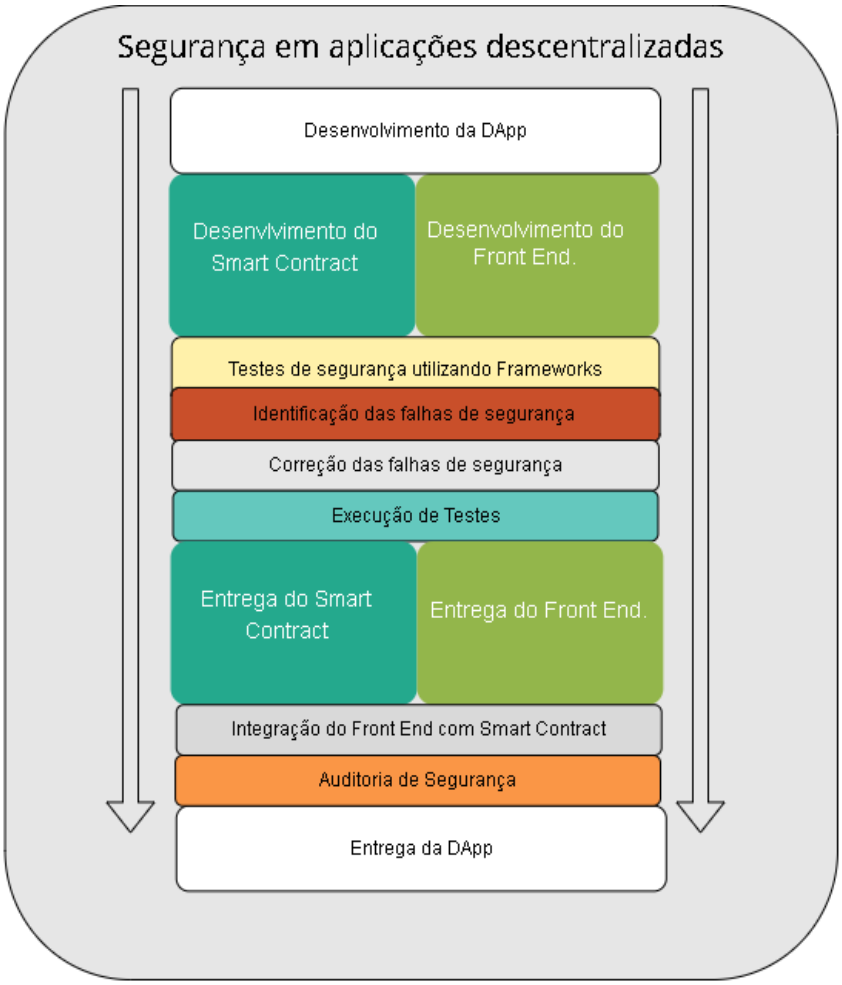


Figura 2.2 Diagrama de blocos do processo de desenvolvimento da BudgetDApp

- Conceitos chave

  Os principais conceitos que envolvem o projeto estão diretamente ligados à segurança e Blockchain. A preocupação da segurança desde a conceção da aplicação, passando por etapas para reduzir riscos e brechas relacionados à segurança, desde a aplicação em si mais voltada à segurança dos dados e o ambiente desenhado para a solução. E a Blockchain, que envolve a base onde a aplicação desenvolvida irá funcionar, fazendo novamente uma ligação com parte de segurança dos dados proporcionados pela tecnologia utilizada. Outro conceito utilizado pode estar relacionado a aplicações descentralizadas, sua composição e possibilidades.

Coimbra Business School | ISCAC
Quinta Agrícola – Bencanta, 3045-601 Coimbra
Tel +351 239 802 000; E-mail: secretariado@iscac.pt; www.iscac.pt

- Descrição da solução

  A descrição da solução baseia-se no desenvolvimento de uma aplicação com base na segurança dos dados. A aplicação BudgetDApp é uma aplicação descentralizada voltada para a troca de orçamentos entre duas partes. Esta aplicação será desenvolvida em uma rede Blockchain *Ethereum* e conta com práticas de segurança para o desenvolvimento e o ambiente idealizado de acordo com o contexto.

- Conhecimento de saída

  Como conhecimento de saída, o pretendido é dar ênfase ao conceito de segurança de uma forma mais ampla. Inicialmente a partir da proposição de uma aplicação descentralizada para proporcionar um ambiente mais seguro para troca de orçamentos e as etapas e boas práticas importantes para reduzir riscos com brechas de segurança.

Coimbra Business School | ISCAC
Quinta Agrícola – Bencanta, 3045-601 Coimbra
Tel +351 239 802 000; E-mail: secretariado@iscac.pt; www.iscac.pt

3 DESENVOLVIMENTO DA *DAPP*

Neste capítulo apresenta-se a DApp desenvolvida no âmbito deste projeto que irá ser objeto de uma análise de vulnerabilidades, apresentada no capítulo seguinte. A criação desta aplicação é uma etapa obrigatória (ver capítulo 3) para a realização da experiência que procura sistematizar a identificação de vulnerabilidades nas DApp.

A DApp a desenvolver, designada de BudgetDApp tem por objetivo guardar documentos de orçamento numa Blockchain, sendo composta de duas partes distintas que interagem entre si: o *smart contract*, que fornece as funcionalidades de interação com a Blockchain e o *front-end* que permite a interação entre o utilizador e as funcionalidades disponibilizadas pelo *smart contract*.

Após o desenvolvimento destes dois componentes, *smart contract* e *front-end*, foram identificadas limitações que implicavam alterações significativas à implementação efetuada, pelo que se optou pelo desenvolvimento de uma nova solução. Dada a importância da primeira solução na identificação das limitações que conduziram à segunda implementação, decidiu-se apresentar neste documento esses dois momentos e não só a aplicação final resultante deste trabalho, pelo que se apresentam de seguida as duas iterações do desenvolvimento da BudgetDApp.

3.1 Primeira iteração

Nesta secção apresentam-se as etapas de desenvolvimento da primeira iteração da BudgetDApp desde a análise e levantamento de requisitos à sua implementação. Dadas as limitações encontradas nesta primeira iteração do desenvolvimento da DApp não se procedeu à realização dos testes de análise de vulnerabilidade.

3.1.1 Análise e enumeração de requisitos

O presente projeto surge a partir de situações reais, onde questões de segurança emergem e necessitam de respostas rápidas. Ataques cibernéticos aumentaram consideravelmente, onde pessoas e empresas ficaram mais expostas nos últimos anos a abordagens cada vez mais sofisticadas de criminosos. Muitas empresas recorrem às ferramentas de *e-mail* como forma de trocar documentos com seus clientes, o que inclui documentos digitais

Coimbra Business School | ISCAC
Quinta Agrícola – Bencanta, 3045-601 Coimbra
Tel +351 239 802 000; E-mail: secretariado@iscac.pt; www.iscac.pt

como orçamentos e faturas com informações sensíveis de valores para transações financeiras, informações sobre empresas, contatos e etc. Em um cenário onde um criminoso utilize um *e-mail* falso com endereços similares com o objetivo de roubar informações ou receber dinheiro por documentos adulterados torna-se cada vez mais factível e depende de atenção cada vez maior por conta das partes envolvidas nas trocas de documentos digitais. O roubo dessas informações utilizando técnicas como *phishing*, onde o criminoso se passa por um utilizador válido para enganar outros pode gerar perdas por vezes irreversíveis para empresas, pois a complexidade envolvida nos golpes deixam a possibilidade rastrear ou reaver valores muito pequenas, sem falar que as ações jurídicas e esforços para identificação dos criminosos parecem não acompanhar a evolução tecnológica tanto no aspeto benéfico quanto no aspeto de prevenção e punição de eventuais crimes praticados. Partindo do princípio que as ferramentas de *e-mail* não fornecem segurança suficiente para veiculação de dados sensíveis, a garantia da fidedignidade dos dados e as soluções para certificação de documentos é um processo dispendioso, nos quais muitas vezes depende de um ambiente criado para garantir a troca dos documentos ou ferramentas para assinar digitalmente os documentos são muito caras e mantêm as informações dependentes de uma terceira parte para gerir a segurança e privacidade, uma aplicação descentralizada para armazenar tais documentos mostrou-se uma solução pertinente.

O desenvolvimento da aplicação BudgetDApp surge com o objetivo de proporcionar um ambiente onde os utilizadores possam confiar que as informações veiculadas dentro da aplicação são únicas, neste caso concreto a manipulação de orçamentos, independentes de uma terceira parte para gestão dos dados e que seja confiável.

3.1.1.1 Descrição

A ideia da aplicação como uma ferramenta de geração de orçamentos é trazer uma solução para organizar os orçamentos que têm valor e importância de forma que os mesmos estejam disponíveis digitalmente para consulta e geração de novos orçamentos.

O conceito de colaboração é bem explorado nesta ideia, pois a utilização dos orçamentos em uma base integrada e partilhada entre os utilizadores, permite com que não sejam mais

Coimbra Business School | ISCAC
Quinta Agrícola – Bencanta, 3045-601 Coimbra
Tel +351 239 802 000; E-mail: secretariado@iscac.pt; www.iscac.pt

necessários espaços físicos, necessidade real de impressão, além de possibilitar a verificação de versões dos orçamentos e veracidade dos mesmos.

O armazenamento de orçamentos através de uma aplicação em um ambiente descentralizado pode ser utilizado tanto interna quanto externamente, a depender da necessidade e objetivo da empresa. Isto permite que filiais de uma mesma empresa ou parceiros possam ter acesso, mediante permissão prévia, aos orçamentos sem a necessidade de envio por aplicações terceiras, como *e-mail*.

A aplicação BudgetDApp é um sistema descentralizado de armazenamento e criação de orçamentos, voltado para envio e consulta de cotações entre duas partes. Desta forma, utilizado uma rede Blockchain é possível garantir que os orçamentos visualizados são únicos e inalteráveis. Outro ponto importante é a não necessidade de uma empresa terceira para garantir este ambiente seguro, o que no final das contas mantém o processo direto e mais barato.

3.1.1.2 Casos de uso

De acordo com o que foi descrito anteriormente definiu-se o diagrama de casos de uso da aplicação. A aplicação BudgetDApp é uma aplicação que faz o envio de orçamentos para um outro utilizador, além de proporcionar a funcionalidade de consultas de orçamentos enviados por ele mesmo. Desta forma, existem dois atores que interagem na aplicação: o primeiro será o gestor no qual ficará responsável por distribuir os *token*s que serão adquiridos por clientes e o segundo o utilizador comum, conforme descrito na Figura 3.1.

Coimbra Business School | ISCAC
Quinta Agrícola – Bencanta, 3045-601 Coimbra
Tel +351 239 802 000; E-mail: secretariado@iscac.pt; www.iscac.pt

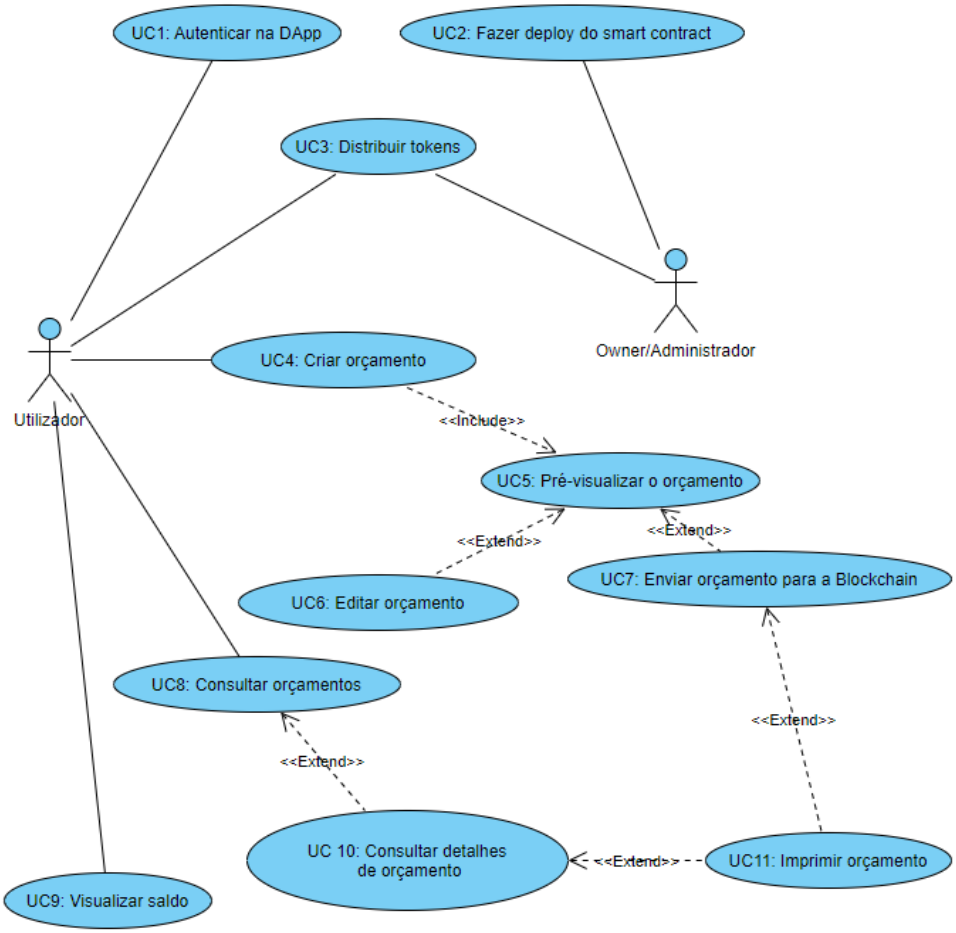


Figura 3.1 Diagrama de casos de uso da BudgetDApp

Do ponto de vista dos atores, observam-se as seguintes interações com a DApp:

- UC 1 Autenticar na DApp: Este caso de uso permite ao utilizador autenticar-se na aplicação para poder efetuar transações. Após inicializar sua carteira digital e efetuar o login, a aplicação reconhece o *token* EVD[1] e o utilizador poderá aceder às funcionalidades da aplicação.
- UC 2 Fazer *deploy* do *smart contract*: Este caso de uso permite ao utilizador responsável pela DApp disponibilizar o contrato na Blockchain.

[1] O *token* EVD é a cripto moeda desenvolvida no âmbito da *BudgetDApp*, apresentado mais à frente neste documento.

Coimbra Business School | ISCAC
Quinta Agrícola – Bencanta, 3045-601 Coimbra
Tel +351 239 802 000; E-mail: secretariado@iscac.pt; www.iscac.pt

- UC 3 Distribuir *token*: Este caso de uso começa após o caso anterior, onde o owner/administrador da DApp distribui *token*s aos demais utilizadores permitindo-lhes dessa forma utilizar a DApp.
- UC 4 Criar orçamento: Este caso de uso permite ao utilizador efetuar o preenchimento de orçamentos para envio a um determinado destinatário.
- UC 5 Pré-visualizar orçamento: Este caso de uso permite visualizar um orçamento preenchido antes de ser enviado para a Blockchain.
- UC 6 Editar orçamento: Este caso de uso permite ao utilizador, durante a visualização do orçamento, editar e alterá-lo de acordo com o que for necessário.
- UC 7 Enviar orçamento para Blockchain: Será permitido ao utilizador, no momento em que estiver visualizando um orçamento criado, finalizar o envio do documento à um destinatário.
- UC 8 Consultar orçamentos: Este caso de uso permite ao utilizador visualizar orçamentos criados anteriormente.
- UC 9 Visualizar saldo: Este caso de uso permite que o utilizador possa visualizar o saldo em sua carteira digital na página inicial sempre que desejar.
- UC 10: Consultar detalhes de orçamento: Este caso de uso permite ao utilizador ver os detalhes de um orçamento constante da lista de orçamentos apresentada no caso de uso *UC 8: Consultar orçamentos*.
- UC 11 Imprimir orçamento: Este caso de uso permite ao utilizador imprimir o orçamento que está a ser visualizado. Esta impressão poderá ser em formatos digitais ou físico.

3.1.2 Conceção

Após a análise inicial e levantamento inicial dos requisitos para a DApp passa-se à etapa de conceção onde se define qual a estrutura da informação a armazenar, aspeto visual da aplicação e a arquitetura física da mesma.

Coimbra Business School | ISCAC
Quinta Agrícola – Bencanta, 3045-601 Coimbra
Tel +351 239 802 000; E-mail: secretariado@iscac.pt; www.iscac.pt

3.1.2.1 Diagrama de classes

Para perceber um pouco da estrutura pretendida e as relações existentes dentro da aplicação, é fundamental a visão de um diagrama de classes conforme Figura 3.2.

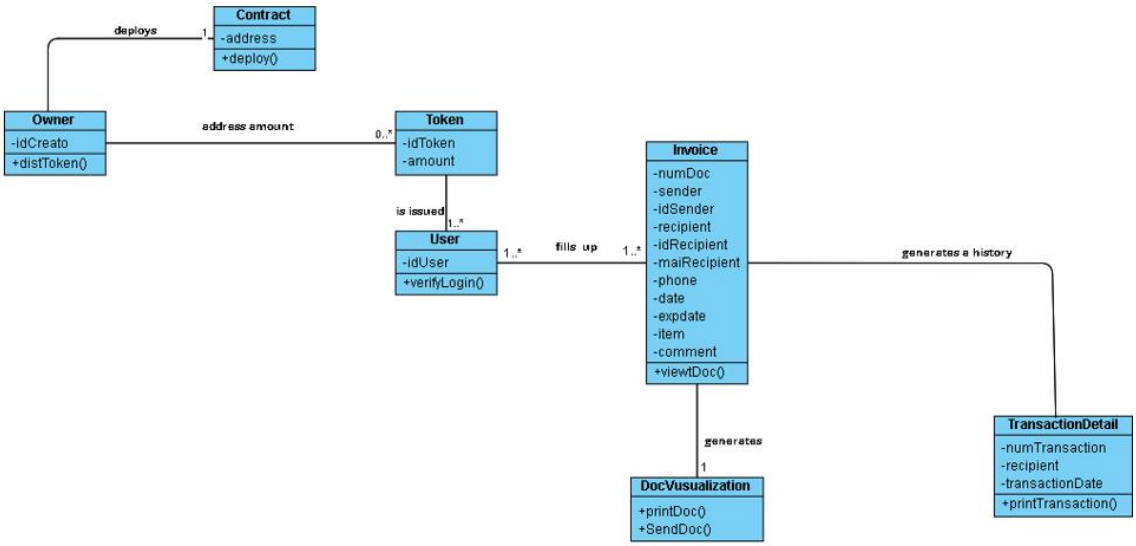


Figura 3.2 Diagrama de classes

Através do diagrama acima, podemos ver que o *owner* é o utilizador inicial que faz o *deploy* do contrato, sendo a partir feita a distribuição dos *token*s aos utilizadores. Uma vez recebidos os *token*s, os utilizadores podem começar a utilizar a DApp, com o preenchimento de um orçamento ou visualização, caso haja, verificar o histórico de transações. Através do preenchimento dos orçamentos (*invoices*), os utilizadores podem visualizar os mesmos, podendo utilizar as funções de impressão ou o envio de orçamentos.

Quando os orçamentos são enviados, geram um histórico com as transações que foram efetuadas pelo utilizador corrente. O histórico de transações, além de possuir informações gerais como destinatário e data de envio, também é possível aceder aos detalhes com todas as informações do momento de envio, possibilitando que os orçamentos já enviados possam ser reimpressos.

Coimbra Business School | ISCAC
Quinta Agrícola – Bencanta, 3045-601 Coimbra
Tel +351 239 802 000; E-mail: secretariado@iscac.pt; www.iscac.pt

3.1.2.2 Diagrama de objetos

Com a ideia de ter a perceção do ponto de vista do que um utilizador pode fazer, baseado no que foi demonstrado no diagrama de classes, apresenta-se o digrama de objetos na Figura 3.3.

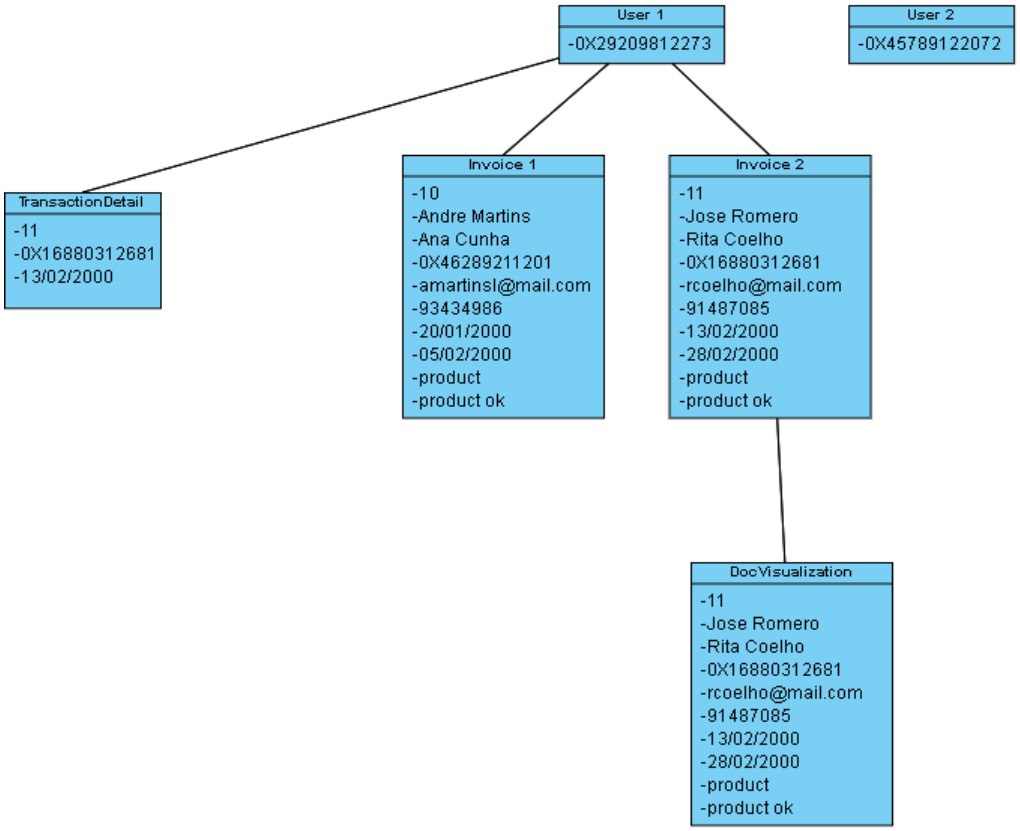


Figura 3.3 Diagrama de objetos

O sistema pode possuir vários utilizadores e os mesmos podem possuir ou não orçamentos preenchidos, além de poderem visualizar o histórico de orçamentos a qualquer momento.

3.1.2.3 Mockups

Os *mockups* permitem ter uma ideia visual de como apresentar as funcionalidades identificadas nos casos de uso propostos ao utilizador. De seguida apresentam-se os *mockups* propostos para a BudgetDApp.

Coimbra Business School | ISCAC
Quinta Agrícola – Bencanta, 3045-601 Coimbra
Tel +351 239 802 000; E-mail: secretariado@iscac.pt; www.iscac.pt

O primeiro *mockup* é o ecrã inicial da aplicação (ver Figura 3.4), designado de *"Home"*, que implementa o caso de uso *UC9: Visualizar saldo*. Este ecrã inicial da aplicação surge após o utilizador efetuar a autenticação em sua carteira digital. Terá o *token* reconhecido e a informação da sua conta e seu saldo estarão visíveis. Desta forma o utilizador terá certeza que poderá efetuar transações ou consultar suas transações.

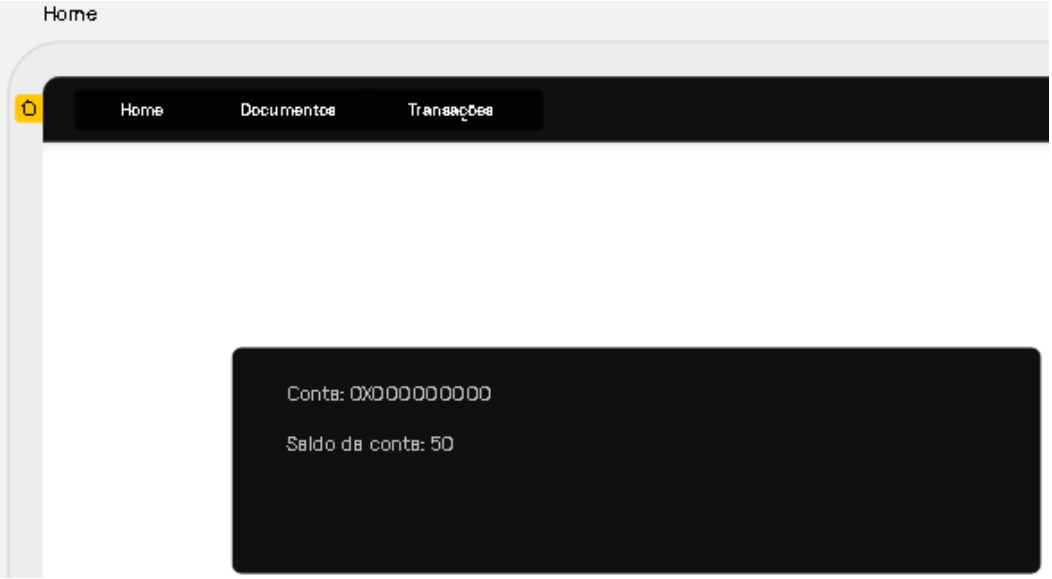


Figura 3.4 Mockup do ecrã relativo ao UC9: Visualizar saldo

Na Figura 3.5 apresenta-se o *mockup* do ecrã relativo ao *UC4: Criar orçamento* que permite ao utilizador criar os orçamentos que serão futuramente enviados para a Blockchain.

Coimbra Business School | ISCAC
Quinta Agrícola – Bencanta, 3045-601 Coimbra
Tel +351 239 802 000; E-mail: secretariado@iscac.pt; www.iscac.pt

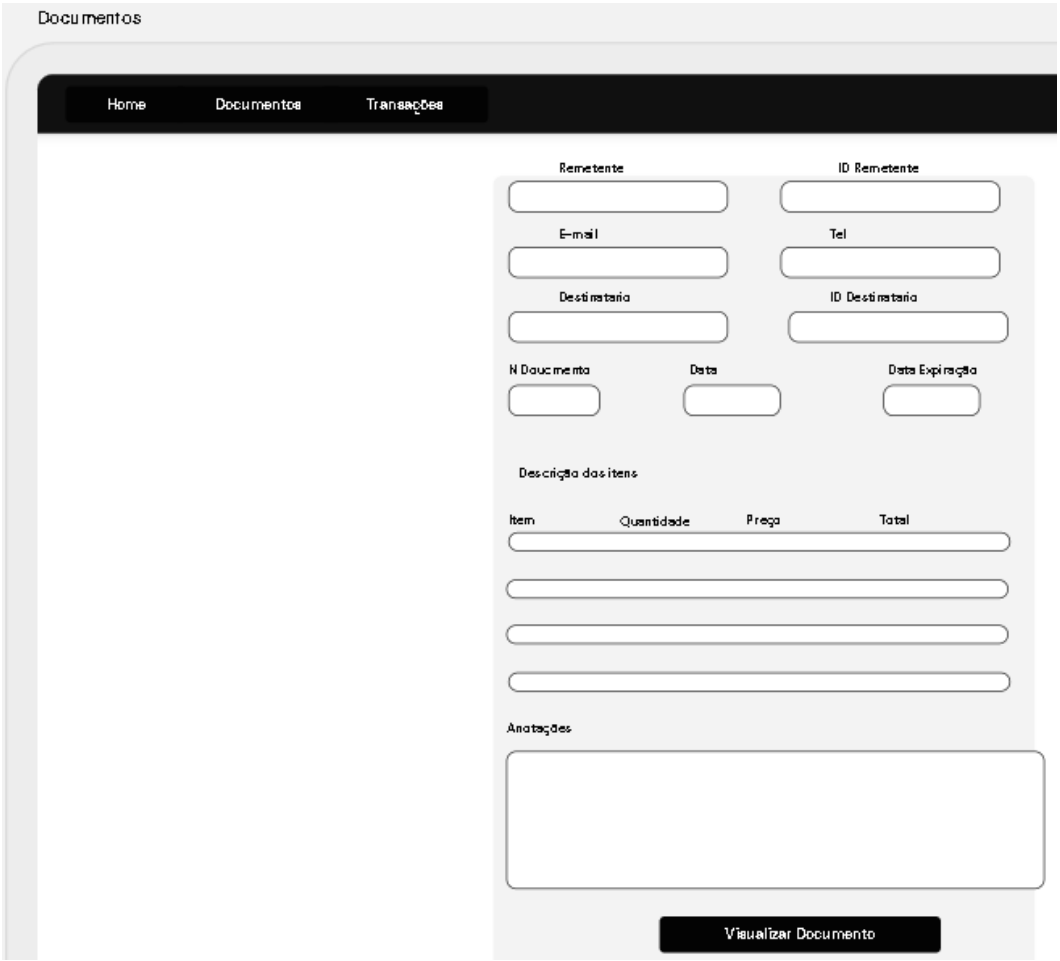


Figura 3.5 Mockup do ecrã relativo ao UC4: Criar orçamento

A Figura 3.6 apresenta o *mockup UC5: Pré-visualizar orçamento* que permite aceder às funcionalidades proporcionadas pelos casos de uso *UC6: Editar orçamento* e *UC7: Enviar orçamento*.

Coimbra Business School | ISCAC
Quinta Agrícola – Bencanta, 3045-601 Coimbra
Tel +351 239 802 000; E-mail: secretariado@iscac.pt; www.iscac.pt

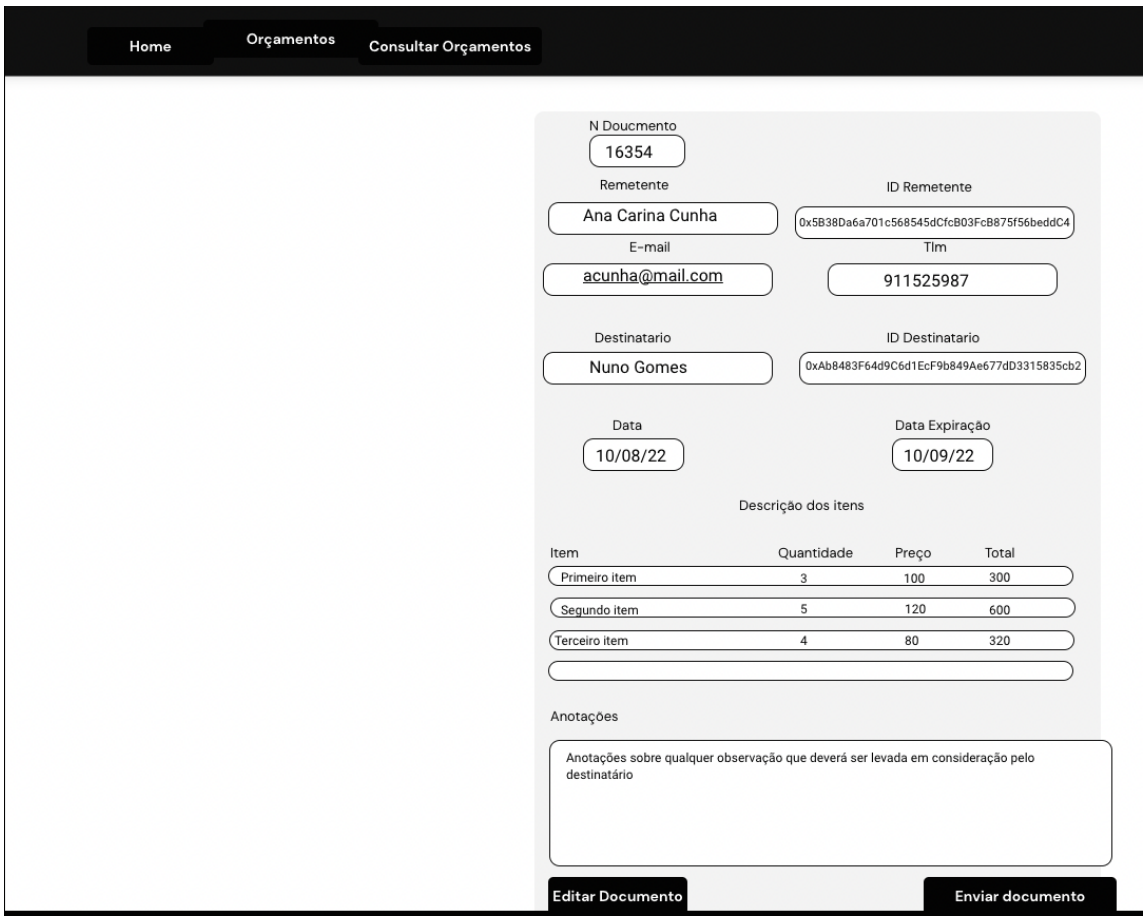


Figura 3.6 Mockup do UC5: Pré-visualizar orçamento

O envio dos orçamentos acontece de uma forma simples, onde o utilizador apenas preenche os dados do formulário, levando em consideração os campos obrigatórios e em seguida efetua o envio para o destinatário pretendido.

A Figura 3.7 refere-se ao caso de uso *UC8: Consultar orçamentos*, onde é apresentado ao utilizador o histórico dos orçamentos por si enviados. Desta forma, além de ter um rastreio com informações das transações efetuadas (orçamentos criados), será permitido ao utilizador aceder aos detalhes de cada transação (orçamento criado), que trará o orçamento enviado e com opções de impressão.

Coimbra Business School | ISCAC
Quinta Agrícola – Bencanta, 3045-601 Coimbra
Tel +351 239 802 000; E-mail: secretariado@iscac.pt; www.iscac.pt

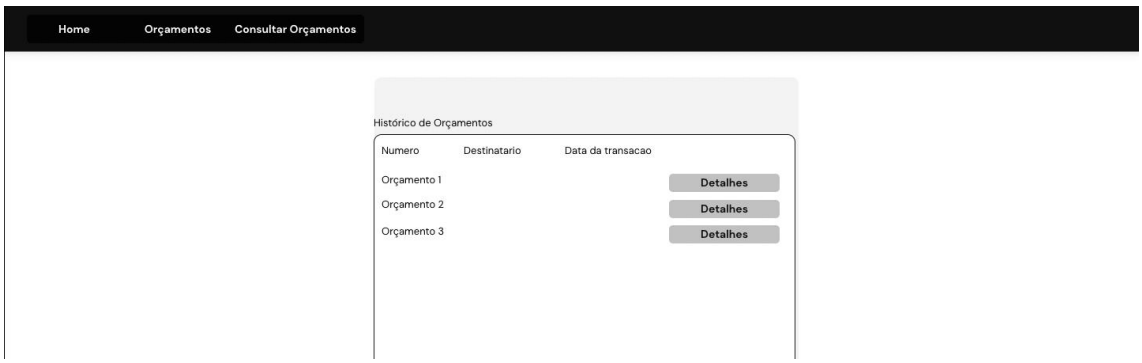


Figura 3.7 Mockup relativo ao UC8: Consultar orçamentos

A Figura 3.8 refere-se ao *mockup UC 10: Ver detalhes de orçamento*, onde será possível que o utilizador consulte as informações de orçamentos já enviados para a Blockchain.

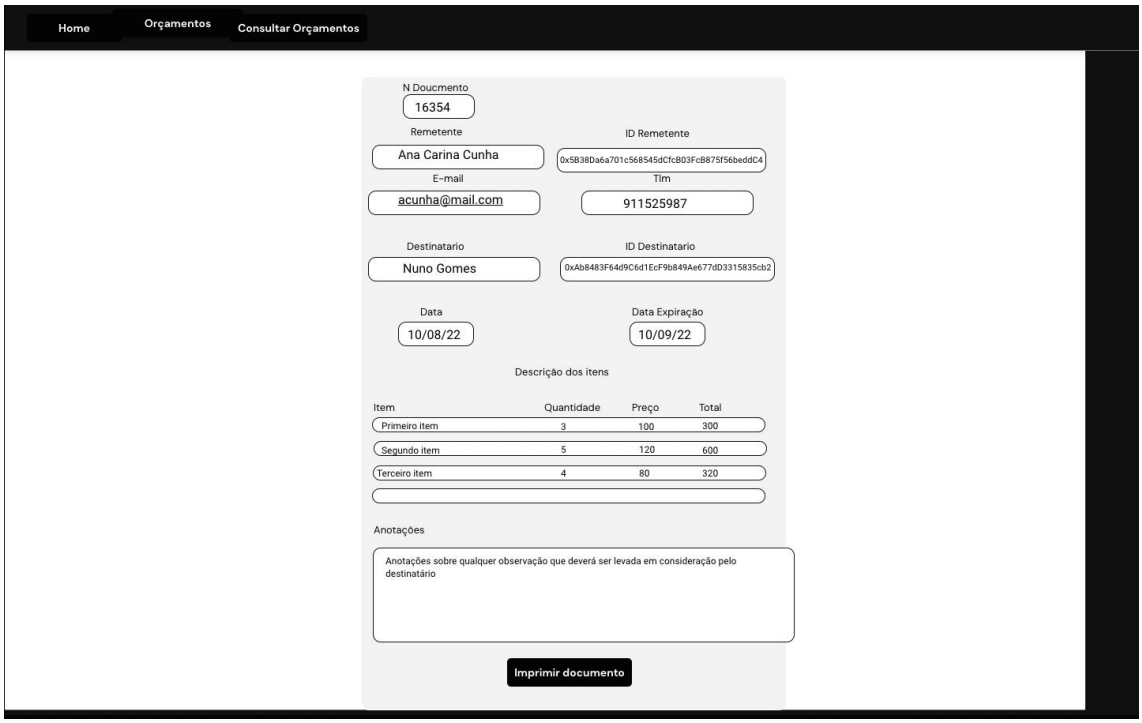


Figura 3.8 Mockup relativo ao caso de uso UC 10: Ver detalhes de orçamento

Como uma alternativa para uma consulta fora do sistema, é possível salvar os orçamentos em PDF ou imprimi-los, clicando no botão "Imprimir documento", visível na Figura 4.8. que irá despoletar o caso de uso *UC 10: Imprimir orçamento*. Embora objetivo da aplicação seja manter os orçamentos em um lugar no qual as partes envolvidas tenham certeza que é um documento único, faz-se necessária e justificável esta característica, uma vez que por vezes pode ser necessária a utilização do documento impresso.

Coimbra Business School | ISCAC
Quinta Agrícola – Bencanta, 3045-601 Coimbra
Tel +351 239 802 000; E-mail: secretariado@iscac.pt; www.iscac.pt

3.1.3 Arquitetura

A arquitetura da aplicação BudgetDApp reflete as duas partes em que a mesma se divide (*front-end* e *smart contract*) e que interagem entre si. Assim, para implementar o *front-end* será necessário um servidor Web, que funcionará numa máquina Linux e possuirá as bibliotecas necessárias para o seu funcionamento. Já no que diz respeito ao *backend* e o *smart contract* será utilizada a rede *Ethereum*, onde estarão armazenados os dados transacionados entre as partes envolvidas na utilização da DApp. A arquitetura proposta está exemplificada na Figura 3.9.

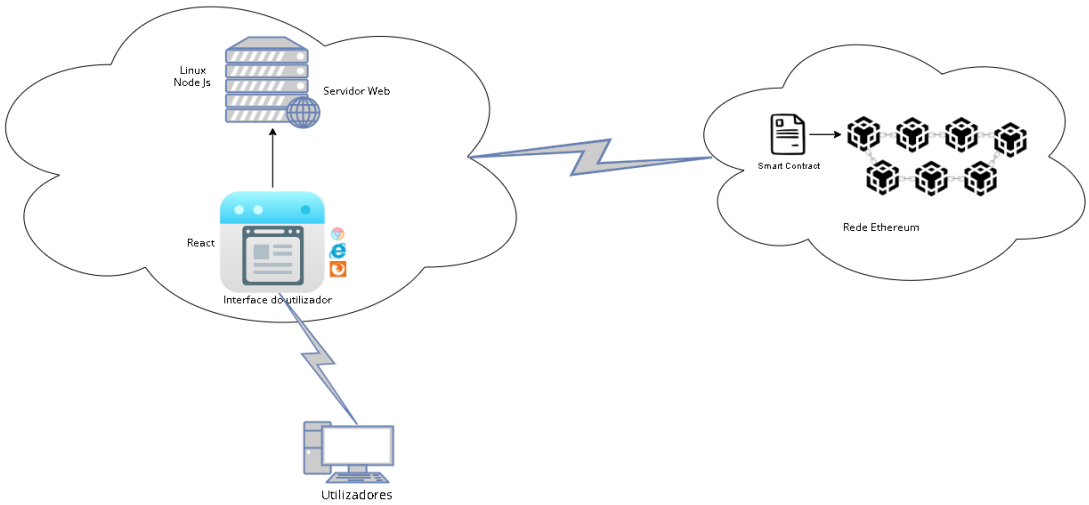


Figura 3.9 Arquitetura da 1ª interação da DApp

3.1.4 Implementação

Nesta secção apresentam-se as tecnologias utilizadas para o desenvolvimento da BudgetDApp e alguns pormenores da implementação da mesma, que, como referido anteriormente, é composta de duas partes distintas que interagem entre si: *smart contract* e *front-end*.

Coimbra Business School | ISCAC
Quinta Agrícola – Bencanta, 3045-601 Coimbra
Tel +351 239 802 000; E-mail: secretariado@iscac.pt; www.iscac.pt

3.1.4.1 Tecnologias utilizadas

Para a criação do ambiente de desenvolvimento da aplicação, foram utilizados os seguintes componentes:

- **Ubuntu 20.0.4**

O Ubuntu é um sistema operativo Linux baseado em licença *General Public License* (GPL), ou seja, uma distribuição com código aberto. A escolha deste sistema operativo foi feita pela estabilidade e melhor funcionamento de aplicações e serviços neste tipo de ambiente.

- **Truffle e Ganache**

Ferramentas utilizadas para simular um ambiente de desenvolvimento e testes Blockchain. O *Ganache* e *Truffle* são ferramentas para auxiliar na simulação de um ambiente real e evitar a necessidade de publicar o contrato, uma vez que cada contrato é único e quando publicado em uma rede Blockchain não pode ser alterado.

- **React ou ReactJS**

É uma biblioteca de código aberta baseada em *JavaScript*, largamente utilizada no desenvolvimento de interfaces de interação com o utilizador. A escolha desta biblioteca foi baseada na ampla documentação disponível e por proporcionar maior agilidade no desenvolvimento da DApp desejada.

- **Metamask**

O metamask é uma carteira digital gratuita utilizada para interação com a Blockchain e *tokens*. Com esta carteira digital, é possível gerir seus *token*s de uma forma simples, facilitando a interação com as contas e a aplicação

3.1.4.2 Smart contracts

Conforme dito anteriormente, a aplicação terá um *token* , que é também ele defindo através de um *smart contract*, e o *smart contract* referente às regras de negócio imaginadas para a aplicação.

Coimbra Business School | ISCAC
Quinta Agrícola – Bencanta, 3045-601 Coimbra
Tel +351 239 802 000; E-mail: secretariado@iscac.pt; www.iscac.pt

Smart contract do token EVDCoin

Como o projeto apresenta como característica a implementação final em uma rede Blockchain pública, pretende-se utilizar um *token* próprio para restringir a utilização da aplicação aos detentores desse *token* público. Para a criação deste *token*, alguns modelos pré-definidos de regras de negócio, que são *tokens* de referência para o *token* principal serão utilizados. Este *token* será chamado de *EVDCoin*, uma abreviação criada a partir do nome da aplicação (originalmente designada EnviaDocs) e a palavra moeda em inglês (Coin).

Para o desenvolvimento do *token*, alguns contratos de dependência foram utilizados. Esses contratos são públicos e encontrados em (Openzeppelin, 2021).

```
// SPDX-License-Identifier: MIT
pragma solidity >=0.4.16 <0.9.0;
// SPDX-License-Identifier: MIT
pragma solidity >=0.4.16 <0.9.0;

import "./ERC20.sol";
import "./ERC20Detailed.sol";

contract TokenEnviaDocs is ERC20, ERC20Detailed {
    address public admin;
    constructor() ERC20Detailed("EvdCoin", "EVD", 18) public {
        uint256 initialSupply = 10000 * 10 ** 18;
        _mint(msg.sender, initialSupply);
        admin = msg.sender;
    }
    function mint (address to, uint amount) external {
        require(msg.sender == admin, 'Apenas admin');
        _mint (to, amount);
    }
}
```

Figura 3.10 Código do contrato de criação do token EVDCoin

Neste contrato de criação do *token* (ver Figura 3.10), podemos observar o nome do *token* criado e o símbolo que será representado quando o mesmo for utilizado. Para além disto, o contrato define o valor inicial do *token* e limita a transferência inicial apenas ao criador ou administrador inicial.

Coimbra Business School | ISCAC
Quinta Agrícola – Bencanta, 3045-601 Coimbra
Tel +351 239 802 000; E-mail: secretariado@iscac.pt; www.iscac.pt

Smart contract da aplicação

O contrato base ainda está com uma estrutura que certamente irá passar por algumas mudanças e melhorias, de acordo com as novas funcionalidades do *front-end*, sem perder a ideia inicial.

```
// SPDX-License-Identifier: GPL-3.0
pragma solidity >=0.4.16 <0.9.0;

contract enviaDocs {
    struct Dados {
        uint ident;
        string remet;
       string idRemet;
       string mail;
       string numCont;
       string destidDest;
       string dataInidataFim;
    }
    Dados[] dados;
    uint proxval; // default value 0, add public to see the value
    function dadosForm(
        string memory _numCont,
        string memory _remet,
        string memory _idRemet,
        string memory _mail,
        string memory _dest_idDest,
        string memory _dataIni_dataFim
        //string memory
        ) public {
                dados.push(Dados(proxval,   _numCont,_remet,   _idRemet,
_mail,  _dest_idDest, _dataIni_dataFim));
        proxval++;
    }
    function encontraDados(uint _ident) internal view returns (uint) {
```

Figura 3.11 Extrato do código do smart contract da aplicação

A Figura 3.11 mostra trecho do código do contrato base da aplicação, com as estruturas pensadas para comportar os dados dos utilizadores.

3.1.4.3 Front-end

O desenvolvimento de uma interface amigável para interação com o *smart contract* é importante, pois de outra forma o utilizador não conseguirá interagir com o *smart*

Coimbra Business School | ISCAC
Quinta Agrícola – Bencanta, 3045-601 Coimbra
Tel +351 239 802 000; E-mail: secretariado@iscac.pt; www.iscac.pt

contract. A ideia inicial tem como base manter uma interface de utilizador simples, funcional e ágil.

A página inicial (*homepage*) da BudgetDApp conforme visto na Figura 3.12 apresenta informações relativas ao utilizador como a conta e saldo da sua carteira digital.

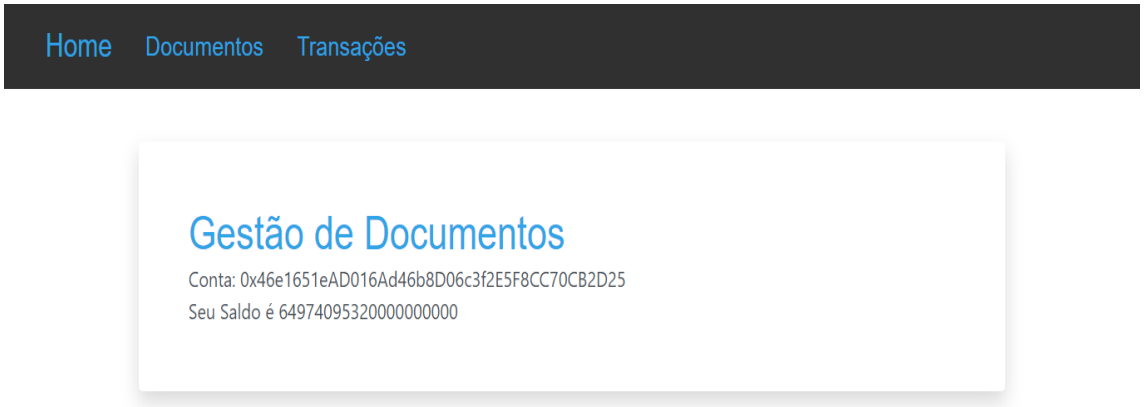


Figura 3.12 Página Home relativa ao UC 9 Visualizar saldo

O utilizador estará devidamente identificado quando tiver suas credenciais devidamente colocadas em sua carteira digital com o *token* específico requerido pela aplicação. A Figura 3.13 mostra a função para conectar à carteira digital.

```
const [defineBalance, setDefineBalance] = useState();

  useEffect(() => {
    async function load() {
      const web3 = new Web3(Web3.givenProvider ||
'http://localhost:7545');
      const accounts = await web3.eth.requestAccounts();

      setAccount(accounts[0]);
      console.log(accounts);
      account2 = (accounts[0]);
      const balance = await web3.eth.getBalance(account2);
       setDefineBalance(balance);

    }

    load();
  }, []);
```

Figura 3.13 Front-end conexão com carteira digital

Após a autenticação, o utilizador terá o retorno exibido na figura acima e terá informações sobre sua conta e seu saldo, mostrados no código abaixo na Figura 3.14:

Coimbra Business School | ISCAC
Quinta Agrícola – Bencanta, 3045-601 Coimbra
Tel +351 239 802 000; E-mail: secretariado@iscac.pt; www.iscac.pt

```
return (
      <>
  <main className="m-5 p-5 bg-white rounded shadow">
     <>
       <Typography variant="h4" nowrap>
           Gestão de Documentos
       </Typography>
     </>
          <>Conta: {account}</>
          <br/>
          <>Seu Saldo é {defineBalance}</>
  </main>

  </>

 );

}
```

Figura 3.14 Código de implementação da homepage

O ecrã da Figura 3.15permite criar os orçamentos que serão futuramente enviados para a Blockchain.

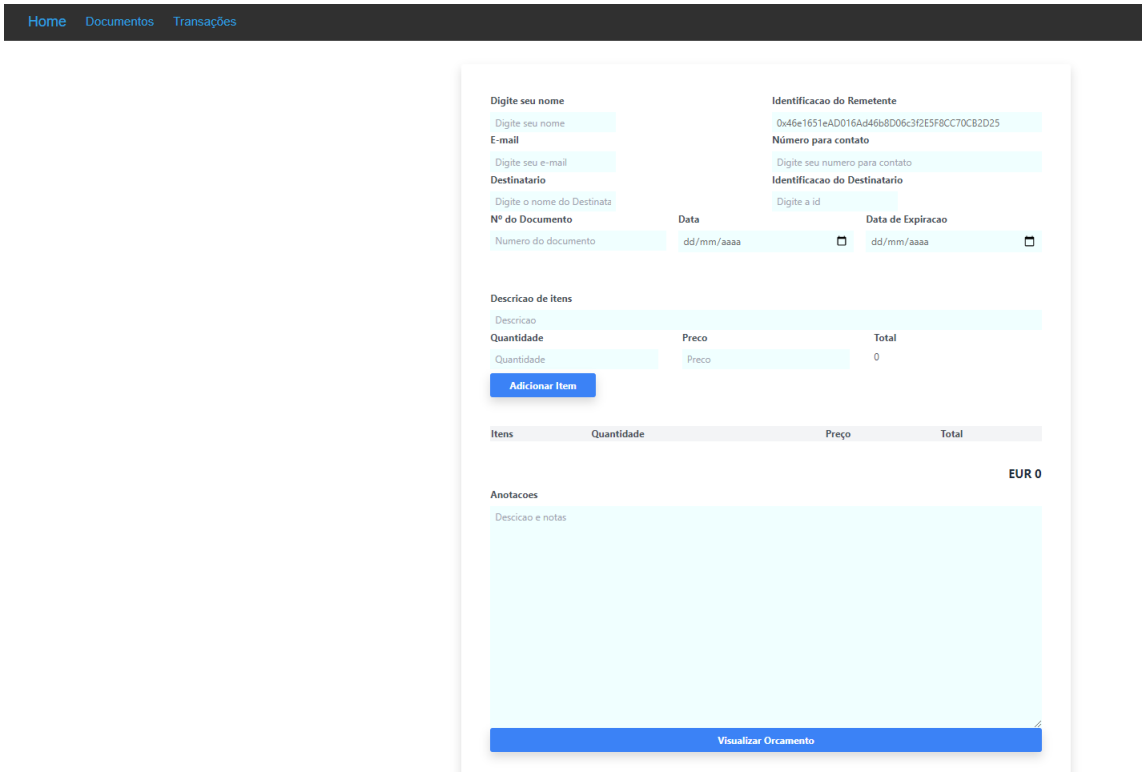


Figura 3.15 Formulário criar Orçamento relativo ao UC 4 Criar orçamento

Coimbra Business School | ISCAC
Quinta Agrícola – Bencanta, 3045-601 Coimbra
Tel +351 239 802 000; E-mail: secretariado@iscac.pt; www.iscac.pt

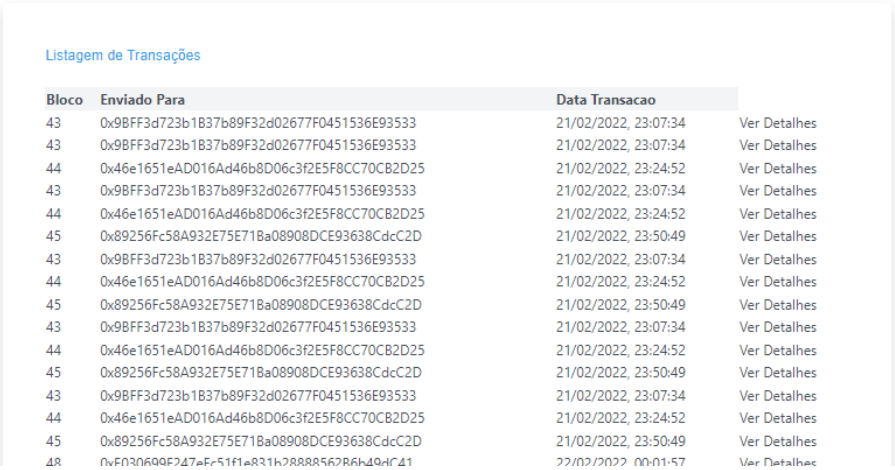
Na Figura 3.16 apresenta-se um extrato do código desenvolvido em *React* de suporte à criação do formulário apresentado na Figura 3.15 e respetivas interações.

```
      {showInvoice ? (
      <>
      <ReactToPrint trigger={() => <button className="bg-red-500 mb-5
text-white font-bold py-2 px-8 rounded shadow boerder-2 border-red-500
hover:bg-transparent hover:text-red-500 transition-all durantion-
300">Imprimir / Download</button>}
       content={() => componentRef.current}/>

       <div ref={componentRef} className="p-5">

       <Header handlePrint={handlePrint} />

       <MainDetails remetente={remetente}
telcont={telcontato}/>

       <ClientoDetails nomecliente={destinatario}
clienteid={iddestnatario}/>

       <Dates numdoc={numerodocumento} datadoc={datadocumento}
dataexp={dataexpiracao}/>

       ): (
           <>
```

Figura 3.16 Código de implementação do formulário de criação de orçamento respeitante ao UC8: Consultar orçamentos

A Figura 3.17 refere-se à página que faz a listagem de todas as transações efetuadas (orçamentos criados e enviados para a Blockchain) pelo utilizador corrente. Para além disso, será possível ver os detalhes de cada transação, ou seja, ver o orçamento que foi enviado para a Blockchain.

Coimbra Business School | ISCAC
Quinta Agrícola – Bencanta, 3045-601 Coimbra
Tel +351 239 802 000; E-mail: secretariado@iscac.pt; www.iscac.pt



Listagem de Transações

Bloco	Enviado Para	Data Transacao	
43	0x9BFF3d723b1B37b89F32d02677F0451536E93533	21/02/2022, 23:07:34	Ver Detalhes
43	0x9BFF3d723b1B37b89F32d02677F0451536E93533	21/02/2022, 23:07:34	Ver Detalhes
44	0x46e1651eAD016Ad46b8D06c3f2E5F8CC70CB2D25	21/02/2022, 23:24:52	Ver Detalhes
43	0x9BFF3d723b1B37b89F32d02677F0451536E93533	21/02/2022, 23:07:34	Ver Detalhes
44	0x46e1651eAD016Ad46b8D06c3f2E5F8CC70CB2D25	21/02/2022, 23:24:52	Ver Detalhes
45	0x89256Fc58A932E75E71Ba08908DCE93638CdcC2D	21/02/2022, 23:50:49	Ver Detalhes
43	0x9BFF3d723b1B37b89F32d02677F0451536E93533	21/02/2022, 23:07:34	Ver Detalhes
44	0x46e1651eAD016Ad46b8D06c3f2E5F8CC70CB2D25	21/02/2022, 23:24:52	Ver Detalhes
45	0x89256Fc58A932E75E71Ba08908DCE93638CdcC2D	21/02/2022, 23:50:49	Ver Detalhes
43	0x9BFF3d723b1B37b89F32d02677F0451536E93533	21/02/2022, 23:07:34	Ver Detalhes
44	0x46e1651eAD016Ad46b8D06c3f2E5F8CC70CB2D25	21/02/2022, 23:24:52	Ver Detalhes
45	0x89256Fc58A932E75E71Ba08908DCE93638CdcC2D	21/02/2022, 23:50:49	Ver Detalhes
43	0x9BFF3d723b1B37b89F32d02677F0451536E93533	21/02/2022, 23:07:34	Ver Detalhes
44	0x46e1651eAD016Ad46b8D06c3f2E5F8CC70CB2D25	21/02/2022, 23:24:52	Ver Detalhes
45	0x89256Fc58A932E75E71Ba08908DCE93638CdcC2D	21/02/2022, 23:50:49	Ver Detalhes
48	0xF030699F247eFc51f1e831b28888562B6b49dC41	22/02/2022, 00:01:57	Ver Detalhes

Figura 3.17 Página relativa ao UC 8 Consultar orçamentos

Apresenta-se na Figura 3.18 o código relativo à página da Figura 3.17 que permite consultar as informações contidas numa transação existente num bloco.

```
const getdata = async() => {
    const latest = await web3.eth.getBlockNumber();

    let lstTrans = [];
    for(let i = 0; i <= latest; i++ ) {
      const block = await web3.eth.getBlock(i);
      if (block && block.transactions) {
        for (let tx of block.transactions) {
          let transaction = await
web3.eth.getTransaction(tx);
          let data = web3.utils.hexToAscii('0x' +
transaction.input.slice(138, transaction.input.length))
          if (account2[0] === transaction.from && transaction.to !==
null) {
            var date = new Date(block.timestamp*1000);
            let dates = new Intl.DateTimeFormat('pt-PT', { year:
'numeric', month: '2-digit', day: '2-digit', hour: '2-digit', minute: '2-
digit', second: '2-digit' }).format(date)
            lstTrans.push({
              Bloco: i,
              To: transaction.to,
              data: data,
              datedoc: dates
            });
          }
        }
```

Figura 3.18 Código de implementação do UC 10: Consultar detalhes de orçamento

3.1.4.4 Limitações e dificuldades

Durante o desenvolvimento do projeto, foram detetadas algumas dificuldades que poderiam comprometer a entrega da aplicação dentro do prazo esperado. Essas

Coimbra Business School | ISCAC
Quinta Agrícola – Bencanta, 3045-601 Coimbra
Tel +351 239 802 000; E-mail: secretariado@iscac.pt; www.iscac.pt

dificuldades tiveram diferentes origens que originou o processo de reformulação da aplicação, conforme listadas a seguir:

- Utilização de um *token* próprio – Como demonstrado anteriormente (ver secção 4.1.4.2.1) foi criado um *token* próprio para utilizar na aplicação, o EVDCoin. No entanto, não foi possível utilizar este *token* com a aplicação desenvolvida, pelo que se teve desistir desta primeira implementação. Tal deveu-se à falta de documentação sobre a utilização de *tokens* próprios em DApp.
- Consumo de GAS dentro da rede *Ethereum* – A DApp desenvolvida guarda todas as informações relativas ao orçamento na rede *Ethereum* algo que é extremamente dispendioso, o que torna economicamente inviável a sua colocação em produção.

3.2 Segunda iteração

A mudança descrita anteriormente resultou na reestruturação da BudgetDApp para adequação aos problemas encontrados. A segunda iteração da aplicação apresenta uma interface similar e com a mesma ideia inicial, porém algumas funcionalidades foram retiradas com o objetivo de simplificar o protótipo e outras adicionadas como forma de ultrapassar as dificuldades.

3.2.1 Análise e enumeração de requisitos

No seguimento do processo, será apresentado apenas o que foi modificado em relação à primeira iteração.

3.2.1.1 Casos de uso

Nesta segunda versão dos casos de uso, a estrutura permanece com a interação de dois atores como visto na primeira. Entretanto, é percetível a maior interação do administrador e dono do contrato, o que reflete as alterações realizadas na segunda iteração, conforme se apresenta na Figura 3.19.

Coimbra Business School | ISCAC
Quinta Agrícola – Bencanta, 3045-601 Coimbra
Tel +351 239 802 000; E-mail: secretariado@iscac.pt; www.iscac.pt

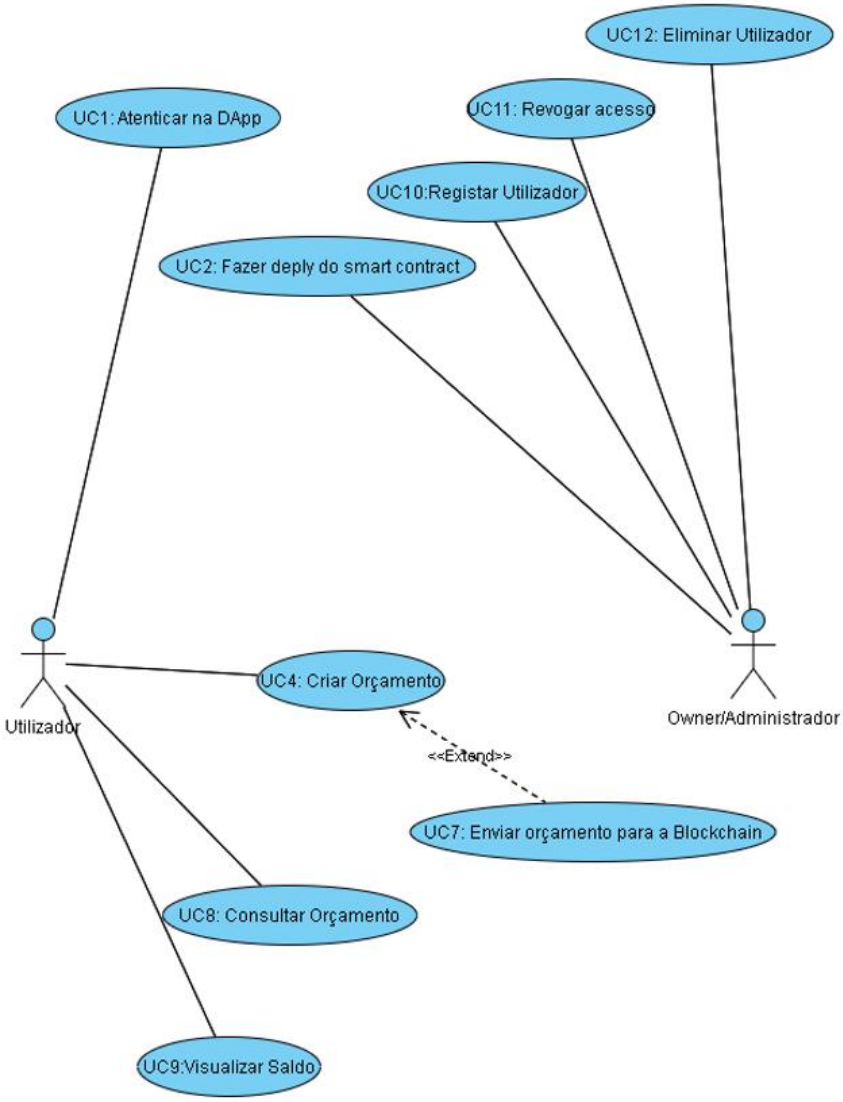


Figura 3.19 Novo diagrama de casos de uso da BudgetDApp

Segue abaixo uma descrição dos novos casos de uso da aplicação BudgetDApp, refletindo as alterações em relação a primeira iteração:

- UC 1 Autenticar na DApp: Este caso de uso permite ao utilizador autenticar-se na aplicação para poder efetuar transações.
- UC 2 Fazer *deploy* do *smart contract*: Este caso de uso parte do utilizador que irá fazer o *deploy* do contrato na Blockchain. Este utilizador será o gestor da DApp, pois ficarão com ele as permissões de adição e edição de utilizadores.
- UC 4 Criar orçamentos – Este caso permite que o utilizador efetuar o preenchimento de orçamentos para envio a um determinado destinatário.

Coimbra Business School | ISCAC
Quinta Agrícola – Bencanta, 3045-601 Coimbra
Tel +351 239 802 000; E-mail: secretariado@iscac.pt; www.iscac.pt

- UC 7 Enviar orçamento para Blockchain: Será permitido ao utilizador, no momento em que estiver visualizando um orçamento criado, finalizar o envio do documento a um destinatário.
- UC 8 Consultar orçamentos – Este caso de uso permite ao utilizador visualizar orçamentos criados anteriormente.
- UC 9 Visualiza saldo – Este caso permite que o utilizador possa visualizar o saldo em sua carteira digital na página inicial sempre que desejar.
- UC 10 Registar utilizador – O administrador e criador do contrato podem criar utilizadores na DApp, permitindo assim acesso às funcionalidades da mesma.
- UC 11 Revogar acesso – Este caso permite com que o administrador possa suspender o acesso de um utilizador qualquer.
- UC 12 Eliminar utilizador – O administrador pode eliminar um utilizador da aplicação e impedir definitivamente seu acesso.

3.2.2 Conceção

Na conceção foi alterada a arquitetura proposta, que agora irá ter uma base de dados, como se apresenta de seguida.

3.2.2.1 Arquitetura

O projeto inicial envolvia dois componentes básicos. O primeiro deles era o *smart contract* com um código relativamente simples desenvolvido na linguagem de programação *Solidity* e um *front-end* desenvolvido em *React* para a interação com o contrato. Inicialmente, não seria necessário nenhum tipo de autenticação extra, pois o projeto contemplava a utilização de um *token* próprio que serviria como um filtro para ter somente os utilizadores desejados a manipular o contrato e as funções por ele disponibilizadas, ou seja, só utilizadores detentores do *token* (após autenticar nas suas carteiras digitais) é que poderiam interagir com o contrato.

Na nova arquitetura proposta procura-se resolver duas questões associadas à limitação relacionada com os custos de armazenamento de dados na Blockchain.

Os projetos baseados em Blockchain precisam ser simples e armazenar apenas informações importantes e relevantes para que fiquem registados na rede descentralizada.

Coimbra Business School | ISCAC
Quinta Agrícola – Bencanta, 3045-601 Coimbra
Tel +351 239 802 000; E-mail: secretariado@iscac.pt; www.iscac.pt

Apesar do conhecimento desta premissa, foi julgado que 10 campos eram necessários para serem armazenados na utilização do contrato. Lembrando que a aplicação proposta baseia-se no envio de orçamentos, a ideia de pretender que as informações estivessem todas armazenadas na Blockchain não estava incorreta. Entretanto, após diversos problemas com retornos das variáveis, sabendo que, para determinadas consultas que envolvam por exemplo algum tipo de alteração ou até mesmo para utilização de variáveis globais envolvem custos, o armazenamento de todas essas informações passou a não ter muito sentido.

A outra questão envolve também os custos, mas num sentido de entender como a rede *Ethereum* utiliza seu *storage* interno. Nesta rede, cada variável de acesso público ocupa um espaço em memória e para acesso aos mesmos e também para cada transação na Blockchain está associada a cobrança de uma taxa chamada GAS. No caso do primeiro *smart contract* desenvolvido, para além de diversas variáveis e funções, também dificultou por conta da utilização de vários *arrays* para desempenhar o funcionamento esperado. Tendo isto como parâmetro e após diversos erros de "*Out of GAS*" por utilizar demasiadas variáveis com profundidades diferentes, foi necessária uma adaptação para que a DApp prosseguisse em funcionamento. Vale ressaltar que, por conta de maior agilidade, o *smart contract* foi desenvolvido utilizando o *IDE Remix*, que não apresentava tais erros até a publicação do contrato juntamente com o *front-end* utilizando o *Truffle*.

Pensando em simplificar as informações no contrato e mantendo a segurança de alguma forma, foi definido que seria melhor alterar a estrutura, utilizando uma arquitetura mista onde parte dos dados seriam armazenados em uma base de dados local e outra parte na Blockchain. Esta divisão foi pensada visto que uma parte das informações do contrato eram informações estáticas, ou seja, informações sobre o utilizador que estariam sendo sempre repetidas e armazenadas na Blockchain a utilizar recursos, conforme visto na Figura 3.20.

Coimbra Business School | ISCAC
Quinta Agrícola – Bencanta, 3045-601 Coimbra
Tel +351 239 802 000; E-mail: secretariado@iscac.pt; www.iscac.pt

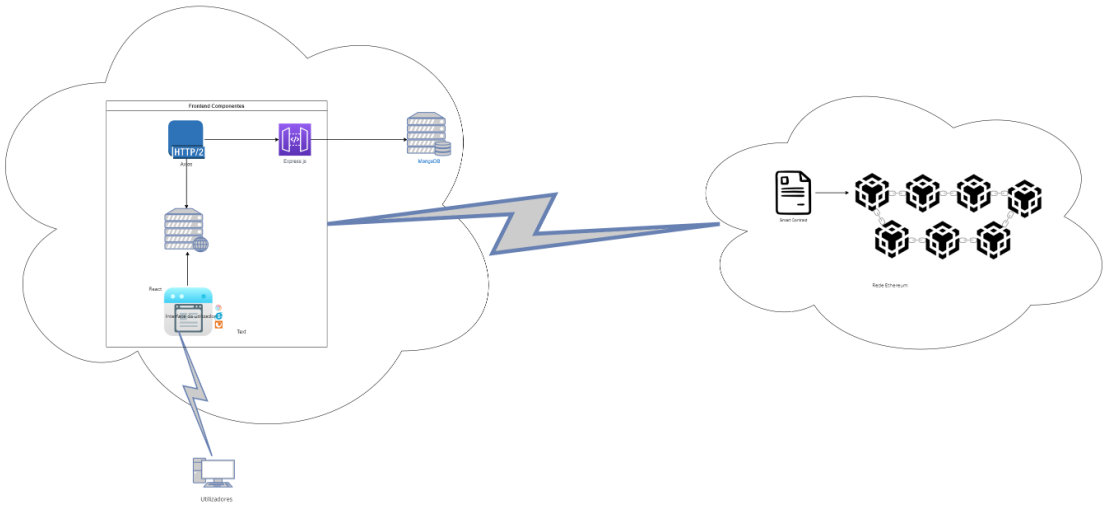


Figura 3.20 Nova arquitetura da BudgetDApp

3.2.3 Implementação

Para além do *smart contract* e do *front-end*, esta segunda iteração como apresentada na nova arquitetura (ver Figura 3.20) tem um novo componente que é o *back-end. Este back-end* possui uma base de dados e uma *Representational State Transfer Application Programmable Interface (REST API)* para comunicação da aplicação baseada no *smart contract*, a BudgetDApp e a base de dados propriamente dita.

3.2.3.1 Tecnologias utilizadas

Para além das tecnologias utilizadas no desenvolvimento do primeiro protótipo o segundo protótipo necessitou de utilizar novas tecnologias, a saber:

- **MongoDB**

O *MongoDB* é um sistema gestor de base dados baseado em código aberto escrito em C++. A escolha desta base de dados foi feita pela fácil integração com a ferramenta utilizada para REST API e simplicidade de instalação, configuração e utilização

- **Express.js**

O *Express.js* é um *framework* desenvolvido em *Node.js* que fornece estrutura para um servidor *web*, que possui uma configuração relativamente simples e segura e rápida. Utilizando esta ferramenta dinâmica e flexível criou-se um modelo de dados a partir do qual foram geradas as tabelas na base de dados.

Coimbra Business School | ISCAC
Quinta Agrícola – Bencanta, 3045-601 Coimbra
Tel +351 239 802 000; E-mail: secretariado@iscac.pt; www.iscac.pt

3.2.3.2 Smart contract

Independente de ser uma aplicação relativamente simples, implementar a ideia em um ambiente descentralizado envolve desafios no sentido de ultrapassar algumas limitações e conceitos aplicados na rede *Ethereum*. Por vezes, a estrutura do *front-end* e o que é pretendido neste componente influencia diretamente o desenvolvimento do contrato. No caso do projeto em questão, o *smart contract* teve de passar por diversas modificações até chegar a um estágio aceitável em termos de funcionalidade e que não violasse qualquer tipo de regra.

A versão atual do *smart contract* conta com uma estrutura de validação de utilizadores e utiliza um esquema de variáveis internas com o recebimento de outras externas para reduzir questões relacionadas a custos de armazenamento interno das variáveis na Blockchain. A parte de validação de utilizadores é uma forma de acrescentar uma camada de segurança, fazendo com que a aplicação seja acedida somente por utilizadores previamente liberados para o acesso. Cada função de alteração no contrato, que diz respeito ao utilizador (ver Figura 3.21), possui validadores para conferir se os utilizadores têm permissão para acesso. Outra forma de camada de segurança é que, para as funções de gestão de utilizadores como criação, revogação de acesso e exclusão do utilizador, o acesso é restrito ao criador do contrato, ou seja, a pessoa que fará a publicação do contrato é a única com este tipo de permissão.

```
37  //funcao interna para atribuir permissão ao utilizador para utilizar o contrato
38      function userPermission(address _account) public onlyOwner {
39          userAccess[_account] = true;
40          emit GrantPermission( _account);
41      }
42  //Funcao receber a permissão e gravar na funçao interna
43      function grantPermission(address _account) external onlyOwner {
44          userPermission(_account);
45      }
46  //Funçao para remover o acesso do utilizador
47        function revokePermission(address _account) public onlyOwner {
48          userAccess[_account] = false;
49          emit RevokePermission(_account);
50      }
```

Figura 3.21 Funções do smart contract de gestão de utilizadores

Podemos observar na Figura 3.21, na função *userPermission* a forma simples com que é atribuído acesso a um novo utilizador, através de uma *flag* com verdadeiro ou falso. De

Coimbra Business School | ISCAC
Quinta Agrícola – Bencanta, 3045-601 Coimbra
Tel +351 239 802 000; E-mail: secretariado@iscac.pt; www.iscac.pt

forma contrária, a *revokePermission* retira esta *flag* e impede que o utilizador siga com acesso.

As funções apresentadas na Figura 3.22 permitem o envio das informações relativas ao orçamento a guardar na Blockchain, designadamente o ID do remetente, o nome e ID do destinatário.

```
61      //função para envio do cabeçalho do form
62      function sendHForm(
63          address _hSender,
64          address _hIdDest,
65          string memory _hDestName
66      ) internal returns (uint256[] memory) {
67          IDdoc++;
68          if (userAccess[msg.sender] == true) {
69              FormHeader.push(formHeader(IDdoc, _hSender, _hIdDest, _hDestName));
70              headerID.push(IDdoc);
71              return (headerID);
72          } else {
73              revert();
74          }
75      }
76
77      function setSendHForm(
78          address _hSender,
79          address _hIdDest,
80          string memory _hDestName
81      ) external {
82          sendHForm(_hSender, _hIdDest, _hDestName);
83      }
```

Figura 3.22 Função do smart contract para envio do orçamento

Como é possível observar a partir da linha 68 (ver Figura 3.22), a validação do utilizador ocorre, verificando se o endereço do utilizador está com valor *true*. Caso esteja o utilizador prossegue com o envio, caso não esteja a transação é revertida. Em caso de seguimento, os dados são enviados obedecendo a estrutura da *struct*.

Para além do envio do orçamento para a Blockchain também existe a função de consulta / retorno de um orçamento da Blockchain, que se apresenta na Figura 3.23.

Coimbra Business School | ISCAC
Quinta Agrícola – Bencanta, 3045-601 Coimbra
Tel +351 239 802 000; E-mail: secretariado@iscac.pt; www.iscac.pt

```
145     function getDocDetail(uint256 _docID)
146         public
147         view
148         returns (
149             uint256,
150             address,
151             string memory,
152             string memory,
153             string memory
154         )
155     {
156         uint256 myID;
157         string memory _uName;
158         address _hIdDest;
159         string memory _hDestName;
160         //string memory _hExpDate;
161         string memory _mItems;
162         string memory _mDate;
163 
164         //address _mAddress = msg.sender;
165         //bool resultCreate = false;
166 
167         for (uint256 i = 0; i < FormHeader.length; i++) {
168             if (FormHeader[i].hID == _docID) {
169                 //resultCreate = true;
170                 _hIdDest = FormHeader[i].hIdDest;
171                 _hDestName = FormHeader[i].hDestName;
172                 //_hExpDate = FormHeader[i].hExpDate;
173                 for (uint256 i = 0; i < MDetail.length; i++) {
174                     if (MDetail[i].mID == FormHeader[i].hID) {
175                         _mItems = MDetail[i].mItems;
176                         _mDate = MDetail[i].mDate;
177                     }
178                 }
179             }
180         }
181 
182         return (IDdoc, _hIdDest, _hDestName, _mItems, _mDate);
183     }
```

Figura 3.23 Função do smart contract para consulta do orçamento na Blockchain

3.2.3.3 Back-end

Nesta segunda iteração da BudgetDApp como referido anteriormente decidiu-se utilizar uma base de dados local, baseada no MongoDB, com dois objetivos: 1) guardar os dados relativos aos orçamentos que ficaria dispendioso guardar na Blockchain e 2) guardar informação sensível dos utilizadores que de outra forma teriam de ficar públicos na Blockchain. Não obstante, as *passwords* não fazerem parte da primeira versão da aplicação, foi necessária a inclusão desta característica de acesso, visto que era pretendido algum tipo de controlo de acesso.

Para este protótipo, o MongoDB foi instalado localmente e não há nenhuma configuração extra para utilização, o que já deixa a aplicação pronta para receber os parâmetros da REST API desenvolvida para o *back-end* para manipulação dos dados determinados em cada modelo.

Coimbra Business School | ISCAC
Quinta Agrícola – Bencanta, 3045-601 Coimbra
Tel +351 239 802 000; E-mail: secretariado@iscac.pt; www.iscac.pt

A REST API utilizada no *back-end* utiliza o *Express.js* como base para garantir a comunicação entre base de dados e o *front-end*. Como modelo estrutural do *back-end*, foi utilizado o projeto em Elnfar (2022) como um referencial, que explica a criação das conexões com base de dados, tratamentos de erros e alertas e utilização criptografia para uma utilização mais segura. Para tratamento das requisições entre o *front-end* e o *Express.js* foi utilizado o Axios. O Axios é uma ferramenta cliente HTTP que faz o tratamento das requisições *POST* e *GET*, trabalhando como intermediário entre o *front-end* e o servidor web de comunicação com a base dados.

A configuração do *Express.js* envolve alguns componentes para compor a REST API seu funcionamento esperado, sendo elas:

- *Express.js* – O servidor web propriamente dito.
- *Monitor* – Utilizado para manter o servidor *Express.js* sempre ativo, mesmo após alterações no código.
- *Mongoose* – Biblioteca responsável pela comunicação entre o servidor e a base de dados, através de comandos internos da mesma.
- *Bcrypt* – Biblioteca que faz a encriptação de dados e faz-se necessária pela utilização, armazenamento e a necessidade de validação de palavra-passe dentro da aplicação.
- *JWT* – Utilizado para geração de um *token* a partir de parâmetros como a dificuldade do mesmo, o tempo de expiração e quais dados farão parte na criação. Esta biblioteca é fundamental para garantir a unicidade do utilizador e garantir que as palavras-passe não serão desencriptadas.
- *dotEnv* – biblioteca utilizada para criação de variáveis que guardam informações sensíveis e que ficam armazenadas em um ficheiro de configuração interno.

Para a instalação de todos os componentes listados acima utilizou-se o seguinte comando:

```
npm install express monitor bcrypt jwt dotenv
```

A configuração da API, de forma geral, passa pelas definições gerais de conexão como porta de acesso, configuração de acesso e comunicação com a base de dados, a definição

Coimbra Business School | ISCAC
Quinta Agrícola – Bencanta, 3045-601 Coimbra
Tel +351 239 802 000; E-mail: secretariado@iscac.pt; www.iscac.pt

dos modelos que serão utilizadas nas *collections*, as rotas de acesso HTTP e o comportamento de cada requisição.

A Figura 3.24 apresenta algumas das configurações gerais utilizadas na REST API, já com algumas das bibliotecas instaladas em funcionamento.

```
36  app.unsubscribe(express.json())
37  app.use(bodyParser.json())
38  app.use(express.json());
39
40  app.get('/', (req, res) => {
41      res.send('Boas vindas')
42  })
43
44  app.use('/api', authRoute)
45
46  const port = process.env.PORT || 5004
47
48
49  const start = async () => {
50      await connectDB(process.env.MONGO_URL)
51
52      app.listen(port, () => {
53          console.log(`a rodar na ${port}`)
54      })
55  }
56
57  start()
```

Figura 3.24 Configuração geral da REST API

Podemos ver na linha 44 (ver Figura 3.24), como a REST API vai se comportar em relação a URL. Basicamente está explícito que o caminho a ser utilizador passará sempre pelo endereço do servidor, seguido com o /api e a rota correspondente de cada método utilizado. Na linha 46 observamos a configuração da porta que o servidor irá funcionar, utilizando as variáveis internas com a biblioteca *dotenv* para ocultar a porta. Já na linha 49, temos a comunicação com a base de dados, onde a variável MOGO_URL representa a *url* de acesso à base de dados.

O ficheiro de definição das rotas contém os métodos utilizados na REST API, tanto as requisições do tipo *GET* quanto *POST*, conforme mostrado abaixo na Figura 3.25.

```
1   import express from "express";
2   import {register, login, getUserInfo, setTransaction, fetchTransaction, fetchDocById} from "../controller/authController.js"
3   const router = express.Router()
4
5   router.route('/register').post(register)
6   router.route('/login').post(login)
7   router.route('/getuserinfo').post(getUserInfo)
8   router.route('/transaction').post(setTransaction)
9   router.route('/getdoc').post(fetchTransaction)
10  router.route('/getdocid').post(fetchDocById)
```

Figura 3.25 Configurações de rotas na REST API

Coimbra Business School | ISCAC
Quinta Agrícola – Bencanta, 3045-601 Coimbra
Tel +351 239 802 000; E-mail: secretariado@iscac.pt; www.iscac.pt

Como forma de organização e centralização das informações, as rotas estão todas especificadas com o caminho de acesso, juntamente com o método utilizado.

Outro ficheiro é utilizado para definir as funções e respetivos parâmetros que farão parte das requisições. A título de exemplo, na Figura 3.26, apresenta-se a função de registo de utilizador. Esta função possui uma entrada, definida pelo *req.body* na linha 6 e segue com a criação do utilizador e respetivo *token*.

```
4   const register = async(req, res) => {
5       try {
6           const {name,idAddress, email,company,userName, contactNumber, password} = req.body
7           const user = await User.create({name,idAddress, email,company, userName, contactNumber, password})
8           const token = user.createJWT()
9
10          res.status(201).json({user:{email:user.email,name:user.name, idAddress:user.idAddress, company:user.company, userName:user.userName, contactNumber:user.contactNumber},token})
11
12      }catch(error) {
13          res.status(500).json({msg: 'Ocorreu um erro'})
14      }
15  }
```

Figura 3.26 Função da REST API de registo de utilizador

Para além da definição das funções que tratam as requisições do *front-end* a *REST API* permite também definir os modelos de dados que serão utilizados na base de dados do *MongoDB* designados de *collections*, sendo que toda a interação com o MongoDB é feita a partir da *REST API*.

O primeiro modelo que se apresenta é o modelo relativo aos utilizadores que foi designado de *Users* e cujos atributos e funções se apresentam nas Figura 3.27 e Figura 3.28.

Coimbra Business School | ISCAC
Quinta Agrícola – Bencanta, 3045-601 Coimbra
Tel +351 239 802 000; E-mail: secretariado@iscac.pt; www.iscac.pt

```
const UserSchema = new mongoose.Schema({
    name:{
        type:String,
        required:[true, 'Insira teu nome'],
        minlength:4,
        maxlength:14

    },
    idAddress:{
        type:String,
        required:[true, 'Insira o ID do utilizador'],
        unique: true,
    },
    email:{
        type:String,
        required:[true, 'Insira teu email'],
        validate:{
            validator:validator.isEmail,
            message:'Por favor informe um email válido'
        },
        unique: true,
    },
    company:{
        type:String,
        required:[true, 'Insira o nome da empresa'],


    },
    userName:{
        type:String,
        required:[true, 'Insira o nome de acesso'],
        unique:true,
    },
    contactNumber:{
        type:String,
        required:[true, 'Insira o numero para contacto'],
    },
    password:{
        type:String,
        required:[true, 'Insira tua palavra passe'],
        minlength:6,
        select:false
    }
})
```

Figura 3.27 Modelo Users

No final do ficheiro de definição do modelo, ainda existe uma série de funções para utilização de criptografia, como podemos ver na Figura 3.28, que complementam as funções de criação de *password* e *token*:

```
UserSchema.pre('save', async function() {
    const salt = await bcryptjs.genSalt(10)
    this.password = await bcryptjs.hash(this.password,salt)
})

UserSchema.methods.createJWT = function() {
    return jwt.sign({userId:this._id},process.env.JWT_SECRET,{expiresIn:process.env.JWT_LIFETIME})
}

UserSchema.methods.comparePassword = async function (candidate) {
    const isMatch = await bcryptjs.compare(candidate, this.password)
    return isMatch
}
```

Figura 3.28 Funções para utilização ode criptografia no modelo Users

É utilizado um parâmetro que, antes que seja efetuada a execução do modelo, é executada uma função para criptografar a *password* antes de ser enviada para base de dados, através da biblioteca *bcrypt*. O mesmo acontece nas funções seguintes, para criação do *token* e validação de passwords para a função de login.

Coimbra Business School | ISCAC
Quinta Agrícola – Bencanta, 3045-601 Coimbra
Tel +351 239 802 000; E-mail: secretariado@iscac.pt; www.iscac.pt

O segundo modelo é responsável por armazenar as informações relativas às transações (orçamentos criados) efetuadas e que servirão para compor parte importante do código para reduzir os acessos à Blockchain. Embora esta tabela seja local, os orçamentos disponibilizados na lista de transações quando carregados para detalhes, trarão os dados do documento diretamente da Blockchain e os dados retornados ao ecrã para visualização. Isto garante que os dados mais sensíveis do documento permaneçam inalterados, pois uma vez armazenados na Blockchain não podem ser eliminados ou alterados, diferentemente dos dados em uma base de dados local. A Figura 3.29 apresenta a definição deste modelo:

```
import mongoose from 'mongoose'
import validator from 'validator'

const TransSchema = new mongoose.Schema({
    docNumber:{
        type: Number,
        required:true,
        unique: true

    },
    transSender:{
        type:String,
        required:[true, 'Insira o ID do utilizador'],

    },
    transReceip:{
        type:String,
        required:[true, 'Insira o ID do destinatário'],

    },
    startDate: {
        type:String,
        required:true
    },
    transItems: {
        type:String,
        required:true
    },
    endDate:{
        type:String,
        required:true
    }

})

export default mongoose.model('Doctran',TransSchema)
```

Figura 3.29 Modelo Transaction

3.2.3.4 Front-end

Em termos aplicacionais, as funcionalidades básicas permaneceram, visto que os utilizadores ainda podem enviar os orçamentos, visualizar seu histórico de transações e consultá-las. Como novas funcionalidades, foram inseridas uma página de *login* e uma

Coimbra Business School | ISCAC
Quinta Agrícola – Bencanta, 3045-601 Coimbra
Tel +351 239 802 000; E-mail: secretariado@iscac.pt; www.iscac.pt

seção de gestão de utilizadores, onde o gestor principal do contrato poderá dar acesso à aplicação.

Para simplificar a estrutura, foi utilizada a biblioteca *Chackra UI*, que é um conjunto de componentes disponíveis no *React*, que já traz *templates* de objetos utilizados no desenvolvimento de aplicações *web*. Apesar de facilitar o desenvolvimento de *layouts* e estruturas da página, ainda assim não é dispensável a necessidade de esforço para adaptação e inserção de código para que as iterações correspondessem ao que foi planeado.

Algumas bibliotecas específicas são fundamentais para o funcionamento da aplicação. Para além de bibliotecas normais, foi necessário a instalação e configuração de alguns parâmetros para que a aplicação funcionasse, de forma atualizada, com o *React* 18 e *Webpack* 5, versões mais recentes. Estas atualizações fazem parte de um dos principais objetivos do projeto, que visa a segurança da aplicação para reduzir brechas de segurança e a utilização de novos componentes utilizados nas versões atuais. Abaixo na Figura 3.30, temos uma visão do ficheiro com a configuração das bibliotecas instaladas na aplicação.

```
6   "@chakra-ui/react": "^2.3.1",
7   "@emotion/react": "^11.10.4",
8   "@emotion/styled": "^11.10.4",
9   "@testing-library/jest-dom": "^5.16.5",
10  "@testing-library/react": "^13.4.0",
11  "@testing-library/user-event": "^13.5.0",
12  "axios": "^0.27.2",
13  "buffer": "^6.0.3",
14  "dotenv": "^16.0.2",
15  "framer-motion": "^7.2.1",
16  "ganache-cli": "^6.12.2",
17  "process": "^0.11.10",
18  "react": "^18.2.0",
19  "react-dom": "^18.2.0",
20  "react-icons": "^4.4.0",
21  "react-router-dom": "^6.3.0",
22  "react-scripts": "^5.0.1",
23  "truffle": "^5.5.30",
24  "web-vitals": "^2.1.4",
25  "web3": "^1.7.5",
26  "webpack": "^5.74.0"
```

Figura 3.30 Ficheiro package.json de configuração das bibliotecas do front-end

Dentre os componentes acima, podemos destacar por exemplo o *Web3.js* (*Ethereum* Revision, 2022b), que é responsável pela interação entre o *front-end* e o *smart contract*. Outra biblioteca é o *Truffle*, que nesta fase de testes traz bibliotecas que possibilitam a utilização de uma Blockchain teste juntamente com o ganache, criação de estruturas para

Coimbra Business School | ISCAC
Quinta Agrícola – Bencanta, 3045-601 Coimbra
Tel +351 239 802 000; E-mail: secretariado@iscac.pt; www.iscac.pt

suportar os contratos entre outras. A biblioteca do Axios, que conforme mencionado anteriormente, faz o tratamento das requisições HTTP de forma simples entre *front-end* e API.

A criação de uma aplicação no *React* é bem rápida para começar. É possível iniciar projetos do zero, configurando manualmente os ficheiros necessários ou utilizando um *template* básico de uma aplicação funcional. Para este projeto, inicialmente foi utilizado o *template* disponível do *React*, com o comando abaixo dentro do diretório do projeto desejado:

```
npm react-create-app
```

Em seguida, a instalação do Chakra UI:

```
npm install chakra-ui
```

Após a instalação do *Chakra UI* utilizam-se os *templates* disponibilizados na biblioteca para implementar os ecrãs da BudgetDApp que correspondem às funcionalidades anteriormente definidas no novo diagrama de caso de uso (ver secção 3.2.1.1).

O primeiro ecrã da aplicação, após a autenticação na mesma, é página de entrada de todos os utilizadores, a *homepage*, em que os utilizadores podem visualizar o seu ID da conta e respetivo saldo. Isto comprova que o utilizador está conectado em uma carteira digital, conforme Figura 3.31.

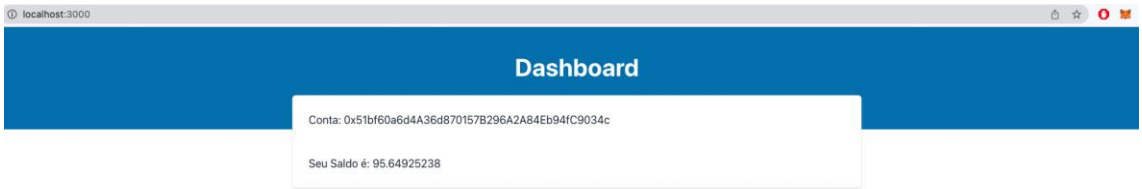


Figura 3.31 Página Home relativa ao UC 9 Visualizar saldo

A Figura 3.32 refere-se a página que antecede todas as outras, onde os utilizadores deverão se autenticar para acesso à aplicação.

Coimbra Business School | ISCAC
Quinta Agrícola – Bencanta, 3045-601 Coimbra
Tel +351 239 802 000; E-mail: secretariado@iscac.pt; www.iscac.pt

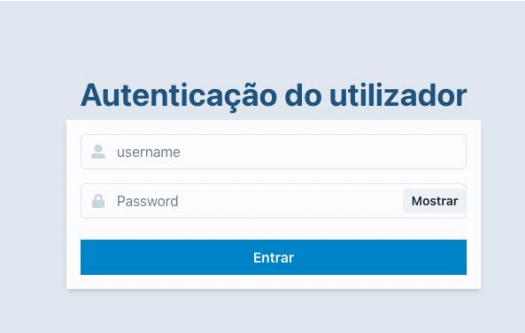


Figura 3.32 Formulário de login referente ao UC 1 Autenticar na DApp

Página de gestão para registo dos utilizadores na aplicação, conforme Figura 3.33.

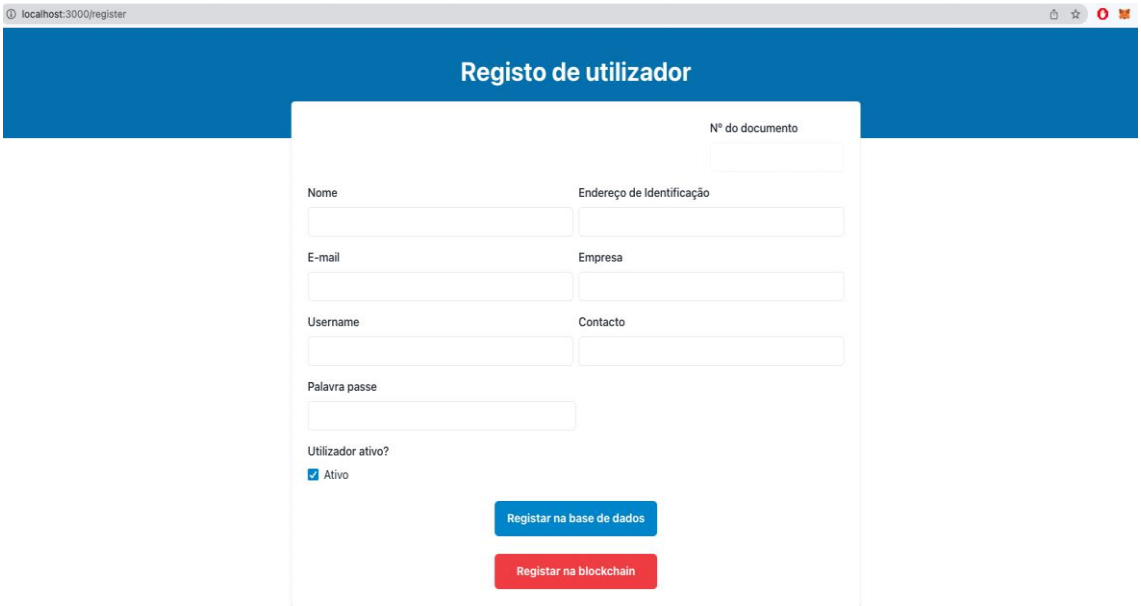


Figura 3.33 Formulário de registo e utilizador referente ao UC 10 Registar utilizador

Principal componente da aplicação, onde os orçamentos são preenchidos e enviados, visto em Figura 3.34.

Coimbra Business School | ISCAC
Quinta Agrícola – Bencanta, 3045-601 Coimbra
Tel +351 239 802 000; E-mail: secretariado@iscac.pt; www.iscac.pt

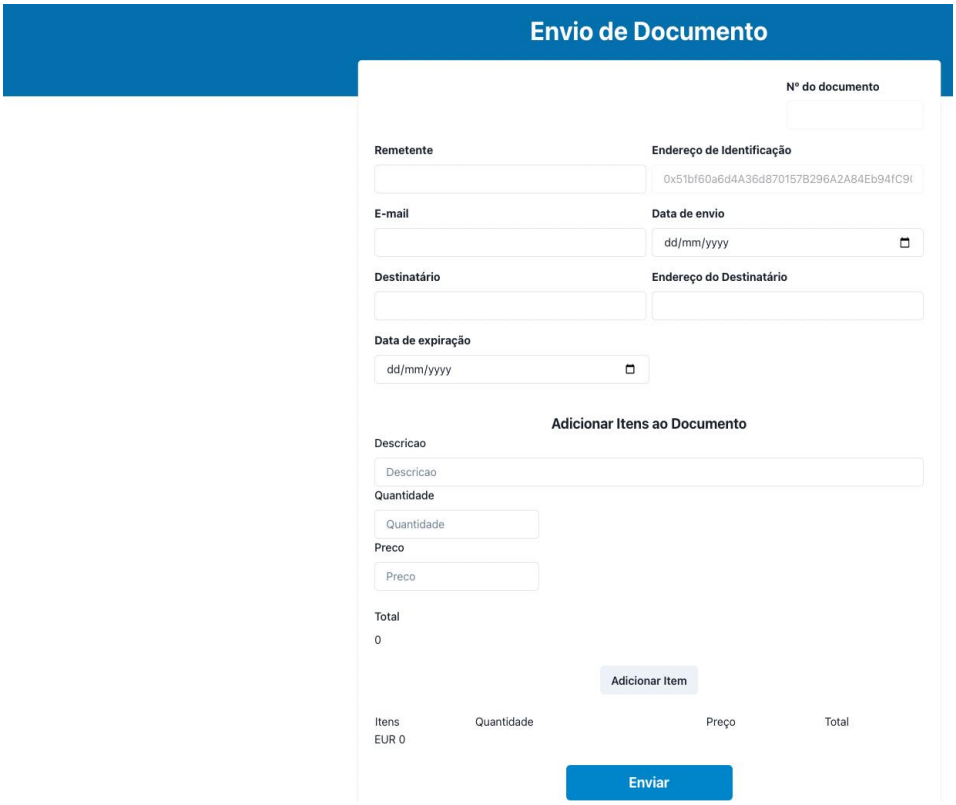


Figura 3.34 Formulário SendDocs referente ao UC 7 Enviar orçamento para Blockchain

A Figura 3.35representa a página com a listagem das transações efetuadas pelo utilizador corrente.

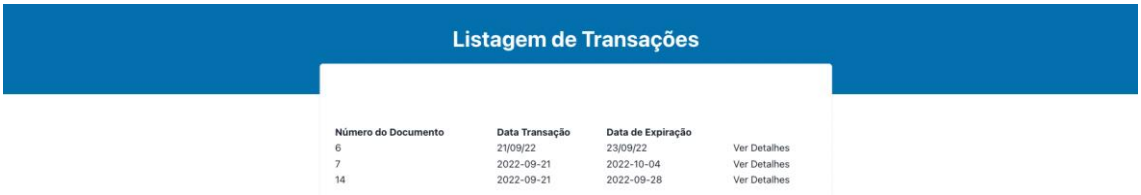


Figura 3.35 Página referente ao UC 8 Consultar orçamentos

Conforme mencionado anteriormente, algumas bibliotecas estão a ser utilizadas no *front-end* para comunicação com outros componentes da DApp nomeadamente a comunicação com a *REST API* e a carteira digital. Como uma forma de exemplificar esta integração, serão listadas abaixo algumas dessas funções utilizadas na BudgetDApp.

A primeira função que se apresenta é a função *load* que valida se o utilizador está conectado em uma carteira digital, conforme código abaixo na Figura 4.37:

Coimbra Business School | ISCAC
Quinta Agrícola – Bencanta, 3045-601 Coimbra
Tel +351 239 802 000; E-mail: secretariado@iscac.pt; www.iscac.pt

```
38  useEffect(() => {
39    async function load() {
40      const web3 = new Web3(Web3.givenProvider || 'http://192.168.1.118:7545');
41      const accounts = await web3.eth.requestAccounts();
42
43      setAccount(accounts[0]);
44      console.log(accounts);
45      account2 = (accounts[0]);
46      const balance = await web3.eth.getBalance(account2);
47      setDefineBalance(balance / 1000000000000000000);
48
49    }
50
51    load();
52  }, []);
```

Figura 3.36 Função load

Pode-se ver na linha 40 (Figura 3.36), que através da biblioteca *Web3.js* (*Ethereum* Revision, 2022b), é feita a conexão com a Blockchain pré-estabelecida, seguindo com parâmetros para verificar qual conta está a ser acedida no momento e atribui o valor à variável *account*. Em seguida é feita a consulta do saldo do utilizador que é retornado na variável account2.

No código abaixo disposto na Figura 3.37, referente à função que devolve os detalhes do orçamento disponíveis na Blockchain, é possível ver que é feita uma consulta à Blockchain para retornar as informações referentes ao destinatário e a parte dos itens da tabela. Em seguida é feita uma requisição com *Axios* para retornar as informações do remetente, compondo o conjunto necessário para retornar o documento.

Coimbra Business School | ISCAC
Quinta Agrícola – Bencanta, 3045-601 Coimbra
Tel +351 239 802 000; E-mail: secretariado@iscac.pt; www.iscac.pt

```
async function SetGetMsg() {
  const web3 = new Web3(Web3.givenProvider || 'http://192.168.1.118:7545');
  const contract = new web3.eth.Contract(Contract_ABI, '0x8FCC014fB765510788901bE99b1E51654Beea83C')
  const Message = 7;//await SetfGetDocId();

  const result = await contract.methods.getMsgDetail(Message).call()
    .then((readmsg) => {
      var lines = readmsg.split("-");
      for (let index = 0; index < lines.length; index++) {
        var line = lines[index].split(";");
        var novoItem = {
          id: index,
          descricao: line[0],
          quantidade: line[1],
          preco: line[2],
          total: line[3]
        }

        if(novoItem.id === 0 && lista.length === 0)
        {
          setLista(lista => [...lista, novoItem]);
        }
          for (let ind = 0; ind < lista.length; ind++) {
            if(lista[ind].id != novoItem.id ){
              console.log("Id Antigo: ", lista[ind].id  );
              console.log("Id Novo: ", novoItem.id  );
              setLista(lista => [...lista, novoItem]);
            }
          }
      }

    contract.methods.getDocDetail(Message).call()
    .then((readDoc) => {
      setDataExpiracao(readDoc[4]);
      setDestinatario(readDoc[2]);
      setIdDestinatario(readDoc[1]);
      setNumeroDocumento(readDoc[0]);
      setDestinatario(readDoc[2]);
    })
    .catch((error) => { console.log(error) });

    const useInfo = axios.post('http://localhost:5003/api/getuserinfo', { 'idAddress': myaccount })
    .then((result) =>

      setUseData(result.data)
    )

    })
    .catch((error) => { console.log(error) });
}
```

Figura 3.37 Função do front-end de comunicação com a REST-API para retorno de conteúdo de orçamentos

Na Figura 3.38 apresenta-se a função do *front-end* que invoca a função da REST API para registo de utilizador. Pode-se verificar que a variável *currentUser* recebe os registos referentes ao *req.body* da função de registo de utilizadores e passa os valores das variáveis que recebem informações da página, seguindo então com a passagem desses valores para execução da função na *REST API*, que fica da forma vista em Figura 3.39.

Coimbra Business School | ISCAC
Quinta Agrícola – Bencanta, 3045-601 Coimbra
Tel +351 239 802 000; E-mail: secretariado@iscac.pt; www.iscac.pt

```
74    const handleSubmit = e => {
75      e.preventDefault()
76
77      if(!userName ||!idAddress || !email || !password || (!isMember && !name)){
78        displayAlert()
79        return
80      }
81      const currentUser = {
82        "name":name,
83        "idAddress":idAddress,
84        "email": email,
85        "company":company,
86        "userName": userName,
87        "contactNumber":company,
88        "password":password
89      }
90      console.log(currentUser)
91      registerUser(currentUser)
92
93      if(isMember) {
94
95        console.log("Utilizador já está registado")
96
97      }else {
98
99      }
100
101   }
```

Figura 3.38 Função do front-end de registo de utilizador

```
54    //Função para registo de um novo utilizador
55    const registerUser = async (currentUser) => {
56        dispatch({
57            type:REGISTER_USER_BEGIN
58        })
59
60        try {
61
62            const response = await axios.post('http://localhost:5003/api/register',currentUser)
63            console.log(response)
64
65            const {user,token} = response.data
66            dispatch({ type:REGISTER_USER_SUCCESS,payload:{user,token}})
67
68            addUserToLocalSt({user,token})
69
70        }catch(error) {
71
72            console.log(error.response);
73
74            dispatch({ type:REGISTER_USER_ERROR,payload:{msg:error.response.data.msg} })
75
76        }
77
78        clearAlert()
79
80    }
```

Figura 3.39 Função do front-end de comunicação com a REST API para registo de utilizador

Esta segunda iteração da BudgetDApp permite controlar o acesso dos utilizadores à mesma e guardar os dados relativos aos utilizadores e orçamentos numa base de dados local baixando desta forma os custos de utilização da BudgetDApp.

Coimbra Business School | ISCAC
Quinta Agrícola – Bencanta, 3045-601 Coimbra
Tel +351 239 802 000; E-mail: secretariado@iscac.pt; www.iscac.pt

4 ANÁLISE DE VULNERABILIDADES

Análise das possíveis vulnerabilidades da BudgetDApp é a próxima etapa da metodologia deste projeto, "Segurança de dados em aplicações baseadas em Blockchain". Esta etapa exige um esforço em conjunto de diferentes equipas e ferramentas. A revisão manual feita por desenvolvedores é tão importante quanto a utilização de ferramentas automatizadas para deteção de brechas de segurança. Neste capítulo, a análise se dividirá com a identificação das vulnerabilidades no *smart contract* e na análise da aplicação. Em cenários reais de aplicações é normal a utilização de diferentes tipos de ferramentas com a utilização de métodos diferentes para o mesmo objetivo, entretanto para efeitos de demonstração de resultados aplicado do projeto, serão utilizadas apenas uma ferramenta para analisar cada um dos principais componentes da aplicação descentralizada.

4.1 Análise do *smart contract*

A utilização de ferramentas para deteção de vulnerabilidades é um processo que, juntamente com a utilização das boas práticas, desempenha um papel essencial no desenvolvimento de uma aplicação. Conforme visto no Capítulo 2, existem diversos tipos de ferramentas que utilizam métodos de análise diferente e em um processo de maior dimensão, por vezes a utilização de mais de uma é importante para encontrar e corrigir falhas no código que não foram detetadas na revisão. Como uma forma ilustrativa, no presente projeto o *smart contract* desenvolvido na segunda iteração será avaliado como forma do processo. Para tal, foi utilizada a ferramenta Mythx (ConsenSys Software Inc™, 2022), que é uma ferramenta automatizada que utiliza uma série técnicas de análise com objetivo de identificar as vulnerabilidades de um contrato.

4.1.1 Ferramenta de análise Mythx

O Mythx é uma ferramenta proprietária que utiliza técnicas de análise dinâmica e estática, utilizando a lista vulnerabilidades disponibilizada por SmartContractSecurity (2022), designada de *Smart contract Weakness Classification* (SWC), que é um referencial para desenvolvedores de contratos inteligentes. Além de classificar as vulnerabilidades, a SWC descreve o problema, indica a solução e possui exemplos do que está sendo tratado.

Coimbra Business School | ISCAC
Quinta Agrícola – Bencanta, 3045-601 Coimbra
Tel +351 239 802 000; E-mail: secretariado@iscac.pt; www.iscac.pt

Para utilização da ferramenta Mythx foi necessária a integração da mesma com o editor de código utilizado, neste caso o Remix IDE disponível em (*REMIX*, 2022). Esta integração foi feita através da instalação de um plugin que faz a conexão entre o editor e a ferramenta de análise, conforme se pode visualizar na Figura 4.1.

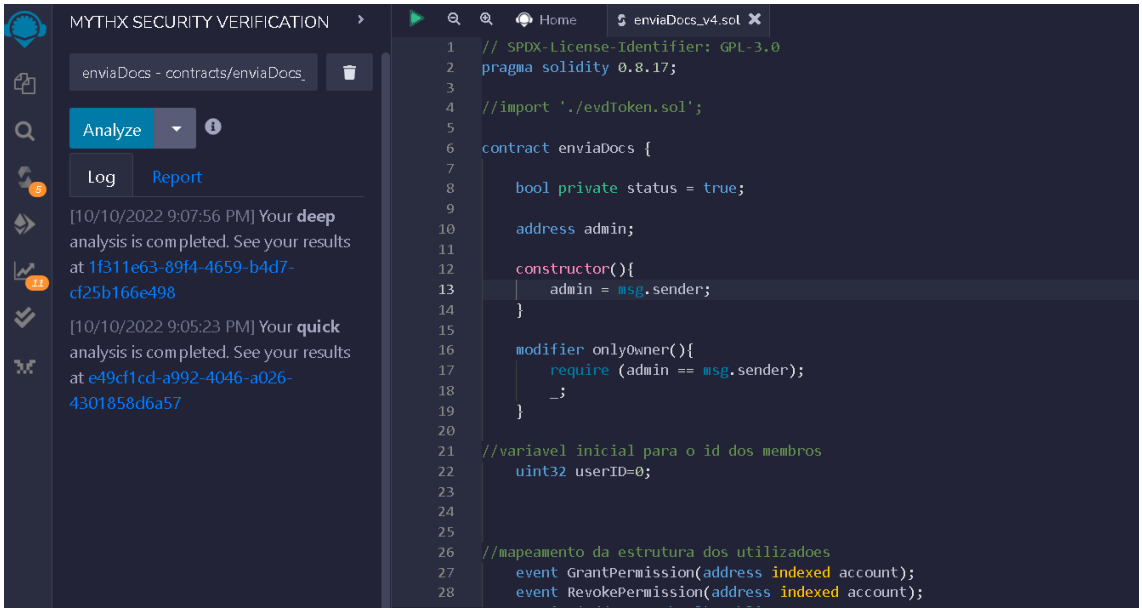


Figura 4.1 Integração do Mythx com REMIX IDE

A Figura 4.1 mostra a integração efetuada no Mythx, a opção de efetuar a análise e um resumo no *Log* sobre o que foi efetuado.

Para executar a análise é preciso compilar previamente o código do contrato no REMIX e então escolher o tipo da análise que será efetuada e o resultado estará disponível na conta do Mythx, juntamente com as informações e respetivos relatórios.

Através da ferramenta é possível fazer a gestão de todos os contratos analisados separando-os por projetos, ter acesso a todos os relatórios efetuados e exportar os resultados, caso assim seja pretendido. A Figura 4.2 ilustra o ambiente de análise da aplicação.

Coimbra Business School | ISCAC
Quinta Agrícola – Bencanta, 3045-601 Coimbra
Tel +351 239 802 000; E-mail: secretariado@iscac.pt; www.iscac.pt

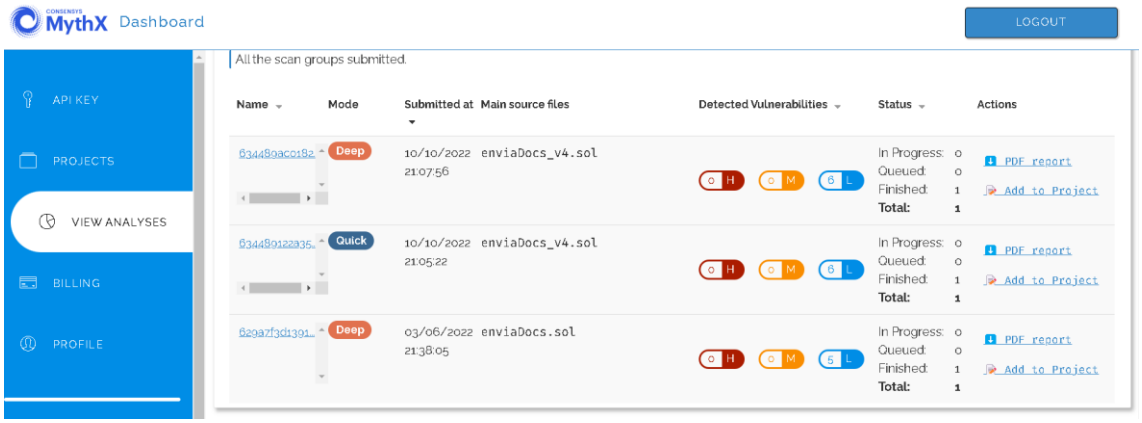


Figura 4.2 Dashboard de análise Mythx

Como é visível na Figura 4.2, o painel ao lado esquerdo mostra um resumo de todos os contratos que foram analisados, qual o estado de execução de cada e o total de vulnerabilidades encontradas, além de indicar qual tipo de análise foi utilizada, se uma análise rápida ou profunda. Na análise rápida é possível detetar erros e vulnerabilidades de baixo impacto, enquanto a análise profunda percorre o código de forma mais complexa e completa a fim de encontrar todas vulnerabilidades nos mais diversos níveis.

Para classificar a severidade das vulnerabilidades a ferramenta Mythx utiliza três níveis de severidade, conforme a Figura 4.3.

Figura 4.3 Níveis de severidade Mythx

As vulnerabilidades de baixa severidade e representadas em amarelo indicam que foram encontradas apenas não conformidades que dizem respeito às boas práticas, mas que de toda forma devem ser analisadas para manter o código dentro de padrões. As indicações nesta categoria por vezes indicar alguma violação que não parece importante, mas faz parte das boas práticas manter um código limpo, simples e de fácil identificação. As vulnerabilidades na cor laranja indicam possíveis brechas de segurança e necessitam uma atenção e ação de correção por parte do desenvolvedor. Enquanto o vermelho representa problemas críticos que deixam iminente o código a ataques através de brechas de segurança catalogadas.

Coimbra Business School | ISCAC
Quinta Agrícola – Bencanta, 3045-601 Coimbra
Tel +351 239 802 000; E-mail: secretariado@iscac.pt; www.iscac.pt

4.1.2 Resultado da análise

Como resultado de uma análise profunda que utiliza técnicas de análise estáticas, dinâmicas e simbólicas, foram encontradas apenas vulnerabilidades de baixo impacto, conforme Figura 4.4.

Severity Low (6) Medium (0) High (0)

Ignored Issues Hidden (0)

ID	Severity	Name	File	Location
SWC-108	Low	State variable visibility is not set.	enviaDocs_v4.sol	L: 10 C: 12
SWC-108	Low	State variable visibility is not set.	enviaDocs_v4.sol	L: 22 C: 11
SWC-108	Low	State variable visibility is not set.	enviaDocs_v4.sol	L: 64 C: 17
SWC-108	Low	State variable visibility is not set.	enviaDocs_v4.sol	L: 103 C: 14
SWC-108	Low	State variable visibility is not set.	enviaDocs_v4.sol	L: 119 C: 7
SWC-108	Low	State variable visibility is not set.	enviaDocs_v4.sol	L: 121 C: 7

Figura 4.4 Resultado de análise no smart contract com o Mythx

O resumo acima representa as vulnerabilidades encontradas juntamente com as referências e classificações respetivas. Logo abaixo encontra-se a definição e descrição da única vulnerabilidade encontrada no *smart contract*:

***SWC-108**, visibilidade de variáveis globais*

Para se ter uma noção mais ampla sobre a importância da visibilidade das variáveis, é necessário perceber um tanto melhor a definição de cada uma delas, como é visto em (Consensys, 2022). Enquanto as variáveis privadas podem ser utilizadas e chamadas dentro do contrato que foi definida, não podendo ser utilizada em outros contratos, a visibilidade pública permite que a variável seja chamada tanto internamente quanto externamente. Já as variáveis e funções externas podem ser chamadas apenas de fora do contrato, enquanto as internas podem ser utilizadas apenas internamente. Desta forma, segundo a SWC, a definição da visibilidade das variáveis é recomendada porque explicita quem deve ter acesso às variáveis, portanto as mesmas devem ser definidas entre públicas, privadas ou internas.

Coimbra Business School | ISCAC
Quinta Agrícola – Bencanta, 3045-601 Coimbra
Tel +351 239 802 000; E-mail: secretariado@iscac.pt; www.iscac.pt

O principal objetivo destas recomendações de segurança é evitar que funções e variáveis sejam utilizadas de forma inadequada, cabendo ao desenvolvedor analisar o risco e objetivo de deixá-las sem tal definição. As variáveis de estado têm por defeito a visibilidade interna e certamente por isso não são consideradas vulnerabilidade de baixo impacto, entretanto é recomendado que esta visibilidade esteja claramente apresentada no código, como é visto em (*Ethereum* Revision, 2022a).

Na Figura 4.5 pode-se perceber a importância desta definição para que não haja dúvidas sobre a clareza desta recomendação:

Figura 4.5 Declaração de variáveis smart contract

A declaração da variável na linha 10 refere-se a uma variável interna, que deveria possuir esta visibilidade declarada conforme recomendação, e que será utilizada mais abaixo sendo chamadas para partes fundamentais do código. Esta parte fundamental determina quem será a pessoa que poderá fazer alterações em várias funções dentro do contrato, portanto é importante que não seja acedida por outro contrato, chamadas de funções ou ação de algum agente malicioso.

4.2 Análise da aplicação

Para analisar inicialmente o *front-end* da aplicação foi utilizada a ferramenta de análise de aplicações da OWASP, chamada Zed Attack Proxy ou simplesmente ZAP (OWASP, 2022a). Esta ferramenta é uma aplicação baseada em código aberto que tem por objetivo analisar a segurança em aplicações web em geral, explorando, simulando ataques e apresentando relatórios sobre as vulnerabilidades encontradas conforme visto no Capítulo 2.

Coimbra Business School | ISCAC
Quinta Agrícola – Bencanta, 3045-601 Coimbra
Tel +351 239 802 000; E-mail: secretariado@iscac.pt; www.iscac.pt

4.2.1 Ferramenta Zed Attack Proxy

Para testar o *front-end* da BudgetDApp, foi utilizada a configuração de análise através do endereço da aplicação utilizado no ambiente de desenvolvimento, conforme ilustrado abaixo na Figura 4.6.

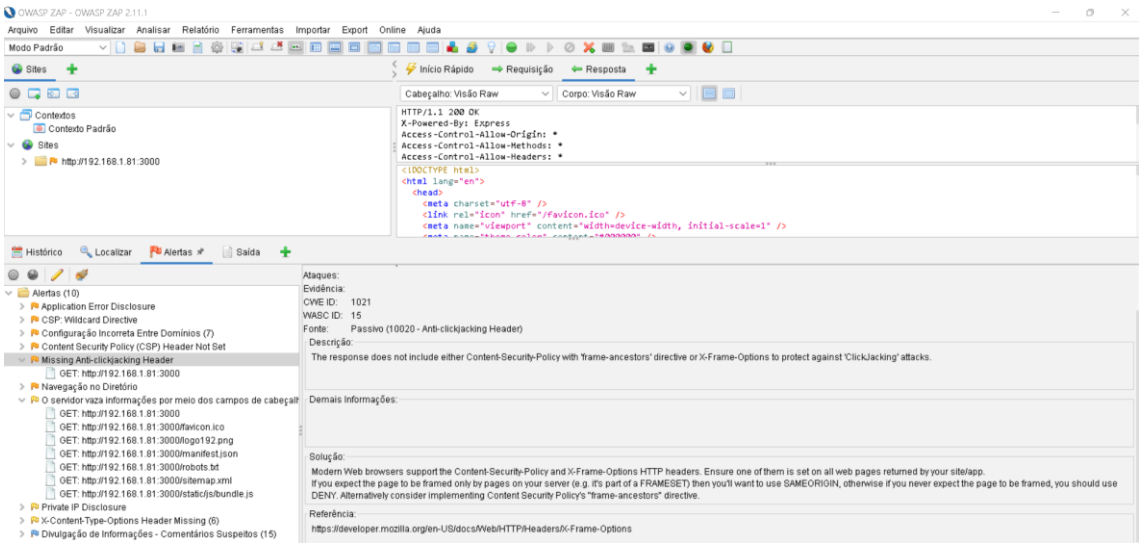


Figura 4.6 Análise de vulnerabilidade ZAP

A Figura 4.6 ilustra a aplicação após realizar a análise em um *website*. No painel inferior esquerdo é possível visualizar entre as opções, os alertas com todas as vulnerabilidades encontradas. No painel à direita, após carregar em uma vulnerabilidade é possível visualizar um resumo sobre o que foi encontrado, com a descrição do problema e soluções de acordo com o referencial interno.

A aplicação ZAP fornece ferramentas internas que executam algumas técnicas durante a análise. Uma das formas para identificação das possíveis falhas é uma avaliação de segurança, onde o sistema analisa questões de segurança expostas na aplicação. Uma outra técnica é a utilização de um teste de penetração, onde a ZAP simula ataques à aplicação e analisa as possíveis vulnerabilidades.

A classificação das vulnerabilidades é mostrada em cada uma das possíveis falhas encontradas. Essa classificação de severidade vai de um nível baixo para alto, levando em consideração a relevância da mesma apresentados inicialmente em uma matriz, onde pode ser visto o risco e relevância e probabilidade das vulnerabilidades. O resultado do

Coimbra Business School | ISCAC
Quinta Agrícola – Bencanta, 3045-601 Coimbra
Tel +351 239 802 000; E-mail: secretariado@iscac.pt; www.iscac.pt

conjunto de técnicas associado com a classificação dos níveis de risco, representam o quanto uma aplicação pode estar vulnerável em termos de segurança.

4.2.2 Resultado da análise

Como resultado, a ferramenta disponibiliza um relatório com todas as vulnerabilidades encontradas, organizados por sessões de níveis de risco. Para além disto, o relatório tem a descrição de cada vulnerabilidade, o que foi feito para obter o resultado, a referência sobre o ocorrido e a solução para contornar a situação.

		Confidence				
		User Confirmed	Alto	MÃ©dio	Baixo	Total
Risk	Alto	0 (0,0%)	0 (0,0%)	0 (0,0%)	0 (0,0%)	0 (0,0%)
	MÃ©dio	0 (0,0%)	2 (20,0%)	3 (30,0%)	1 (10,0%)	6 (60,0%)
	Baixo	0 (0,0%)	0 (0,0%)	3 (30,0%)	0 (0,0%)	3 (30,0%)
	Informativo	0 (0,0%)	0 (0,0%)	1 (10,0%)	0 (0,0%)	1 (10,0%)
	Total	0 (0,0%)	2 (20,0%)	7 (70,0%)	1 (10,0%)	10 (100%)

Figura 4.7 Tabela de risco e confiança

A Figura 4.7 retorna o percentual de alertas encontrados e a classificação de cada de acordo com a criticidade. Através de uma matriz que considera o risco e a confiança, observa-se que as vulnerabilidades encontradas se concentram sem sua maioria entre médio e baixo risco, possuindo apenas um alerta de alto risco que está relacionado com a falta de confiança.

O conceito de confiança neste caso está relacionado com o nível de classificação da aplicação OWASP ZAP de medir o quanto uma possível falha representa a potencialidade de vazar algum tipo de informação que não deveria.

Coimbra Business School | ISCAC
Quinta Agrícola – Bencanta, 3045-601 Coimbra
Tel +351 239 802 000; E-mail: secretariado@iscac.pt; www.iscac.pt

Figura 4.8 Alertas e referências

A Figura 4.8 acima apresenta os alertas encontrados, onde através dos *links* de cada um, encaminha-se para uma descrição completa da vulnerabilidade e a solução para cada problema, como pode ser vista na Figura 4.9 logo abaixo.

Coimbra Business School | ISCAC
Quinta Agrícola – Bencanta, 3045-601 Coimbra
Tel +351 239 802 000; E-mail: secretariado@iscac.pt; www.iscac.pt

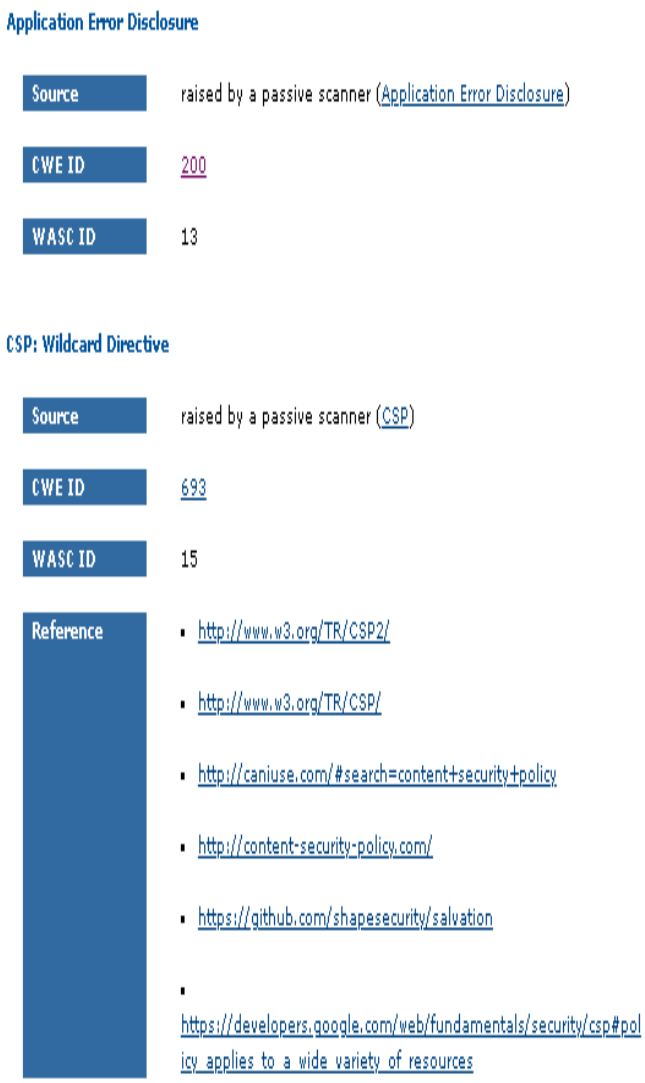


Figura 4.9 Descrição da vulnerabilidade

Coimbra Business School | ISCAC
Quinta Agrícola – Bencanta, 3045-601 Coimbra
Tel +351 239 802 000; E-mail: secretariado@iscac.pt; www.iscac.pt

Alert type	Risk	Count
Application Error Disclosure	MÃ©dio	1 (10,0%)
CSP: Wildcard Directive	MÃ©dio	1 (10,0%)
Configuração Incorreta Entre Domínios	MÃ©dio	7 (70,0%)
Content Security Policy (CSP) Header Not Set	MÃ©dio	1 (10,0%)
Missing Anti-clickjacking Header	MÃ©dio	1 (10,0%)
Navegação no Diretório	MÃ©dio	1 (10,0%)
O servidor vaza informações por meio dos campos de cabeçalho de resposta HTTP "X-Powered-By"	Baixo	7 (70,0%)
Private IP Disclosure	Baixo	1 (10,0%)
X-Content-Type-Options Header Missing	Baixo	6 (60,0%)
Divulgação de Informações - Comentários Suspeitos	Informativo	15 (150,0%)
Total		10

Figura 4.10 Tipo de alerta

A Figura 4.10 representa os alertas encontrados e classificados de acordo com percentual de risco e a identificação de cada vulnerabilidade. Abaixo, encontram-se a descrição de algumas das vulnerabilidades listadas:

- *Missing Anti-clickjacking Header*

  Segundo o manual para o desenvolvimento de aplicações da Mozzila (Mozilla.org, 2022) o cabeçalho X-Frame-Options é utilizado para evitar ataques de *click-hijacking*, que se baseiam em ataques que podem obter informações do utilizador através de botões, links e etc.

- *Content Security Policy (CSP) Header Not Set*

  De acordo com o W3C (2022) esta é uma ferramenta que pode reduzir os possíveis danos causados por ataques do tipo *injection*, Segundo os autores, esta ferramenta deve ser utilizada como um complemento, mitigando os riscos em ataques por *scripting*, por exemplo, através da configuração desta política.

Coimbra Business School | ISCAC
Quinta Agrícola – Bencanta, 3045-601 Coimbra
Tel +351 239 802 000; E-mail: secretariado@iscac.pt; www.iscac.pt

- O servidor vaza informações por meio dos campos de cabeçalho de resposta HTTP "X-Powered-By"

  Conforme visto em (OWASP, 2022a), este é o vazamento de informações através de um cabeçalho "X-Powered-By" que pode facilitar a identificação de componentes de uma aplicação web por hackers, possibilitando a exploração das vulnerabilidades de cada.

- *Private IP Disclosure*

  De acordo com os referenciais exibidos no relatório e descritos em (OWASP, 2022a), esta é uma falha que expõe o endereço IP interno de possíveis servidores e podem indicar futuros alvos de ataques. Os endereçamentos IPs internos são utilizados para determinar uma ligação com algum servidor ou serviço interno e muitas vezes durante o desenvolvimento, podem ser utilizados de forma imprudente.

4.3 Melhorias e correções

Utilizando a abordagem descrita pela OWASP (2022) as práticas de segurança devem seguir algumas etapas que podem envolver equipas e personagens diferentes dentro de uma organização. A execução do *framework* descrito no OWASP é um conjunto complexo e extenso de testes que tem por objetivo permear todos os componentes da aplicação em desenvolvimento e reduzir os riscos de segurança da mesma. As áreas de testes concentram atenção na inspeção manual de processos, pessoas e arquitetura do sistema. Segundo há a modelagem das ameaças, com o objetivo de identificar as ameaças e repensar as questões de segurança associadas. A revisão do código, que se trata da identificação manual feita por profissionais que procuram por potenciais falhas que não são mapeadas em ferramentas automatizadas. E, por fim, os testes de penetração, que visam testar na prática as potenciais vulnerabilidades da aplicação.

Partindo do pressuposto que foi feito inicialmente uma análise de vulnerabilidade tanto no *smart contract* quanto na aplicação utilizando duas ferramentas distintas, sendo que uma delas uma ferramenta de teste de penetração e outra específica de vulnerabilidade de contrato, alguns passos serão refeitos como forma de demonstrar como deveriam ser feito desde o início do desenvolvimento da aplicação.

Coimbra Business School | ISCAC
Quinta Agrícola – Bencanta, 3045-601 Coimbra
Tel +351 239 802 000; E-mail: secretariado@iscac.pt; www.iscac.pt

A primeira etapa para análise para iniciar uma visão analítica sobre segurança será o desenho da arquitetura da aplicação para uma visão.

4.3.1 Correções no *smart contract*

Conforme reportado na análise feita no código parcial do contrato apresentada na Secção 4.1.1 foi identificada a seguinte vulnerabilidade:

- Visibilidade de variáveis

De acordo com a recomendação feita em Consensys (2022) a visibilidade das variáveis de estado deve ser explicitamente declarada como *internal*, o que foi feito, como se pode ver Figura 4.11.

```
formHeader[] internal FormHeader;
address internal admin;
```

Figura 4.11 Visibilidade das variáveis

4.3.2 Correções em vulnerabilidades do *front-end*

As correções em falhas encontradas no *front-end* são um tanto mais complexas, pois podem envolver diversos componentes testados durante a análise de vulnerabilidade. A execução de mais testes envolvendo outras camadas que fazem parte da arquitetura geral da aplicação são fundamentais para eliminar ou reduzir falhas em camadas como a de rede. Para efeito de ilustração, nesta etapa serão levadas em consideração alguns exemplos mais simples e que estão focados na camada de aplicação.

- O servidor vaza informações por meio dos campos de cabeçalho de resposta HTTP "X-Powered-By" (1)

Conforme informado no relatório de análise da aplicação, esta falha de segurança pode representar o vazamento de informações por meio de cabeçalhos que podem ser um facilitador para atacantes identificarem componentes e estruturas. Para correção desta vulnerabilidade, pode ser feita a configuração no código da aplicação para que as informações de cabeçalho sejam desativadas, como apresentado na Figura 4.12.

Coimbra Business School | ISCAC
Quinta Agrícola – Bencanta, 3045-601 Coimbra
Tel +351 239 802 000; E-mail: secretariado@iscac.pt; www.iscac.pt

```
const app = express();
app.disable("x-powered-by");
```

Figura 4.12 Proposta de correção da vulnerabilidade HTTP "X-Powered-By"

- *Private IP Disclosure*

Para contornar o problema das ligações entre componentes da aplicação e servidores internos (requisições HTTP) é necessário utilizar uma forma de manter esses endereços acedidos internamente pelo código. O trecho de código apresentado na Figura 4.13 apresenta a forma incorreta de fazer as requisições HTTP.

```
 try {

            const response = await
axios.post('http://localhost:5003/api/register',currentUser)
            console.log(response)

            const {user,token} = response.data
            dispatch({ type:REGISTER_USER_SUCCESS,payload:{user,token}})

            addUserToLocalSt({user,token})
```

Figura 4.13 Trecho de código incorreto na requisição HTTP

Como podemos ver o endereço *localhost* que também é conhecido como endereço de *loopback* e indica que o endereço IP do Axios é o mesmo endereço que a aplicação está a funcionar, além de exibir a porta de acesso da API.

Existem várias formas de fazer o esse mascaramento dos endereços IPs, mas a utilizada neste projeto foi a utilização da biblioteca já instalada dotenv. Conforme informado anteriormente, esta biblioteca cria variáveis de ambiente, que podem receber um nome de identificação qualquer para ser utilizado e exposto no código, enquanto o endereço real é armazenado em um ficheiro de configuração interno, deixando assim os endereços fora da exposição.

Para armazenar as os valores das variáveis, é necessário instalar a biblioteca dontenv no componente do *front-end*, caso não esteja, e configurar um novo ficheiro .env com os nomes das novas variáveis e os respetivos endereços. Na Figura 4.14 apresenta-se a forma correta de fazer as requisições HTTP.

Coimbra Business School | ISCAC
Quinta Agrícola – Bencanta, 3045-601 Coimbra
Tel +351 239 802 000; E-mail: secretariado@iscac.pt; www.iscac.pt

```
MTD_REG_URL=http://localhost:5003/api/register
try {
const response = await axios.post(process.env.MTD_REG_URL,currentUser)
          console.log(response)
```

Figura 4.14 Trecho de código correto na requisição HTTP

4.4 Discussão

A proposta desta experiência tem como objetivos principais duas situações distintas, mas que devem estar associadas para que uma não traga problemas para outra. A primeira proposta foi de trabalhar para o desenvolvimento de uma aplicação para uma tarefa rotineira de várias empresas e que, atualmente passam por terem opções nada claras, com custos elevados e com gestão de terceiros. A aplicação BudgetDApp foi concebida com intuito de facilitar a tarefa rotineira do envio de orçamentos entre duas partes utilizando os benefícios da Blockchain. Em contrapartida, outra preocupação com o desenvolvimento da aplicação é voltada para reduzir os riscos de segurança que envolvem o desenvolvimento de uma aplicação descentralizada.

As potencialidades proporcionadas pela Blockchain trazem diversos benefícios e facilidades para desenvolvedores em geral. Com o surgimento e popularização dos *smart contracts*, hoje se torna muito fácil e ágil a criação de aplicações que utilizam desses contratos para estabelecerem as regras e garantirem a transparência do que foi estabelecido. No entanto, da mesma forma que a agilidade torna o surgimento de aplicações, a falta de critérios e métodos na criação de aplicações descentralizadas representam riscos para que falhas de segurança, que muitas vezes são simples, causem prejuízos consideráveis.

Coimbra Business School | ISCAC
Quinta Agrícola – Bencanta, 3045-601 Coimbra
Tel +351 239 802 000; E-mail: secretariado@iscac.pt; www.iscac.pt

CONCLUSÃO

As tecnologias associadas à Blockchain vêm trazendo diversas possibilidades de garantias de privacidade e segurança para utilização em diversas áreas de atuação como logística, identidade digital e internet das coisas. O potencial da Blockchain é indiscutível, entretanto por ainda se tratar de uma tecnologia relativamente recente e a popularização através dos *smart contract*s vêm causando constrangimentos, principalmente na parte financeira.

A falta de governança e padronização com utilização de ferramentas e técnicas para acompanhamento das etapas de conceção de um projeto que envolve Blockchain é um tema sensível que pode comprometer de diversas maneiras uma aplicação. Quanto mais tardia é deteção de falhas de segurança, maior o prejuízo para todos envolvidos.

É com esta premissa que este projeto apresentou uma experiência de desenvolvimento de uma aplicação descentralizada com o objetivo de gerir documentos de forma descentralizada através de uma *Blockchain*, que garanta não só a segurança dos dados manipulados pela aplicação na *Blockchain* como também a segurança da própria aplicação.

O projeto BudgetDApp, como referido no parágrafo anterior, tem uma preocupação com a segurança dos dados em dois aspetos diferentes. Primeiramente, apresentar a ideia de uma aplicação de troca de orçamentos em um ambiente onde a imutabilidade dos dados e outros aspetos de segurança são importantes. Em segundo lugar, apresentar formas de aplicar medidas de segurança durante o processo de conceção do projeto. Esta etapa também é fundamental, visto que Blockchain é uma tecnologia que pode proporcionar camadas conceituais que fortalecem a segurança, mas que precisa de uma série de reforços para que estas características sejam aproveitadas da melhor forma. A utilização de metodologias que envolvem boas práticas e ferramentas para análise dos diversos componentes da aplicação reforçam a ideia de se obter segurança em vários aspetos na idealização da BudgetDApp.

Coimbra Business School | ISCAC
Quinta Agrícola – Bencanta, 3045-601 Coimbra
Tel +351 239 802 000; E-mail: secretariado@iscac.pt; www.iscac.pt

Com contributos deste trabalho temos a aplicação desenvolvida e uma primeira proposta para o processo de desenvolvimento de aplicações descentralizadas, ainda que numa fase muito inicial, apresentado na Figura 2.2.

A aplicação desenvolvida passou por duas iterações no seu desenvolvimento, os quais representam as adequações para que o objetivo do projeto fosse alcançado. Na primeira iteração toda a base da aplicação foi construída tendo por base a ideia inicial de guardar todos os dados na Blockchain e a utilização de um *token* específico para aceder à mesma. No entanto, por conta de limitações do ambiente que foram ao encontro da ideia inicial, e a não implementação do *token* próprio da aplicação. A segunda iteração passa por adequações na arquitetura para contornar todos os problemas encontrados durante a primeira iteração, chegando ao resultado final do projeto que, apesar das alterações estruturais, mantém a base de uma aplicação descentralizada e entrega o que foi planeado inicialmente.

Uma das dificuldades do projeto foi tentar trazer, mesmo que de forma básica, ferramentas e referenciais que pregam boas práticas para o desenvolvimento de contratos e aplicações web. É de total consciência que testes de segurança, políticas e boas práticas normalmente são executadas por equipas diferentes e com suas respetivas especialidades, entretanto o presente projeto mostra que é possível, mesmo para desenvolvedores não experientes, tomar medidas básicas para reduzir falhas.

As limitações do projeto concentraram-se principalmente durante a primeira iteração da aplicação desenvolvida. Apesar de passar por um estudo prévio para conhecimento do ambiente, limitações estruturais da rede *Ethereum* foram fundamentais para a alteração descrita anteriormente. A forma com que a plataforma trata a utilização de variáveis globais e os custos associados a estas operações impediram a utilização da ideia inicial, que apresentava um conjunto de dados baseados em estruturas que necessitavam de ocupar espaço e profundidade que geraram custos e erros dentro da aplicação.

A outra limitação dentro do projeto envolve a utilização de um *token* próprio da aplicação. A criação do *token* foi efetuada e foi possível efetuar testes isolados dentro da primeira iteração, entretanto a implementação não foi possível dentro do prazo pensado para o projeto.

Coimbra Business School | ISCAC
Quinta Agrícola – Bencanta, 3045-601 Coimbra
Tel +351 239 802 000; E-mail: secretariado@iscac.pt; www.iscac.pt

Como trabalhos futuros, a implementação do *token* pode ser retomada e, apesar de envolver conceitos relacionados à forma de tornar moedas atrativas e com valor, podem representar uma mais valia interessante para o projeto. Outro ponto a ser incorporado ao projeto futuramente é a inserção de um conjunto de ferramentas de testes, com o objetivo de aumentar o rigor dos testes durante a conceção, e consequentemente a efetividade do modelo proposto. Para além disto, a revisão dos códigos por uma equipa de desenvolvimento também é um ponto interessante, visto que este projeto não foi elaborado por um especialista na área e muita coisa pode ser melhorada e simplificada.

Coimbra Business School | ISCAC
Quinta Agrícola – Bencanta, 3045-601 Coimbra
Tel +351 239 802 000; E-mail: secretariado@iscac.pt; www.iscac.pt

REFERÊNCIAS

Ahmed, M., & Pathan, A. S. K. (2020). Blockchain: Can It Be Trusted? *Computer*, *53*(4), 31–35. https://doi.org/10.1109/MC.2019.2922950

Ali Syed, T., Alzahrani, A., Jan, S., Siddiqui, M. S., Nadeem, A., & Alghamdi, T. (2019). A Comparative Analysis of Blockchain Architecture and its Applications: Problems and Recommendations. *IEEE Access*, *7*, 176838–176869. https://doi.org/10.1109/ACCESS.2019.2957660

Anita, N. (2019). *Blockchain Security Attack : A Brief Survey*. 6–11.

Bamakan, S. M. H., Motavali, A., & Babaei Bondarti, A. (2020). A survey of blockchain consensus algorithms performance evaluation criteria. *Expert Systems with Applications*, *154*, 113385. https://doi.org/10.1016/j.eswa.2020.113385

Belotti, M., Božić, N., Pujolle, G., & Secci, S. (2019). A Vademecum on Blockchain Technologies: When, Which, and How. *IEEE Communications Surveys and Tutorials*, *21*(4), 3796–3838. https://doi.org/10.1109/COMST.2019.2928178

Bhushan, B., Sinha, P., Sagayam, K. M., & J, A. (2021). Untangling blockchain technology: A survey on state of the art, security threats, privacy services, applications and future research directions. *Computers & Electrical Engineering*, *90*, 106897. https://doi.org/10.1016/j.compeleceng.2020.106897

Bhutta, M. N. M., Khwaja, A. A., Nadeem, A., Ahmad, H. F., Khan, M. K., Hanif, M. A., Song, H., Alshamari, M., & Cao, Y. (2021). A Survey on Blockchain Technology: Evolution, Architecture and Security. *IEEE Access*, *9*, 61048–61073. https://doi.org/10.1109/ACCESS.2021.3072849

Bouraga, S. (2021). A taxonomy of blockchain consensus protocols: A survey and classification framework. *Expert Systems with Applications*, *168*(November 2020), 114384. https://doi.org/10.1016/j.eswa.2020.114384

Buterin, V. (2014). A next-generation smart contract and decentralized application platform. *White Paper*, *3*(37), 1–2.

Chen, Y., Chen, S., Liang, J., Feagan, L. W., Han, W., Huang, S., & Wang, X. S. (2020).

Coimbra Business School | ISCAC
Quinta Agrícola – Bencanta, 3045-601 Coimbra
Tel +351 239 802 000; E-mail: secretariado@iscac.pt; www.iscac.pt

Decentralized data access control over consortium blockchains. *Information Systems*, *94*, 101590. https://doi.org/10.1016/j.is.2020.101590

Consensys. (2022). *Ethereum Smart Contract Best Practices*. https://consensys.github.io/smart-contract-best-practices/

ConsenSys Software Inc™. (2022). *MythX™*.

Devi, R. S. (2020). *using Ethical Hacking*. *Icoei*, 354–361.

Duran, R. G., Yarleque-Ruesta, D., Belles-Munoz, M., Jimenez-Viguer, A., & Munoz-Tapia, J. L. (2020). An Architecture for Easy Onboarding and Key Life-Cycle Management in Blockchain Applications. *IEEE Access*, *8*, 115005–115016. https://doi.org/10.1109/ACCESS.2020.3003995

Elnfar. (2022). *MERN_AUTH React Backend*. https://github.com/elnfar/MERN_AUTH/tree/main/server

Ethereum Revision. (2022a). *Solidity*.

Ethereum Revision. (2022b). *web3.js - Ethereum JavaScript API*. https://web3js.readthedocs.io/en/v1.8.0/

Ethereum Smart Contract Security Best Practices. (n.d.).

Gao, W., Hatcher, W. G., & Yu, W. (2018). A survey of blockchain: Techniques, applications, and challenges. *Proceedings - International Conference on Computer Communications and Networks, ICCCN*, *2018-July*(i), 1–11. https://doi.org/10.1109/ICCCN.2018.8487348

Gomathi, S., Soni, M., Dhiman, G., Govindaraj, R., & Kumar, P. (2021). A survey on applications and security issues of blockchain technology in business sectors. *Materials Today: Proceedings*, *xxxx*. https://doi.org/10.1016/j.matpr.2021.02.088

Guaman, D., Guaman, F., Jaramillo, D., & Sucunuta, M. (2017). Implementación de técnicas y recomendaciones de seguridad OWASP para evitar ataques de tipo inyección SQL, XSS utilizando J2EE y WS-Security. *Iberian Conference on Information Systems and Technologies, CISTI*. https://doi.org/10.23919/CISTI.2017.7975981

Guo, L., Xie, H., & Li, Y. (2020). Data encryption based blockchain and privacy preserving

mechanisms towards big data. *Journal of Visual Communication and Image Representation*, *70*, 102741. https://doi.org/10.1016/j.jvcir.2019.102741

Haber, S., & Scott Stornetta, W. (1991). How to time-stamp a digital document. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *537 LNCS*, 437–455. https://doi.org/10.1007/3-540-38424-3_32

Han, R., Yan, Z., Liang, X., & Yang, L. T. (2022). How Can Incentive Mechanisms and Blockchain Benefit with Each Other? A Survey. *ACM Computing Surveys*. https://doi.org/10.1145/3539604

Hassan, M. U., Rehmani, M. H., & Chen, J. (2019). Privacy preservation in blockchain based IoT systems: Integration issues, prospects, challenges, and future research directions. *Future Generation Computer Systems*, *97*, 512–529. https://doi.org/10.1016/j.future.2019.02.060

Hevner et al. (1996). Design science 97. *AI and Society*, *10*(2), 199–217. https://doi.org/10.1007/BF01205282

Homoliak, I., Venugopalan, S., Reijsbergen, D., Hum, Q., Schumi, R., & Szalachowski, P. (2021). The Security Reference Architecture for Blockchains: Toward a Standardized Model for Studying Vulnerabilities, Threats, and Defenses. *IEEE Communications Surveys & Tutorials*, *23*(1), 341–390. https://doi.org/10.1109/COMST.2020.3033665

Ji, S., Kim, D., & Im, H. (2021). Evaluating Countermeasures for Verifying the Integrity of Ethereum Smart Contract Applications. *IEEE Access*, *9*, 90029–90042. https://doi.org/10.1109/ACCESS.2021.3091317

Jonathan, K., & Sari, A. K. (2019). Security Issues and Vulnerabilities on A Blockchain System: A Review. *2019 2nd International Seminar on Research of Information Technology and Intelligent Systems, ISRITI 2019*, 228–232. https://doi.org/10.1109/ISRITI48646.2019.9034659

Kim, K. B., & Lee, J. (2020). Automated Generation of Test Cases for Smart Contract Security Analyzers. *IEEE Access*. https://doi.org/10.1109/ACCESS.2020.3039990

Coimbra Business School | ISCAC
Quinta Agrícola – Bencanta, 3045-601 Coimbra
Tel +351 239 802 000; E-mail: secretariado@iscac.pt; www.iscac.pt

Kong, Y. R., & Yue, J. W. (2020). Game Studies on Accounting Entity Behavior Based on Blockchain Technology. *ACM International Conference Proceeding Series*, 512–518. https://doi.org/10.1145/3444370.3444621

Kunda, M. A., & Alsmadi, I. (2022). *Practical web security testing : Evolution of web application modules and open source testing tools*. 152–155.

Li, T., Chen, Y., Wang, Y., Wang, Y., Zhao, M., Zhu, H., Tian, Y., Yu, X., & Yang, Y. (2020). Rational Protocols and Attacks in Blockchain System. *Security and Communication Networks*, *2020*. https://doi.org/10.1155/2020/8839047

Liu, Z., Luong, N. C., Wang, W., Niyato, D., Wang, P., Liang, Y. C., & Kim, D. I. (2019). A Survey on Blockchain: A Game Theoretical Perspective. *IEEE Access*, *7*, 47615–47643. https://doi.org/10.1109/ACCESS.2019.2909924

Lloyd, W. S. (1994). Exploring the Byzantine generals problem with beginning computer science students. *ACM SIGCSE Bulletin*, *26*(4), 21–24. https://doi.org/10.1145/190650.190656

Monrat, A. A., Schelén, O., & Andersson, K. (2019). A survey of blockchain from the perspectives of applications, challenges, and opportunities. *IEEE Access*, *7*, 117134–117151. https://doi.org/10.1109/ACCESS.2019.2936094

Moore, R. (2021). *Ethers*. https://docs.ethers.io/v5/

Moubarak, J., Filiol, E., & Chamoun, M. (2018). On blockchain security and relevant attacks. *2018 IEEE Middle East and North Africa Communications Conference, MENACOMM 2018*, 1–6. https://doi.org/10.1109/MENACOMM.2018.8371010

Mozilla.org. (2022). *X-Frame-Options*. https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options

Nakamoto, S. (2008). *Bitcoin: A Peer-to-Peer Electronic Cash System*. www.bitcoin.org

OWASP. (2022a). *OWASP Zed Attack Proxy (ZAP)*. https://www.zaproxy.org/

OWASP. (2022b). *The Web Security Testing Framework*. https://owasp.org/www-project-web-security-testing-guide/stable/3-The_OWASP_Testing_Framework/0-The_Web_Security_Testing_Framework

Coimbra Business School | ISCAC
Quinta Agrícola – Bencanta, 3045-601 Coimbra
Tel +351 239 802 000; E-mail: secretariado@iscac.pt; www.iscac.pt

Praitheeshan, P., Pan, L., Yu, J., Liu, J., & Doss, R. (2019). *Security Analysis Methods on Ethereum Smart Contract Vulnerabilities: A Survey*. 1–21.

REMIX. (2022).

Ren, M., Ma, F., Yin, Z., Fu, Y., Li, H., Chang, W., & Jiang, Y. (2021). Making smart contract development more secure and easier. In *ESEC/FSE 2021 - Proceedings of the 29th ACM Joint Meeting European Software Engineering Conference and Symposium on the Foundations of Software Engineering* (Vol. 1, Issue 1). Association for Computing Machinery. https://doi.org/10.1145/3468264.3473929

Salman, T., Jain, R., & Gupta, L. (2018). Probabilistic Blockchains: A Blockchain Paradigm for Collaborative Decision-Making. *2018 9th IEEE Annual Ubiquitous Computing, Electronics and Mobile Communication Conference, UEMCON 2018, May 2019*, 457–465. https://doi.org/10.1109/UEMCON.2018.8796512

Sanka, A. I., & Cheung, R. C. C. (2021). A systematic review of blockchain scalability: Issues, solutions, analysis and future research. *Journal of Network and Computer Applications*, *195*(December 2020), 103232. https://doi.org/10.1016/j.jnca.2021.103232

Saxena, S., Bhushan, B., & Ahad, M. A. (2021). Blockchain based solutions to secure IoT: Background, integration trends and a way forward. *Journal of Network and Computer Applications*, *181*(December 2020), 103050. https://doi.org/10.1016/j.jnca.2021.103050

SmartContractSecurity. (2022). *SWC Registry*.

Song, L. H., Li, T., & Wang, Y. L. (2019). Applications of game theory in blockchain. *Journal of Cryptologic Research*, *6*(1), 100–111. https://doi.org/10.13868/j.cnki.jcr.000287

Taveira, E. E. L. (2020). *Uma Visão sobre a Tecnologia Blockchain: Domínios de Aplicação e Especificidades na Cadeia de Abastecimento* [Instituto Superior de Engenharia do Porto]. https://recipp.ipp.pt/handle/10400.22/16755

Trojanowska, N., Kedziora, M., Hanif, M., & Song, H. (2020). Secure Decentralized

Coimbra Business School | ISCAC
Quinta Agrícola – Bencanta, 3045-601 Coimbra
Tel +351 239 802 000; E-mail: secretariado@iscac.pt; www.iscac.pt

Application Development of Blockchain-based Games. *2020 IEEE 39th International Performance Computing and Communications Conference, IPCCC 2020*, 1–8. https://doi.org/10.1109/IPCCC50635.2020.9391556

vom Brocke, J., Hevner, A., & Maedche, A. (2020). *Introduction to Design Science Research* (pp. 1–13). https://doi.org/10.1007/978-3-030-46781-4_1

W3C, W. W. W. C. (2022). *Content Security Policy Level 3*. https://w3c.github.io/webappsec-csp/

Wang, Y., Gou, G., Liu, C., Cui, M., Li, Z., & Xiong, G. (2021). Survey of security supervision on blockchain from the perspective of technology. *Journal of Information Security and Applications*, *60*(May), 102859. https://doi.org/10.1016/j.jisa.2021.102859

Wen, Y., Lu, F., Liu, Y., & Huang, X. (2021). Attacks and countermeasures on blockchains: A survey from layering perspective. *Computer Networks*, *191*(January), 107978. https://doi.org/10.1016/j.comnet.2021.107978

Wohrer, M., Zdun, U., & Rinderle-Ma, S. (2021). Architecture Design of Blockchain-Based Applications. *2021 3rd Conference on Blockchain Research and Applications for Innovative Networks and Services, BRAINS 2021*, 173–180. https://doi.org/10.1109/BRAINS52497.2021.9569813

Zhang, R., Xue, R., & Liu, L. (2019). Security and privacy on blockchain. *ACM Computing Surveys*, *52*(3). https://doi.org/10.1145/3316481

Zhdarkin, E., & Anureev, I. (2021). Development and Verification of Smart-Contracts for the ScientificCoin Platform. *International Conference of Young Specialists on Micro/Nanotechnologies and Electron Devices, EDM*, *2021-June*, 528–532. https://doi.org/10.1109/EDM52169.2021.9507717

Coimbra Business School | ISCAC
Quinta Agrícola – Bencanta, 3045-601 Coimbra
Tel +351 239 802 000; E-mail: secretariado@iscac.pt; www.iscac.pt