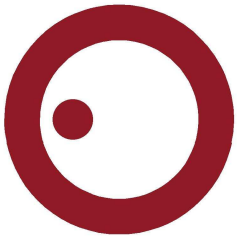


escola superior de tecnologia e gestão
instituto politécnico de leiria

Instituto Politécnico de Leiria
Escola Superior de Tecnologia e Gestão
Departamento de Engenharia Informática
Mestrado em Cibersegurança e Informática Forense

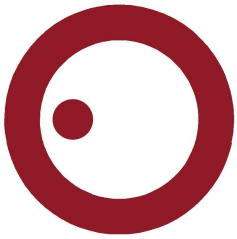
MEDBLOCK: DESIGN E DESENVOLVIMENTO DE UMA PLATAFORMA BASEADA EM BLOCKCHAIN PARA PARTILHA SEGURA DE DADOS DE SAÚDE

ESTUDANTE JOSÉ DIOGO PALMA IRIO

Leiria, junho de 2026

[9 de junho de 2026]

[9 de junho de 2026]



escola superior de tecnologia e gestão
instituto politécnico de leiria

Instituto Politécnico de Leiria
Escola Superior de Tecnologia e Gestão
Departamento de Engenharia Informática
Mestrado em Cibersegurança e Informática Forense

MEDBLOCK: DESIGN E DESENVOLVIMENTO DE UMA PLATAFORMA BASEADA EM BLOCKCHAIN PARA PARTILHA SEGURA DE DADOS DE SAÚDE

ESTUDANTE JOSÉ DIOGO PALMA IRIO
Número: 2230053

Projeto realizado sob orientação do Professor Carlos Jorge Machado Antunes(carlos.machado@ipleiria.pt).

Leiria, junho de 2026

[9 de junho de 2026]

[9 de junho de 2026]

AGRADECIMENTOS

Em primeiro lugar, gostaria de agradecer ao Professor Carlos Jorge Machado Antunes pela orientação técnica, pela disponibilidade constante e pelo rigor crítico com que acompanhou cada etapa deste trabalho. As suas observações e sugestões foram determinantes para a qualidade e término desta tese.

Gostaria também de agradecer a minha mulher e filho, por serem a minha maior inspiração e força para que nos momentos mais difíceis conseguisse continuar a elaboração deste trabalho chegando agora à sua conclusão.

Quero também agradecer à minha avó por sempre estar presente a todos os níveis e por também ser crucial na elaboração deste trabalho.

Por fim a todos os meus amigos e família agradeço por estarem sempre presentes e por tornarem este percurso académico mais fácil de ser superado.

[9 de junho de 2026]

RESUMO

A fragmentação dos dados de saúde constitui uma falha estrutural nos sistemas nacionais de saúde. As redes de prestadores de cuidados evoluíram historicamente em isolamento técnico. Portugal ilustra bem esta condição. Hospitais, centros de saúde e outras unidades de especialidade utilizam sistemas de registo que não comunicam entre si. Esta lacuna compromete de forma persistente tanto a segurança do doente como a integridade dos dados.

Esta tese propõe, implementa e avalia o MedBlock, uma plataforma baseada em *blockchain* com acesso permissionado, assente no Hyperledger Fabric. Esta *framework open-source* e, mantida pela Linux Foundation, é orientada para ambientes empresariais que requerem controlos de acesso, privacidade de dados e governação dos mesmos. A plataforma foi concebida para colmatar estas deficiências mediante a aplicação de uma tecnologia de registo distribuído e partilha segura de registos eletrónicos de saúde. A plataforma foi avaliada face a três objetivos de segurança: impedir a modificação não autorizada de registos, garantir a rastreabilidade de cada evento de acesso a dados e aplicar a diferenciação de papéis entre doentes e organizações de saúde. Os resultados confirmam o cumprimento destes objetivos. A integridade dos registos é assegurada por mecanismos criptográficos, cada acesso gera uma cadeia de auditoria imutável e o controlo de papéis é imposto pela própria rede.

Este trabalho apresenta três contribuições distintas. Em primeiro lugar, produz uma arquitetura de referência validada para a partilha de dados de saúde baseada em *blockchain*, enquadrada nas exigências do RGPD e no contexto português. Em segundo lugar, o protótipo MedBlock demonstra que a arquitetura é tecnicamente concretizável e não apenas teoricamente coerente, o que constitui evidência funcional da sua viabilidade. Em terceiro lugar, a implementação aborda barreiras concretas, como a incerteza regulatória, a integração com sistemas mais antigos e a adoção por parte dos profissionais de saúde.

No conjunto, estes resultados oferecem um modelo transferível: outros sistemas nacionais de saúde que enfrentem infraestruturas fragmentadas e legislação rigorosa em matéria de proteção de dados podem adaptar esta arquitetura e este roteiro sem necessidade de partir do zero.

[9 de junho de 2026]

ABSTRACT

Healthcare data fragmentation remains a structural liability in national health systems. Provider networks have historically evolved in technical isolation. Portugal exemplifies this condition. Hospitals, primary care units, and other centres use record systems that do not communicate with one another. This gap places both patient safety and data integrity under persistent strain.

This thesis proposes and evaluates MedBlock, a permissioned *blockchain*-based platform built on Hyperledger Fabric. This *open-source* framework, maintained by the Linux Foundation, is designed for enterprise environments that require access controls, data privacy, and configurable governance. The platform was conceived to address these deficiencies by applying distributed ledger technology to the secure sharing of electronic health records. The platform was evaluated against three security objectives: preventing unauthorised record modification, ensuring the traceability of every data access event, and enforcing role differentiation between patients and healthcare organisations. Evaluation results confirm that these objectives were met. Record integrity is ensured through cryptographic mechanisms, every access generates an immutable audit trail, and role control is enforced by the network itself.

This work yields three distinct contributions. First, it produces a validated reference architecture for *blockchain*-based health data sharing, framed within the requirements of the GPDR and the Portuguese context. Second, the MedBlock prototype demonstrates that the architecture is technically realisable, not just theoretically sound, providing functional evidence of its feasibility. Third, the implementation addresses concrete barriers such as regulatory uncertainty, legacy system integration, and adoption by healthcare professionals.

Collectively, these outputs offer a transferable model: other national health systems facing fragmented infrastructure and strict data protection legislation can adapt this architecture and roadmap without having to start from scratch.

[9 de junho de 2026]

ÍNDICE

AGRADECIMENTOS i

RESUMO iii

ABSTRACT v

ÍNDICE vii

LISTA DE FIGURAS ix

LISTA DE TABELAS xi

LISTA DE LISTAGENS xi

LISTA DE ABREVIATURAS xv

1 INTRODUÇÃO 1

1.1 Contexto e Motivação 1

1.1.1 Transformação Digital para a Gestão de Dados de Saúde . . . 1

1.1.2 Ameaças crescentes à segurança dos dados na área da saúde . 3

1.1.3 Fragmentação de dados no ecossistema de saúde português . 4

1.2 Descrição do problema 6

1.2.1 Convergência de desafios não resolvidos 6

1.2.2 Imutabilidade versus direito ao esquecimento 7

1.3 Objetivos 8

1.3.1 Objetivo Geral 9

1.3.2 Objetivos específicos 9

1.4 Âmbito 10

1.4.1 O que a tese abrange 10

1.4.2 O que a tese não abrange 11

1.5 Contribuições esperadas 12

1.6 Organização do Documento 13

2 ESTADO DA ARTE 15

2.1 Tecnologia *blockchain*: fundamentos 15

2.1.1 Origem e evolução conceptual 15

2.1.2 Estrutura do *ledger* distribuído 16

2.1.3 Propriedades de arquitetura 17

2.1.4 Aplicabilidade da *blockchain* 17

2.2 Modelos de rede e mecanismos de consenso 18
2.3 *Smart contracts* 19
2.4 Hyperledger Fabric: arquitetura e componentes 20
2.4.1 Visão geral da arquitetura modular 20
2.4.2 Canais 21
2.4.3 *Peers* 21
2.4.4 *Ordering service* 22
2.4.5 *Chaincode* 23
2.4.6 *Membership Service Provider* 23
2.4.7 Autenticação X.509: paralelo entre MSP e Autenticação.Gov 24
2.5 Contexto português 25
2.6 Motivação para a investigação 25

3 TRABALHOS RELACIONADOS 27
3.1 Sistemas de registos de saúde eletrónicos baseados em *blockchain* . . 27
3.1.1 *Blockchain* como base para a partilha de dados na área da saúde 28
3.1.2 MedRec: Gestão de registos centralizada do paciente em Ethereum 28
3.1.3 MedChain: Arquitetura híbrida de blockchain P2P 29
3.1.4 Partilha de dados de saúde baseada em *blockchain* com consideração pelas normas regulamentares 30
3.1.5 *Blockchain* de consórcio com encriptação baseada em atributos 30
3.2 Plataformas de *blockchain* permissionadas: avaliação comparativa . . 31
3.3 O Hyperledger Fabric na área da saúde 32
3.3.1 Hospital Provincial Frere: Implementação Empírica 33
3.4 Análise comparativa e posicionamento do MedBlock 34
3.5 Resumo do capítulo 35
4 METODOLOGIA 37
4.1 Metodologia de Investigação: Design Science Research 37
4.2 Identificação do Problema e Levantamento de Requisitos 38
4.2.1 Seleção da Norma de Dados Clínicos 39
4.2.2 Seleção de Identidade e Autenticação 40
4.3 Decisões de *Design* Arquitetural 41
4.3.1 Justificação da adoção de *blockchain* 41
4.3.2 Estratégia de Separação de Dados *On-Chain* e *Off-Chain* . . 42
4.3.3 *Design* da Topologia de Rede 43
4.3.4 Estratégia de Canal de *Staging* 44

4.3.5 Estratégia de *Design* do *Chaincode* . . . 45
4.3.6 *Design* da Arquitetura de Segurança . . . 47
4.4 Estratégia de Conformidade Regulamentar . . . 48
4.5 Estratégia de Avaliação . . . 49
4.5.1 Princípio de Testes com Prioridade para o *Staging* . . . 49
4.5.2 Abordagem à Avaliação de Desempenho . . . 49
4.5.3 Abordagem à Avaliação de Segurança . . . 50
4.6 Ambiente de Desenvolvimento e Ferramentas . . . 50
4.7 Cronograma do Projeto . . . 51

5 DESENVOLVIMENTO 55
5.1 Infraestrutura de Rede e Contentorização . . . 55
5.1.1 Infraestrutura de Alojamento . . . 56
5.1.2 Contentorização . . . 57
5.2 *Public Key Infrastructure* e Hierarquia de *Certificate Authority* . . . 59
5.3 Configuração do canal e Geração do *Genesis Block* . . . 61
5.4 Implementação de *Chaincode*: Gestão de Consentimento . . . 63
5.5 Camada de Dados Clínicos *Off-Chain*: Servidores FHIR e Arquitetura de *Reverse Proxy* Nginx . . . 65
5.5.1 *Deployment* do Servidor HAPI FHIR . . . 65
5.5.2 Configuração do *Reverse Proxy* Nginx para *Endpoints* FHIR . . . 67
5.6 Aplicação *Web*: Implementação do *Backend* . . . 69
5.6.1 *Gateway Connection* e Integração do Fabric SDK . . . 69
5.6.2 Autenticação: Integração CMD e Provisionamento de Identidade 70
5.6.3 Gestão de Sessões: Implementação de *Tokens* JWE . . . 72
5.6.4 Camada de Obtenção de Dados FHIR . . . 73
5.6.5 *Design* da REST API e Estrutura de Rotas . . . 74
5.7 Aplicação *Web*: Implementação do *Frontend* . . . 76
5.8 Detalhes de Implementação de Segurança . . . 80
5.8.1 Controlos de Segurança na Camada Aplicacional . . . 81
5.8.2 Segurança de Perímetro: Integração Cloudflare WAF . . . 81
5.9 Resumo do capítulo . . . 85

6 TESTES E RESULTADOS 87
6.1 Avaliação de Desempenho: *Benchmark* Hyperledger Caliper . . . 87
6.1.1 Configuração do Ambiente de Teste . . . 87
6.1.2 Resultados de *Throughput* . . . 88
6.1.3 Resultados de Latência . . . 90
6.1.4 Discussão e Contextualização dos Resultados . . . 92

6.2 Avaliação de Segurança 93
6.2.1 Análise Estática de Código: SonarQube 93
6.2.2 Análise Dinâmica: OWASP ZAP 95
6.2.3 Análise de Vulnerabilidades: Nessus Professional 97
6.3 Resumo do Capítulo 99

7 CONCLUSÕES 101
7.1 Limitações 103
7.2 Perspetivas futuras 104

BIBLIOGRAFIA 107

APÊNDICES

A APÊNDICE A 113
A.1 Código-Fonte do Chaincode `org-authorizations` 113

DECLARAÇÃO 117

LISTA DE FIGURAS

Figura 2.1 Árvore de decisão para determinar a aplicabilidade de *blockchain*. 18
Figura 4.1 Topologia da rede da plataforma MedBlock. 44
Figura 4.2 Diagrama de Gantt com o cronograma das fases da elaboração desta tese. 53
Figura 5.1 Topologia de *containers* Docker da rede MedBlock. 56
Figura 5.2 Hierarquia da infraestrutura de chaves públicas das entidades de certificação. 60
Figura 5.3 Estrutura dos perfis do `configtx.yaml` 62
Figura 5.4 Diagrama de sequência do fluxo de um pedido clínico desde a aplicação MedBlock até ao servidor HAPI FHIR. 68
Figura 5.5 Diagrama de sequência do fluxo de autenticação do paciente na plataforma MedBlock. 70
Figura 5.6 Página de *login* de testes do paciente. 77
Figura 5.7 Página de autenticação via Chave Móvel Digital no portal do paciente. 77
Figura 5.8 Página de perfil do paciente no portal da plataforma MedBlock. 78
Figura 5.9 Página de consultas clínicas do paciente no portal da plataforma MedBlock. 78
Figura 5.10 Página de exames do paciente no portal da plataforma MedBlock. 78
Figura 5.11 Página de gestão de autorizações do paciente no portal da plataforma MedBlock. 79
Figura 5.12 Página de *login* da organização no portal da plataforma MedBlock. 79
Figura 5.13 Página de pesquisa de paciente por número nacional de saúde no portal da organização, com acesso concedido. 80
Figura 5.14 Página de pesquisa de paciente por número nacional de saúde no portal da organização, com acesso negado. 80
Figura 5.15 Diagrama de segurança em camadas ilustrando o perímetro de segurança completo da plataforma MedBlock. 84

Figura 6.1 Throughput obtido ao longo de cinco níveis progressivos de carga para o *chaincode* `org-authorizations` na rede Hyperledger Fabric 2.5 do MedBlock. 89
Figura 6.2 Latência média e máxima das transações ao longo de cinco níveis de carga. 91
Figura 6.3 Resultados da análise estática SonarQube do projeto `med-block`, correspondente à aplicação *web*. 94
Figura 6.4 Resultados da análise estática SonarQube do projeto `org-authorizations`, correspondente ao *chaincode*. 94
Figura 6.5 Matriz de alertas por nível de risco e confiança resultante do *scan* automatizado OWASP ZAP à aplicação *web* MedBlock. 95
Figura 6.6 Distribuição dos alertas por risco e número de ocorrências resultante do *scan* automatizado OWASP ZAP à aplicação *web* MedBlock. 96
Figura 6.7 Resultado da verificação de cabeçalhos de segurança HTTP com a ferramenta shcheck 96
Figura 6.8 Filtro aplicado aos resultados dos *scans* Tenable Nessus Professional. 97
Figura 6.9 Resultado do *scan* Tenable Nessus Professional inicial à aplicação *web* MedBlock 98
Figura 6.10 Resultado do *scan* Tenable Nessus Professional inicial ao servidor. 98
Figura 6.11 Resultado do *scan* Tenable Nessus Professional pós-remediação à aplicação *web* MedBlock 99
Figura 6.12 Resultado do *scan* Tenable Nessus Professional pós-remediação ao servidor. 99

LISTA DE TABELAS

Tabela 3.1 Resumo comparativo dos sistemas revistos de partilha de dados de saúde baseados em *blockchain*. 34
Tabela 5.1 Resumo dos containers Docker da rede MedBlock 58
Tabela 5.2 Resumo dos *endpoints* da interface aplicacional *REST* da plataforma MedBlock. 75
Tabela 5.3 Configurações de segurança da Cloudflare 82
Tabela 6.1 Configuração dos cinco níveis de carga utilizados no *benchmark* Hyperledger Caliper. 88
Tabela 6.2 Resultados do *benchmark* Hyperledger Caliper ao longo de cinco níveis de carga. 89

[9 de junho de 2026]

LISTA DE LISTAGENS

Listagem 1 Geração do bloco génesis com `configtxgen`. 62
Listagem 2 Geração da transação de criação do canal de aplicação `medblockchain`. 62
Listagem 3 Amostra doódigo-fonte do *smart contract* `OrgAuthorizationsContract`. 63
Listagem 4 Totalidade do código-fonte do *smart contract* `OrgAuthorizationsContract`. 113

[9 de junho de 2026]

LISTA DE ABREVIATURAS

ABE	*Attribute-Based Encryption.*
AMA	Agência para a Modernização Administrativa.
API	*Application Programming Interface.*
BFT	*Byzantine Fault Tolerance.*
BFT-SMaRt	*Byzantine Fault Tolerance State Machine Replication.*
CA	*Certificate Authority.*
CDSS	*Clinical Decision Support System.*
CLI	*Command Line Interface.*
CMD	Chave Móvel Digital.
CSP	*Content Security Policy.*
CSRF	*Cross-Site Request Forgery.*
CSS	*Cascading Style Sheets.*
DDoS	*Distributed Denial of Service.*
DHT	*Distributed Hash Table.*
DNS	*Domain Name System.*
DSRM	*Design Science Research Methodology.*
EDPB	*European Data Protection Board.*
EEDS	Espaço Europeu de Dados de Saúde.
EHDS	*European Health Data Space.*
EHR	*Electronic Health Record.*
eIDAS	*electronic IDentification, Authentication and trust Services.*
EVM	*Ethereum Virtual Machine.*
EVM	*Ethereum Virtual Machine.*

FHIR	*Fast Healthcare Interoperability Resources.*
GPDR	*General Data Protection Regulation.*
gRPC	*Google Remote Procedure Call.*
HAPI	HAPI FHIR (implementação de referência *open-source* do HL7 FHIR).
HIPAA	*Health Insurance Portability and Accountability Act.*
HL7	*Health Level Seven International.*
HSTS	*HTTP Strict Transport Security.*
HTML	*HyperText Markup Language.*
HTTP	*Hypertext Transfer Protocol.*
HTTPS	*Hypertext Transfer Protocol Secure.*
IA	Inteligência Artificial.
IBM	*International Business Machines.*
IETF	*Internet Engineering Task Force.*
IoT	*Internet of Things.*
ITU-T	*International Telecommunication Union, Telecommunication Standardization Sector.*
JDBC	*Java Database Connectivity.*
JSON	*JavaScript Object Notation.*
JWE	*JSON Web Encryption.*
JWT	*JSON Web Token.*
MIT	*Massachusetts Institute of Technology.*
MSP	*Membership Service Provider.*
mTLS	*mutual Transport Layer Security.*
NIST	*National Institute of Standards and Technology.*
NSNS	Número Serviço Nacional de Saúde.
OAuth 2.0	*Open Authorization 2.0.*
OWASP	*Open Worldwide Application Security Project.*

P2P	*Peer-to-Peer*.
PBFT	*Practical Byzantine Fault Tolerance*.
PKI	*Public Key Infrastructure*.
PoS	*Proof-of-Stake*.
PoW	*Proof-of-Work*.
PPR	*Patient-Provider Relationship*.
RBAC	*Role-Based Access Control*.
RES	Registos Eletrónicos de Saúde.
REST	*Representational State Transfer*.
RGPD	Regulamento Geral sobre a Proteção de Dados.
RSE	Registo de Saúde Eletrónico.
SDK	*Software Development Kit*.
SNS	Serviço Nacional de Saúde.
SPMS	Serviços Partilhados do Ministério da Saúde.
SQL	*Structured Query Language*.
TLS	*Transport Layer Security*.
TPS	*Transactions Per Second*.
UE	União Europeia.
URI	*Uniform Resource Identifier*.
VPS	*Virtual Private Server*.
WAF	*Web Application Firewall*.
XSS	*Cross-Site Scripting*.

1

INTRODUÇÃO

A fragmentação dos dados de saúde constitui uma falha estrutural nos sistemas nacionais de saúde. As redes de prestadores de cuidados evoluíram historicamente em isolamento técnico. Portugal ilustra bem esta condição. Hospitais, centros de saúde e outras unidades de especialidade utilizam sistemas de registo que não comunicam entre si. Esta lacuna compromete de forma persistente tanto a segurança do doente como a integridade dos dados.

Este capítulo enquadra o problema que motivou o desenvolvimento desta tese e fixa os parâmetros sob os quais o trabalho foi conduzido. A apresentação parte do contexto tecnológico e regulamentar que envolve a partilha de dados de saúde em Portugal, formaliza o problema de investigação que decorre da convergência entre digitalização clínica, escalada de ameaças à cibersegurança e fragmentação institucional.

1.1 CONTEXTO E MOTIVAÇÃO

A motivação para este trabalho surge da convergência de três fenómenos: a digitalização acelerada dos processos clínicos, o crescimento das ameaças à segurança dos dados de saúde e a fragmentação persistente do ecossistema digital português. As subsecções seguintes examinam cada um destes fenómenos de forma individual antes de apresentar a tecnologia *blockchain* como resposta integrada a estes desafios.

1.1.1 *Transformação Digital para a Gestão de Dados de Saúde*

A prestação de cuidados de saúde passou por uma profunda mudança estrutural nas últimas duas décadas. Essa mudança foi impulsionada, em grande parte, pela proliferação de tecnologias digitais. Atualmente, essas tecnologias facilitam os fluxos de trabalho clínicos e os processos administrativos e também melhoram os canais de comunicação com os pacientes. Os Registos Eletrónicos de Saúde (RES), as plataformas de telemedicina e as aplicações móveis de saúde redefiniram a forma

como os dados dos pacientes são gerados, documentados e trocados entre instituições (Xi et al., 2022). Esses sistemas são fundamentais para reduzir as ineficiências do papel. No entanto, também trouxeram desafios complexos de interoperabilidade, segurança e privacidade. Os sistemas nacionais de saúde continuam a enfrentar esses desafios (Theodouli et al., 2018). A transformação não é apenas incremental. Um único atendimento num hospital moderno pode gerar centenas de pontos de dados discretos. Estes incluem imagens de diagnóstico, resultados laboratoriais, registos de prescrições e notas clínicas. Cada ponto de dados deve ser armazenado, transmitido e recuperado com precisão e confidencialidade (Harman et al., 2012).

No contexto europeu, o Regulamento Geral sobre a Proteção de Dados (RGPD), promulgado ao abrigo do Regulamento 2016/679, estabeleceu um quadro jurídico rigoroso para o tratamento de dados pessoais. Os dados de saúde recebem classificação específica no artigo 9.º, o que justifica proteção reforçada . Os artigos 5.º e 25.º impõem a minimização de dados, a limitação da finalidade e a privacidade desde a conceção. O artigo 32.º exige que os responsáveis e os subcontratantes implementem medidas técnicas e organizacionais adequadas, incluindo a encriptação e a pseudonimização. Essas medidas devem garantir segurança proporcional ao risco (European Parliament and Council of the European Union, 2016). O artigo 17.º consagra o direito ao esquecimento, levantando questões arquitetónicas para sistemas baseados em estruturas de dados imutáveis. Podemos então concluir que as exigências regulamentares têm um forte poder de execução. O *European Data Protection Board* (EDPB) já impôs multas superiores a centenas de milhões de euros por incumprimento em todos os setores. Isso sublinha como as instituições de saúde operam sob pressão legal considerável para proteger as informações dos pacientes (European Data Protection Board (EDPB), 2026).

Simultaneamente, a *Health Insurance Portability and Accountability Act* (HIPAA) nos Estados Unidos estabeleceu precedentes internacionais influentes para a proteção de dados de saúde, estabelecendo salvaguardas obrigatórias para Informações de Saúde Protegidas nas vertentes eletrónica e física (U.S. Department of Health and Human Services, 1996). Embora a HIPAA não vincule diretamente as instituições europeias, as suas medidas de segurança e privacidade moldaram as melhores práticas globais e são frequentemente citadas como referências complementares por quem projeta plataformas de dados de saúde intercontinentais (U.S. Department of Health and Human Services, 1996; Xi et al., 2022). Para as instituições de saúde portuguesas que colaboram com parceiros de investigação internacionais ou atendem pacientes de várias jurisdições regulatórias, alcançar a dupla conformidade com

as normas alinhadas ao RGPD e à HIPAA constitui um requisito arquitetónico complexo que deve ser abordado na fase de elaboração do projeto.

1.1.2 *Ameaças crescentes à segurança dos dados na área da saúde*

O setor de saúde continua a ser o setor com maior custo financeiro em termos de remediação de violações de dados. De acordo com a vigésima edição do Relatório sobre o Custo de uma Violação de Dados da IBM e do Ponemon Institute, publicado em 2025 e baseado em 600 organizações afetadas entre março de 2024 e fevereiro de 2025, o custo médio de uma violação de dados na área da saúde foi de US$ 7,42 milhões, uma redução significativa em relação aos US$ 9,77 milhões registados em 2024, mas ainda assim substancialmente superior à média global entre todos os setores, de US$ 4,44 milhões (IBM Security and Ponemon Institute, 2025). A área da saúde ocupa esta posição há vários anos consecutivos, uma persistência que reflete vulnerabilidades profundamente estruturais, em vez de incidentes isolados. Além do custo financeiro, as violações de dados na área da saúde apresentaram uma duração, na identificação e na contenção, mais longa do que a média global de 241 dias, com 279 dias (IBM Security and Ponemon Institute, 2025). Essa janela de exposição prolongada oferece aos adversários uma oportunidade considerável de extrair dados clínicos confidenciais, escalar privilégios em redes comprometidas e interromper a prestação de cuidados antes que a organização detete o problema.

Vários fatores tornam os dados de saúde particularmente atraentes para os *threat actors* e, quando comprometidos, bastante valiosos. As informações pessoais identificáveis dos clientes, que, no contexto da saúde, abrangem dados demográficos, detalhes de seguros, históricos de diagnósticos e informações genéticas, foram a categoria de dados mais frequentemente afetada em 2025, comprometida em 53% de todas as violações globalmente, enquanto a propriedade intelectual teve o maior custo por registo, de 178 USD (IBM Security and Ponemon Institute, 2025). A pressão operacional sobre hospitais e redes clínicas para manter o atendimento ininterrupto aos pacientes aumenta ainda mais a probabilidade de pagamentos de resgate; o mesmo estudo de 2025 constatou que, embora 63% das vítimas de *ransomware* se recusassem a pagar, o custo médio de um incidente de extorsão ou *ransomware* permaneceu elevado, em 5,08 milhões USD, quando divulgado pelo invasor (IBM Security and Ponemon Institute, 2025). Além disso, o *phishing* ultrapassou as credenciais roubadas como o vetor de ataque inicial mais prevalente, com 16% de todas as violações e uma média de 4,8 milhões USD por incidente,

enquanto o comprometimento da *supply chain* surgiu como o segundo vetor mais comum, com 15% (IBM Security and Ponemon Institute, 2025). Estas conclusões sublinham que o panorama de ameaças que as instituições de saúde enfrentam está simultaneamente a alargar-se em termos de superfície de ataque e a aprofundar-se em termos de consequências financeiras.

A edição de 2025 deste relatório introduz ainda uma dimensão de particular relevância para as instituições que adotam tecnologias emergentes: as implicações de segurança da Inteligência Artificial (IA). Entre as organizações que sofreram um incidente de segurança relacionado com IA, 97% não possuíam controlos de acesso adequados nos seus sistemas de IA (IBM Security and Ponemon Institute, 2025). A adoção de ferramentas de IA sem aprovação ou supervisão organizacional esteve associada a 20% das violações e acrescentou uma média de 670 000 dólares aos custos dessas violações, em comparação com organizações com baixa ou nenhuma exposição à IA paralela (IBM Security and Ponemon Institute, 2025). Estas conclusões têm implicações diretas para as instituições de saúde que exploram plataformas de partilha de dados baseadas ou ampliadas por IA, uma vez que destacam a importância crítica de estruturas de governação que acompanhem a adoção tecnológica. No contexto regulatório europeu, as organizações de saúde que não implementarem medidas de proteção adequadas enfrentam multas de até 20 milhões de euros ou 4% da faturação global anual, conforme o Artigo 83(5) do RGPD (European Parliament and Council of the European Union, 2016). As arquiteturas convencionais de bases de dados centralizadas, ao concentrarem os dados dos pacientes em sistemas únicos, constituem tanto alvo para atacantes quanto um único ponto de falha (Cyran, 2018) (Azaria et al., 2016).

1.1.3 *Fragmentação de dados no ecossistema de saúde português*

O Serviço Nacional de Saúde (SNS) de Portugal oferece cobertura universal de saúde financiada principalmente por impostos, atendendo aproximadamente 10 milhões de cidadãos por meio de uma rede de hospitais públicos, centros de saúde e unidades clínicas especializadas (SPMS — Serviços Partilhados do Ministério da Saúde, 2026a). Os Serviços Partilhados do Ministério da Saúde (SPMS), criados em 2010 como agência de saúde digital e fornecedor centralizado de gestão de sistemas de informação para o Ministério da Saúde, têm sido fundamentais para o avanço da infraestrutura de saúde eletrónica em Portugal (Martins, 2020; SPMS — Serviços Partilhados do Ministério da Saúde, 2026a). Entre os avanços tecnológicos notáveis

incluem-se a implementação da prescrição eletrónica e a criação do Registo de Saúde Eletrónico (RSE), um portal de RES voltado para o cidadão que ultrapassou 2,25 milhões de utilizadores registados (Martins, 2020). Portugal foi também um dos primeiros países europeus a permitir a dispensa transfronteiriça de medicamentos por via eletrónica através do serviço MyHealth@EU, demonstrando um compromisso genuíno com a interoperabilidade transnacional dos dados de saúde (SPMS — Serviços Partilhados do Ministério da Saúde, 2026b).

Apesar destes avanços, persiste uma fragmentação significativa na superfície deste ecossistema digital. Os prestadores de cuidados de saúde privados, que atendem aproximadamente 20% da população portuguesa, não estão legalmente obrigados a partilhar os seus dados clínicos com o SNS (TEHDAS — Towards the European Health Data Space, 2023). Esta lacuna regulamentar cria uma assimetria estrutural de informação: um paciente que recebe um diagnóstico por imagem num hospital privado e é posteriormente encaminhado para um especialista público frequentemente tem de recorrer a cópias físicas ou a transferências de registos mediadas pelo próprio paciente, um processo ineficiente e suscetível a perdas de dados ou erros de transcrição. Dentro do próprio sistema público, coexistem aproximadamente 80 registos de dados e sistemas de informação distintos (TEHDAS — Towards the European Health Data Space, 2023), muitos dos quais empregam padrões de dados heterogéneos e foram desenvolvidos de forma independente ao longo de diferentes gerações tecnológicas. Embora a adoção do *Health Level Seven International* (HL7) V2.5 em Portugal, com uma migração planeada para o HL7 *Fast Healthcare Interoperability Resources* (FHIR), tenha melhorado a interoperabilidade, que garante que os conceitos clínicos tenham um significado idêntico em todos os sistemas, continua a ser um desafio por resolver (TEHDAS — Towards the European Health Data Space, 2023).

O envolvimento ativo do governo português no Regulamento do Espaço Europeu de Dados de Saúde (EEDS), um quadro regulatório publicado em março de 2025, com entrada em vigor prevista para 2027, reforça ainda mais a urgência de abordar estas questões de fragmentação (SPMS — Serviços Partilhados do Ministério da Saúde, 2023). O EEDS exige que os Estados-Membros criem um Organismo Responsável pelo Acesso a Dados de Saúde para facilitar tanto a utilização primária (cuidados diretos aos doentes) como a secundária (investigação, elaboração de políticas e inovação) dos dados eletrónicos de saúde. O SPMS já iniciou a ação HealthData@PT, uma iniciativa no âmbito do Programa EU4Health 2021-2027, para construir a infraestrutura técnica necessária à integração nacional no ecossistema HealthData@EU (SPMS — Serviços Partilhados do Ministério da Saúde, 2023). Neste panorama regulamentar

e técnico em evolução, a necessidade de uma arquitetura de partilha de dados que satisfaça simultaneamente os requisitos do RGPD, apoie a interoperabilidade transfronteiriça e colmate o fosso entre os setores público e privado nos cuidados de saúde portugueses é claramente articulada pelas partes interessadas.

1.2 DESCRIÇÃO DO PROBLEMA

A secção anterior contextualizou a transformação digital na saúde, as ameaças crescentes à segurança dos dados e a fragmentação do ecossistema português. A presente secção articula formalmente o problema de investigação, demonstrando como estes fenómenos convergem num desafio singular. Examina-se, em particular, a tensão entre a imutabilidade inerente à *blockchain* e o direito ao esquecimento consagrado no RGPD, bem como as condições específicas do contexto português que tornam este problema especialmente relevante.

1.2.1 *Convergência de desafios não resolvidos*

Na secção 1.1 foi estabelecido um panorama da gestão de dados de saúde através de três dimensões: fragmentação sistémica de dados no SNS português, um ambiente de ameaças à cibersegurança em que a saúde continua a ser o setor mais dispendioso em termos de violações (IBM Security and Ponemon Institute, 2025) e um quadro regulamentar, de acordo com o RGPD e complementado pela HIPAA, que impõe restrições rigorosas sobre a forma como os dados dos pacientes podem ser armazenados, partilhados e eliminados (European Parliament and Council of the European Union, 2016; U.S. Department of Health and Human Services, 1996). Cada dimensão foi examinada de forma independente; no entanto, o problema que esta tese aborda surge da sua interseção.

A fragmentação, por si só, não justifica uma solução baseada em *blockchain*. As plataformas de integração centralizadas, tais como os modelos de partilha de Informações de Saúde implementados na Dinamarca, na Estónia e em partes dos Estados Unidos, demonstraram que a interoperabilidade pode ser melhorada sem recorrer a tecnologias distribuídas (Xi et al., 2022). O que diferencia o caso português é a combinação da fragmentação com uma assimetria de governação entre prestadores públicos e privados, um ambiente regulatório que penaliza a proteção inadequada de dados com multas que podem chegar a 20 milhões de euros ou 4% do volume de negócios global anual e uma obrigação simultânea de se preparar para as exigências

de partilha transfronteiriça de dados do EEDS até 2027. Nenhuma plataforma de integração centralizada satisfaz os três requisitos, ao mesmo tempo em que fornece a possibilidade de auditoria inviolável e os mecanismos de consentimento controlados pelo paciente que a arquitetura de autorizações descentralizada descrita neste documento fornece (Azaria et al., 2016; Shen et al., 2019).

Por outro lado, a tecnologia de *blockchain*, por si só, não resolve todos os desafios mencionados. No capítulo 3, são apresentados vários protótipos que demonstraram uma viabilidade técnica apenas parcial. O problema não é, portanto, se a *blockchain* pode melhorar a partilha de dados de saúde em termos abstratos, mas sim se é possível desenhar e desenvolver uma arquitetura de autorizações em *blockchain* específica que satisfaça os requisitos concretos e simultâneos do ecossistema de saúde português: tratamento de dados em conformidade com o RGPD, integração com a infraestrutura de saúde digital e estruturas, que acomodem tanto a participação pública obrigatória quanto o envolvimento voluntário do setor privado.

1.2.2 *Imutabilidade versus direito ao esquecimento*

Anteriormente foi apresentado que a dificuldade de conciliar o artigo 17.º do RGPD, o direito ao esquecimento, com um modelo de dados baseado apenas em *blockchain*. Foi identificado brevemente o armazenamento de dados fora da cadeia como uma possível estratégia de mitigação. Esta secção examina esse desafio com maior profundidade, porque constitui a restrição de *design* mais consequente para qualquer plataforma de saúde baseada em *blockchain* implementada na jurisdição da União Europeia (UE).

Quando um paciente português exerce o direito ao apagamento contra um prestador de cuidados de saúde cujos registos estão ancorados numa *blockchain*, o responsável pelo tratamento enfrenta uma impossibilidade técnica: o encadeamento criptográfico que garante a integridade dos dados é o mesmo mecanismo que impede a eliminação retroativa. Xi et al., na sua revisão sistemática, identificaram este conflito como um dos principais desafios em aberto para a adoção da *blockchain* nos cuidados de saúde europeus (Xi et al., 2022).

Duas estratégias de mitigação para este problema ganharam alguma tração, embora nenhuma tenha sido definitiva. A primeira abordagem passa pela colocação de dados fora da *blockchain*, apenas armazenando *hashes* e metadados do acesso à *blockchain*, mantendo os registos reais dos pacientes em repositórios convencionais ou *Peer-to-Peer* (P2P) fora da *blockchain*, onde a eliminação é tecnicamente viável

(Shen et al., 2019). A destruição dos dados fora da cadeia torna a *hash* na *blockchain* um artefacto sem sentido, o que, de certo modo, leva ao esquecimento funcional, sem modificar a *blockchain*. Uma outra variante desta estratégia na *blockchain* seria a adopção de um consórcio para sistemas de saúde eletrónica (Zhang e Lin, 2018), e a arquitetura híbrida da MedChain opera com um princípio semelhante (Shen et al., 2019). No entanto, nenhum dos estudos avaliou formalmente se essa separação atende ao padrão legal de esquecimento previsto no Artigo 17 do RGPD. A segunda estratégia passa pelo uso de uma técnica denominada *crypto-shredding*, que criptografa os dados dos pacientes antes do armazenamento na cadeia ou próximo à cadeia e apaga, destruindo irremediavelmente, as chaves de cifragem (Xi et al., 2022). Uma vez que a chave é eliminada, os dados cifrados tornam-se computacionalmente indistinguíveis do ruído. Esta técnica introduz complicações quanto à gestão de chaves multipartidárias, à durabilidade dos esquemas de encriptação atuais diante de futuros avanços criptográficos e à ausência de precedentes regulamentares que confirmem que a destruição de chaves constitui uma ação legal.

A *framework* S3EF-HBCA abordou esta categoria de desafios no nível da engenharia de requisitos, incorporando objetivos de privacidade, como minimização de dados, gestão de consentimento, transparência, portabilidade e retenção. O artigo 25.º do RGPD exige precisamente este tipo de *design* antecipatório, exigindo que a proteção de dados seja incorporada no momento da definição dos meios de processamento, em vez de ser adaptada após a implementação (European Parliament and Council of the European Union, 2016). Para a presente tese, esta exigência regulamentar traduz-se numa obrigação concreta: a plataforma deve incorporar uma estratégia deliberada de colocação de dados que distinga os metadados na cadeia dos conteúdos clínicos fora da cadeia, com vias de eliminação claramente definidas e juridicamente defensáveis para cada categoria.

1.3 OBJETIVOS

A presente secção define o propósito desta investigação, num objetivo geral e num conjunto de objetivos específicos alinhados com as lacunas identificadas na secção anterior (Hevner et al., 2004; Peffers et al., 2007).

1.3.1 *Objetivo Geral*

O objetivo geral desta tese é conceber, desenvolver e avaliar uma plataforma baseada em *blockchain*, designada MedBlock, para a partilha segura de dados de saúde centrada no paciente no contexto dos cuidados de saúde portugueses, demonstrando conformidade com o RGPD e consideração dos requisitos da HIPAA. A plataforma aborda três deficiências identificadas no panorama atual da informação de saúde em Portugal: segurança inadequada na troca de dados entre instituições, ausência de controlo do consentimento ao nível do paciente e interoperabilidade limitada no ecossistema heterogéneo de prestadores do SNS (European Parliament and Council of the European Union, 2016; Theodouli et al., 2018; Xi et al., 2022).

1.3.2 *Objetivos específicos*

Para concretizar o objetivo geral, foram definidos cinco objetivos específicos. Cada um corresponde a uma fase distinta do processo de investigação, desde a análise do contexto até à formulação de recomendações.

- Analisar o panorama da partilha de dados de saúde em Portugal e elaborar uma arquitetura de requisitos que capte as necessidades funcionais (concessão e revogação de consentimento, recuperação de dados entre instituições, auditabilidade), as qualidades não funcionais (minimização de dados, latência, disponibilidade) e as restrições regulamentares impostas pelo RGPD — em particular os artigos 5.º, 17.º, 25.º e 32.º.
- Conceber uma arquitetura de *blockchain* permissionada que separe os metadados de consentimento na *blockchain* dos dados clínicos fora da *blockchain*, incorporando *smart contracts* para automatizar a criação, revogação e verificação do consentimento do paciente. O projeto aborda o dilema entre a imutabilidade da *blockchain* e o direito de esquecimento do RGPD (artigo 17.º) por meio de uma abordagem híbrida, na *blockchain* e fora da *blockchain*, devido ao seu isolamento de dados, baseado em canais, e à gestão de identidade de nível empresarial.
- Implementar um protótipo funcional que integre o Hyperledger Fabric, servidores de dados clínicos FHIR e o sistema de identidade digital Autenticação.Gov de Portugal (Chave Móvel Digital (CMD)). A integração com a CMD é necessária porque, num sistema de partilha de dados de saúde, a identidade do

paciente que concede ou revoga consentimento deve ser verificada de forma inequívoca; a CMD, enquanto mecanismo de autenticação forte, garante essa verificação sem que a plataforma necessite de gerir credenciais de identidade próprias. O protótipo abrange uma rede *blockchain* multi-organização, um *chaincode* de gestão de consentimento, servidores HAPI FHIR com segurança *mutual Transport Layer Security* (mTLS) e uma aplicação *web* com gestão de sessões cifradas. (Agência para a Modernização Administrativa (AMA), 2026)

- Avaliar as propriedades de segurança e as características de desempenho do protótipo por meio de testes de segurança da informação, incluindo uma avaliação de segurança estruturada em torno do Quadro de Cibersegurança do *National Institute of Standards and Technology* (NIST) (National Institute of Standards and Technology, 2018) e do Guia de Testes da *Open Worldwide Application Security Project* (OWASP) (OWASP Foundation, 2020).
- Formular recomendações para a adoção e implementação da partilha de dados de saúde baseada em *blockchain* no SNS português, abordando as barreiras técnicas, as estratégias de alinhamento regulatório e as considerações organizacionais.

No seu conjunto, estes objetivos cobrem o ciclo completo da metodologia *Design Science Research* adotada nesta tese, desde a identificação do problema até à comunicação dos resultados.

1.4 ÂMBITO

Esta secção estabelece as fronteiras tecnológicas e funcionais do trabalho, distinguindo o que foi efetivamente desenvolvido e avaliado do que permanece fora do seu alcance.

1.4.1 *O que a tese abrange*

O trabalho desenvolvido é delimitado por quatro dimensões que definem as suas vertentes tecnológicas, regulamentares e funcionais.

- Estrutura de *blockchain* permissionada. O *design* e a implementação baseiam-se no Hyperledger Fabric 2.5, selecionado pelo seu isolamento de dados por canais, arquitetura de consenso modular e provedores de serviços de associação

de nível empresarial (Polge et al., 2021). O modelo autorizado é apropriado para a área da saúde, pois a participação na rede deve ser restrita a entidades autenticadas e autorizadas, uma propriedade que as *blockchains* públicas e não permissionadas não podem impor por *design* (Cyran, 2018).

- Tipos específicos de dados RES. O protótipo implementa três tipos de recursos: Paciente (dados pessoais), Consulta (consultas agendadas) e Relatório de Diagnóstico (resultados de exames). Estes representam o conjunto mínimo viável necessário para demonstrar a recuperação e agregação de dados clínicos entre instituições. A especificação FHIR R4 define mais de 150 tipos de recursos; ampliar a cobertura constitui um trabalho futuro, mas não altera os princípios de arquitetura demonstrados (HL7 International, 2019).
- Contexto regulatório e institucional português. A integração com o Autenticação.Gov para autenticação de pacientes via CMD e o uso do Número Serviço Nacional de Saúde (NSNS) como identificador do paciente são específicos de Portugal (Agência para a Modernização Administrativa (AMA), 2026). A análise regulatória centra-se no RGPD, com a HIPAA como referência comparativa (European Parliament and Council of the European Union, 2016; U.S. Department of Health and Human Services, 1996).
- Gestão de consentimento baseada em *smart contracts*. A camada de *blockchain* é deliberadamente restrita à gestão de consentimento, concessão, revogação e verificação de direitos de acesso organizacionais, em vez do armazenamento de dados clínicos. Esta separação *on-chain*/*off-chain* serve tanto à conformidade com o RGPD (preservando o direito ao apagamento) como à escalabilidade (evitando transações de *blockchain* com grande carga útil) (Cyran, 2018; Shen et al., 2019).

Estas quatro dimensões estabelecem o espaço de *design* dentro do qual todas as decisões de arquitetura, implementação e avaliação foram tomadas.

1.4.2 *O que a tese não abrange*

De forma complementar, são explicitadas três exclusões que clarificam os limites do protótipo.

- Implementação clínica completa. O MedBlock é um protótipo de investigação destinado a validar decisões de *design*, não um sistema pronto para produção. Questões relacionadas à produção, como escalonamento horizontal entre *peers*,

armazenamento persistente, configurações de *orderers* de alta disponibilidade e recuperação de desastres.

- Integração com sistemas SNS ativos. O protótipo não se conecta a sistemas reais de informação de saúde portugueses. Os servidores FHIR são preenchidos com dados clínicos sintéticos; nenhum dado real de pacientes é processado em qualquer fase.
- *Tokenização* de transações. Nenhum mecanismo de criptomoeda, *token* ou transação financeira é implementado. A *blockchain* é usada exclusivamente para gestão de metadados de consentimento.

1.5 CONTRIBUIÇÕES ESPERADAS

No lado prático, o principal resultado é o MedBlock — um protótipo funcional que integra uma *blockchain* permissionada do Hyperledger Fabric, servidores de dados clínicos FHIR, gestão de consentimento baseada em *smart contracts* e o sistema nacional de identidade digital de Portugal numa única plataforma. Esta combinação de uma *blockchain* permissionada, um formato de dados clínicos padronizado e um mecanismo nacional de identificação eletrónica não foi demonstrada na literatura até a data da elaboração desta tese. As implementações anteriores dependiam de *blockchains* públicas, com limitações de confidencialidade inerentes, empregavam formatos de dados personalizados que dificultavam a interoperabilidade ou omitiam a integração com a identidade nacional. A avaliação produz ainda dados empíricos de segurança e desempenho, escassos num campo dominado por propostas conceptuais (Xi et al., 2022).

Do ponto de vista teórico, a tese desenvolve e valida uma abordagem estruturada para conciliar as propriedades de arquitetura da *blockchain* (imutabilidade, transparência, armazenamento distribuído) com os requisitos do RGPD (direito ao esquecimento, minimização de dados, proteção de dados desde a conceção). O modelo de *smart contract*, restrito a três operações (criação, revogação, consulta) sobre uma estrutura de dados mínima, incorpora princípios de *design* transferíveis: minimização de dados no nível do *chaincode*, auditabilidade, separação do consentimento aos dados clínicos e governança multi-organizacional.

1.6 ORGANIZAÇÃO DO DOCUMENTO

Para além do presente capítulo introdutório, o restante documento organiza-se em mais seis capítulos.

O Capítulo 2 apresenta o estado da arte. Parte dos fundamentos da tecnologia *blockchain*, percorre os modelos de rede e mecanismos de consenso, caracteriza os *smart contracts* e detalha a arquitetura do Hyperledger Fabric, terminando com o enquadramento institucional português e a formulação da lacuna de investigação que este projeto procura colmatar.

O Capítulo 3 examina trabalhos relacionados, nomeadamente sistemas de RES baseados em *blockchain*, plataformas permissionadas em avaliação comparativa e implementações empíricas do Hyperledger Fabric no domínio da saúde. O capítulo termina com a análise comparativa que posiciona o MedBlock face às soluções existentes.

O Capítulo 4 descreve a metodologia adotada, baseada na *Design Science Research Methodology*, e expõe a identificação do problema, as decisões de *design* arquitetural, a estratégia de conformidade regulamentar com o RGPD, a estratégia de avaliação, o ambiente de desenvolvimento e o cronograma do projeto.

O Capítulo 5 documenta a implementação do protótipo. Cobre a infraestrutura de rede e a contentorização, a hierarquia de *Public Key Infrastructure* (PKI), a configuração do canal e a geração do *genesis block*, o *chaincode* de gestão de consentimento, a camada de dados clínicos *off-chain* com servidores HAPI FHIR (implementação de referência *open-source* do HL7 FHIR) (HAPI) FHIR e *reverse proxies* Nginx, a aplicação *web* no *backend* e no *frontend* e os controlos de segurança aplicados na camada aplicacional e perimetral.

O Capítulo 6 apresenta a avaliação. Inclui o *benchmarking* de desempenho com Hyperledger Caliper sob cinco níveis progressivos de carga e a avaliação de segurança suportada por análise estática de código com SonarQube, análise dinâmica com OWASP ZAP e *scanning* de vulnerabilidades de infraestrutura com o Tenable Nessus Professional.

Por fim, o Capítulo 7 sintetiza as conclusões do projeto, identifica as limitações do trabalho e formula perspetivas futuras de investigação e desenvolvimento.

[9 de junho de 2026]

2

ESTADO DA ARTE

O Capítulo 1 estabeleceu o problema que motiva esta tese: a fragmentação dos dados clínicos no ecossistema português, um panorama de ameaças à cibersegurança em agravamento e um quadro regulamentar construído sobre o RGPD e sobre o futuro *European Health Data Space* (EHDS).

O presente capítulo parte dos fundamentos da tecnologia *blockchain*, progride pelos mecanismos de consenso que governam o seu funcionamento em redes permissionadas, aprofunda a arquitetura do Hyperledger Fabric, *framework* adoptada neste trabalho, e termina com a caracterização do contexto institucional português e com a formulação da lacuna de investigação que a tese visa colmatar.

2.1 TECNOLOGIA *BLOCKCHAIN*: FUNDAMENTOS

Esta secção relata os conceitos da tecnologia *blockchain* relevantes. As subsecções que se seguem descrevem a génese histórica da *blockchain*, a estrutura do *ledger* distribuído, as propriedades arquiteturais que sustentam a sua adoção em domínios regulados e os critérios que devem orientar a decisão para implementar uma solução.

2.1.1 *Origem e evolução conceptual*

A génese da *blockchain* situa-se em 1991, num trabalho sobre marcação criptográfica temporal de documentos digitais, cujo objetivo era impedir a adulteração retroativa de registos digitalizados (Haber e Stornetta, 1991). O encadeamento de blocos por funções de *hash* surgiu então como mecanismo para ancorar documentos numa sequência verificável: cada novo documento era ligado ao anterior por um registo criptográfico, de modo que qualquer alteração *a posteriori* invalidaria toda a cadeia subsequente. Durante quase duas décadas, a ideia permaneceu essencialmente académica.

A operacionalização em larga escala ocorreu em 2008, com o artigo científico de Nakamoto, que descreveu o protocolo Bitcoin como um sistema de pagamen-

tos eletrónico P2P capaz de dispensar qualquer autoridade central de confiança (Nakamoto, 2008). O contributo original do Bitcoin não reside no uso de *hashes* criptográficos, mas na combinação com um mecanismo de consenso descentralizado, *Proof-of-Work* (PoW), analisado com mais detalhe na secção 2.2. A partir dessa implementação, a tecnologia poderia ser aplicada a domínios distintos do financeiro: cadeias de abastecimento, registos notariais, certificação académica e, mais recentemente, partilha de dados clínicos (Xi et al., 2022).

A CompTIA caracteriza a *blockchain* como uma estrutura matemática para armazenar transações ou dados digitais numa solução digital imutável, distribuída e descentralizada, composta por blocos ligados por meio de assinaturas criptográficas quase impossíveis de falsificar ou alterar (CompTIA Blockchain Advisory Council, 2023).

2.1.2 *Estrutura do ledger distribuído*

Um *ledger* distribuído é uma sequência ordenada de blocos replicada por um conjunto de nós participantes. Cada bloco contém um cabeçalho com o *hash* do bloco imediatamente anterior, um *timestamp* e a raiz de uma árvore de Merkle que condensa as transações incluídas. O corpo do bloco contém as próprias transações, cada uma delas uma operação que altera o estado global mantido pela rede (Nakamoto, 2008). O encadeamento por *hashes* garante que qualquer modificação posterior de um bloco implica o recálculo de todos os blocos subsequentes, o que torna a adulteração computacionalmente inviável num sistema com replicação e consenso robustos.

A arquitetura funciona apenas na condição dos nós manterem cópias convergentes do *ledger*. Essa convergência é assegurada pelo protocolo de consenso, tema aprofundado na Secção 2.2, que determina quando uma transação é considerada confirmada e incorporada de forma definitiva. Em regime estável, qualquer nó pode verificar independentemente a integridade do *ledger*: basta recalcular os *hashes* em cadeia e compará-los com os valores armazenados. Esta propriedade de auditabilidade universal é responsável por grande parte do interesse que a *blockchain* suscita em domínios regulados, onde a demonstrabilidade da integridade dos registos é uma exigência legal e não uma conveniência técnica.

2.1.3 *Propriedades de arquitetura*

Quatro propriedades da arquitetura *blockchain* alinham-se diretamente com os requisitos da partilha segura de dados de saúde. A primeira é a imutabilidade: uma vez confirmada uma transação, alterá-la exige esforço computacional proporcional à segurança do mecanismo de consenso, o que assegura que o histórico de acessos a registos clínicos permanece íntegro e auditável (Cyran, 2018; Xi et al., 2022). A segunda é a descentralização: distribuir o *ledger* por vários nós elimina o ponto único de falha característico das bases de dados centralizadas e distribui a confiança entre múltiplas entidades, aumentando a resiliência face a ataques e a falhas de infraestrutura (Azaria et al., 2016; Cyran, 2018). A terceira é a transparência: todos os participantes com acesso ao *ledger* observam o mesmo conjunto de transações, o que elimina discrepâncias entre registos paralelos e simplifica auditorias regulatórias. A quarta, particularmente relevante em contextos multi-institucionais como o português, é a consensualidade: a validação de uma transação depende de acordo entre várias organizações, e não da decisão unilateral de qualquer uma delas (Polge et al., 2021).

A imutabilidade entra em conflito direto com o artigo 17.º do RGPD, que consagra o direito ao esquecimento e obriga o responsável pelo tratamento a eliminar dados pessoais a pedido do titular. A resolução destes constrangimentos requer que os dados clínicos sensíveis sejam armazenados fora da cadeia, enquanto apenas metadados de acesso e *hashes* de integridade permanecem *on-chain*. A descentralização tem um custo de latência e complexidade operacional, o que torna essencial a escolha criteriosa do modelo de rede e do algoritmo de consenso.

2.1.4 *Aplicabilidade da blockchain*

A adoção de uma tecnologia como a *blockchain* numa solução deve assentar numa análise rigorosa dos requisitos do sistema. A literatura propõe diversos esquemas formais para esta avaliação (Koens e Poll, 2018; Yaga et al., 2018), sendo o mais citado o proposto por Wüst e Gervais em 2018: uma árvore de decisão, representada na Figura 2.1, com seis questões que ajudam a decidir, de forma progressiva, entre base de dados convencional, *blockchain* permissionada pública e *blockchain* permissionada privada (Wüst e Gervais, 2018). As questões abrangem a necessidade de manter estado, a existência de múltiplos escritores, a disponibilidade de terceiros

de confiança, o grau de conhecimento mútuo entre participantes e a exigência de verificabilidade pública.

Este esquema foi considerado na conceção do MedBlock. O percurso pelas seis questões, e a fundamentação das respostas no contexto do ecossistema de saúde português, são apresentados no Capítulo 4, onde serve de base à decisão de implementar uma *blockchain* permissionada.

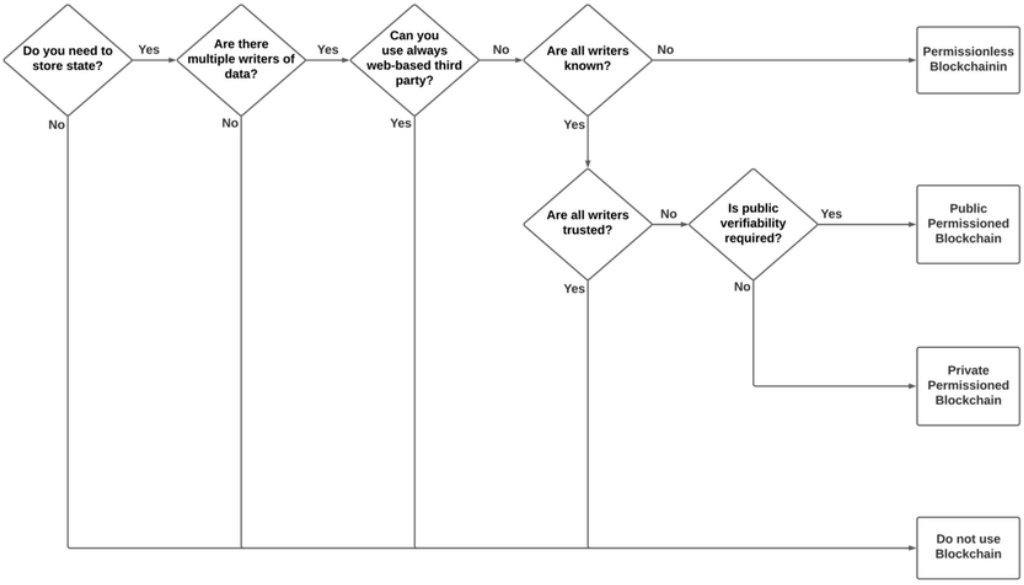


Figura 2.1: Árvore de decisão para determinar a aplicabilidade de *blockchain*.

2.2 MODELOS DE REDE E MECANISMOS DE CONSENSO

As redes de *blockchain* distinguem-se quanto ao grau de abertura, à participação e ao controlo exercido sobre a autenticação das partes interessadas. As redes sem permissão permitem que qualquer nó se junte à rede e participe no processo de validação; Bitcoin e Ethereum, na sua configuração pública, são os exemplos mais conhecidos. As redes permissionadas restringem a participação a entidades autenticadas e autorizadas, modelo adequado a cenários empresariais ou setoriais em que a identidade dos participantes é conhecida e regulamentada. Uma variante importante das redes permissionadas é o modelo de consórcio, em que várias organizações partilham a governação e operam nós conjuntamente (Polge et al., 2021). O sistema de saúde português, com múltiplos hospitais, centros de saúde e instituições privadas sob diferentes regimes jurídicos, enquadra-se naturalmente nesse modelo.

Em redes sem permissão, o consenso assenta tipicamente em PoW ou *Proof-of-Stake* (PoS). O PoW, adotado pelo Bitcoin, exige que os nós resolvam problemas computacionalmente complexos para poder produzir blocos, o que protege a rede à custa de consumo computacional e de latência elevada (Nakamoto, 2008). O PoS, adotado pelo Ethereum 2.0, substitui o esforço computacional pela posse de ativos na rede, reduzindo o consumo energético. Nenhum dos dois é adequado a aplicações clínicas: o custo transacional e a latência são incompatíveis com os requisitos de desempenho, e a participação aberta torna inaplicáveis as exigências de controlo de acesso impostas pelo RGPD.

Em redes permissionadas, o número de algoritmos disponíveis é maior. Especificamente no ecossistema do Hyperledger Fabric existem três opções. A configuração Solo, com um único nó de *ordering*, serve apenas para desenvolvimento e testes, dado que a falha desse nó afeta toda a rede (Androulaki et al., 2018). O Raft, adotado a partir do Fabric 1.4.1, implementa tolerância a faltas por paragem (*crash fault tolerance*) através de um modelo *leader-follower* em que um nó líder ordena as transações e replica-as por um conjunto de seguidores; sobrevive à falha de até metade menos um dos nós e tolera atrasos de rede (Ongaro e Ousterhout, 2014). Mais recentemente, o Fabric 3.0 introduziu suporte nativo a *Byzantine Fault Tolerance State Machine Replication* (BFT-SMaRt), um protocolo de replicação de *state machines* (Bessani et al., 2014).

2.3 SMART CONTRACTS

O conceito de *smart contract* antecede a *blockchain*. Foi formulado por Nick Szabo em 1994, e publicado em 1997, como um protocolo de transação computadorizado que executa os termos de um contrato, com o objetivo explícito de traduzir obrigações contratuais em código (Szabo, 1997).

Num sistema *blockchain*, um *smart contract* é um programa armazenado no *ledger* e executado de forma igual por todos os nós da rede. O determinismo é requisito indispensável: dada a mesma entrada e o mesmo estado inicial, cada nó tem de obter o mesmo resultado, evitando divergência entre cópias do *ledger* (Androulaki et al., 2018). Esta exigência condiciona as linguagens de implementação: no Ethereum, os *smart contracts* são escritos em Solidity e executados numa máquina virtual dedicada, a *Ethereum Virtual Machine* (EVM); no Hyperledger Fabric, o equivalente, designado *chaincode*, é implementado em Go, Java ou Node.js e executado em contentores isolados (Polge et al., 2021).

Nos domínios da saúde, a vantagem arquitetural é clara. Em vez de depender de lógica aplicacional centralizada, que exige confiança no operador do servidor, as regras de acesso são codificadas no *ledger* e aplicadas uniformemente por todos os nós. As implicações de segurança, porém, não são triviais. A imutabilidade dos *smart contracts* significa que erros descobertos após a instalação e vulnerabilidades exigem auditoria rigorosa antes da publicação (Atzei et al., 2017).

2.4 HYPERLEDGER FABRIC: ARQUITETURA E COMPONENTES

Esta secção descreve a arquitetura do Hyperledger Fabric, a *framework* de *blockchain* permissionada selecionada para o MedBlock.

2.4.1 *Visão geral da arquitetura modular*

O Hyperledger Fabric, inicialmente desenvolvido pela *International Business Machines* (IBM) e mantido pela Linux Foundation desde 2015, é uma *framework* de *blockchain* permissionada desenhada especificamente para cenários empresariais (Androulaki et al., 2018). Distingue-se de outras plataformas nomeadamente por três opções arquiteturais.

A primeira é a ausência de criptomoeda nativa. As transações no Fabric não envolvem taxas de execução, o que elimina um vetor de complexidade regulatória e operacional particularmente relevante em contextos clínicos, nos quais qualquer mecanismo de pagamento embebido no *software* exigiria autorização separada por parte das autoridades financeiras (Polge et al., 2021). A segunda é o modelo de execução *execute-order-validate*: uma transação é primeiro executada e assinada (*endorsed*) por um subconjunto de nós, depois ordenada pelo serviço de *ordering*, e só então validada e incorporada no *ledger* por cada nó (Androulaki et al., 2018). Esta separação permite paralelismo na execução e simplifica a integração com sistemas empresariais existentes. A terceira é a modularidade: cada componente do Fabric, o mecanismo de consenso, o *Membership Service Provider* (MSP), a linguagem de *chaincode*, pode ser substituído de forma independente, o que adapta a *framework* a domínios com requisitos heterogéneos.

A rede Fabric é composta por um conjunto de organizações, cada uma operando os seus próprios nós, que comunicam entre si através de canais. Cada organização pode dispor de uma autoridade de certificação dedicada e de um MSP que traduz

certificados X.509 em identidades na rede. Os nós desempenham papéis distintos: os nós designados por *peers* executam *chaincode* e mantêm uma cópia do *ledger*; nós de *ordering* que agrupam transações em blocos e o conjunto é coordenado através de políticas de *endorsement* que explicitam, para cada operação, quais as organizações cuja assinatura é necessária. As subsecções seguintes detalham cada um destes componentes.

2.4.2 *Canais*

Um canal, no Fabric, é uma partição lógica do *ledger* visível apenas a um subconjunto de organizações (Androulaki et al., 2018). Corresponde, na prática, a uma sub-rede privada dentro da rede global: cada canal mantém o seu próprio *ledger*, a sua própria configuração e as suas próprias políticas de *endorsement*, e o *chaincode* instalado num canal não está acessível aos outros. Esta arquitetura permite que uma mesma rede Fabric aloje múltiplas interações confidenciais em paralelo, cada uma isolada das restantes.

Em contextos de saúde, a utilidade dos canais é evidente. Um hospital público e um hospital privado podem operar num canal partilhado para a partilha de referências clínicas, enquanto um canal separado, mais restrito, pode ser usado para dados de investigação anonimizados. Do ponto de vista regulamentar, o isolamento por canais suporta diretamente o princípio da minimização de dados consagrado no artigo 5.º do RGPD, uma vez que cada transação é replicada apenas pelo subconjunto estritamente necessário de nós (European Parliament and Council of the European Union, 2016; Gangula et al., 2021). No protótipo MedBlock, esta propriedade é explorada pela adoção de um canal de *staging*, usado para *benchmarking* e testes de integração, separado do canal de produção.

2.4.3 *Peers*

Os *peers* são os nós que mantêm cópias do *ledger* e executam o *chaincode*. O Fabric distingue três papéis funcionais que um *peer* pode assumir. Um *endorsing peer* é aquele que executa transações e assina criptograficamente os resultados antes de estas serem submetidas ao serviço de *ordering*; são os *endorsements* que determinam se uma transação satisfaz a política de *endorsement* da organização. Um *committing peer* recebe blocos ordenados do serviço de *ordering*, valida cada transação contra a política de *endorsement* e atualiza o *ledger*. Um *anchor peer* é designado pela confi-

guração do canal como ponto de entrada para *gossip* entre organizações, facilitando a disseminação eficiente de blocos numa rede multi-organizacional (Androulaki et al., 2018).

O ciclo de vida de uma transação reflete esta divisão de tarefas. A aplicação cliente envia a proposta de transação aos *endorsing peers* da sua organização, estes simulam a execução do *chaincode* e devolvem respostas assinadas. O cliente agrega as respostas e submete a transação, acompanhada dos *endorsements*, ao serviço de *ordering*. Este ordena a transação dentro de um bloco e distribui o bloco pelos *committing peers*, que aplicam as mudanças ao seu *ledger* local (Androulaki et al., 2018; Polge et al., 2021).

2.4.4 *Ordering service*

O serviço de *ordering* é o componente que garante a consistência global do *ledger*. A sua função consiste em receber transações de vários clientes, ordená-las numa sequência canónica e agrupá-las em blocos, que são depois difundidos para todos os *peers* participantes do canal. Ao contrário do que sucede em *blockchains* baseadas em PoW, em que a *ordering* e a validação são entrelaçadas no mesmo processo, no Fabric estas duas operações estão explicitamente separadas, o que permite otimizar cada uma de forma independente (Androulaki et al., 2018).

As topologias suportadas correspondem às opções de consenso discutidas na Secção 2.2. A configuração Solo, com um único nó de *ordering*, é adequada apenas a ambientes de desenvolvimento. O Raft é a opção recomendada para produção desde o Fabric 2.0: um *cluster* de nós de *ordering* elege um líder que produz blocos, replicando-os pelos seguidores; o sistema tolera a falha simultânea de uma minoria dos nós e recupera automaticamente por nova eleição (Ongaro e Ousterhout, 2014). O BFT-SMaRt, introduzido no Fabric 3.0, adiciona tolerância a comportamentos bizantinos, à custa de maior tráfego de *ordering* e de requisitos de infraestrutura superiores (Bessani et al., 2014). A escolha do *orderer* influencia diretamente a resiliência global da rede: uma falha do serviço de *ordering* paralisa a produção de novos blocos, ainda que os *peers* continuem a responder a consultas sobre o *ledger* existente.

2.4.5 *Chaincode*

O *chaincode* é a implementação de *smart contract* no Hyperledger Fabric. Cada *chaincode* é instalado em *peers* designados e invocado através de transações propostas pelos clientes da rede. O ciclo de vida segue três fases explícitas: instalação no *peer*, aprovação pela organização (mediante assinatura pelos administradores do MSP) e *commit* no canal, este último só possível quando um número suficiente de organizações, definido pela política de *lifecycle* do canal, tiver aprovado a mesma versão (Androulaki et al., 2018). Este mecanismo de governação partilhada impede que uma única organização altere unilateralmente a lógica de negócio da rede.

A política de *endorsement* associada ao *chaincode* explicita quais as organizações cujas assinaturas são necessárias para validar uma transação: pode exigir-se, por exemplo, que qualquer operação de escrita seja aprovada por pelo menos um *peer* de cada organização participante, ou que operações críticas requeiram *endorsement* por maioria qualificada. Esta expressividade permite modelar cenários de governação sofisticados, incluindo padrões de separação de funções e de verificação cruzada entre instituições, propriedade particularmente valiosa em ambientes clínicos multi-organizacionais (Gangula et al., 2021).

2.4.6 *Membership Service Provider*

O MSP é um componente que traduz a infraestrutura criptográfica de chaves públicas em identidades operacionais de rede. Cada organização Fabric opera o seu próprio MSP, que define os critérios pelos quais um certificado X.509 é reconhecido como pertencente a um elemento legítimo e determina o papel do mesmo, designadamente, *admin*, *client*, *peer* ou *orderer*. Qualquer transação submetida à rede é assinada com a chave privada associada a um certificado emitido pela autoridade de certificação da organização, e é a validação dessa assinatura contra as regras do MSP que determina se a transação é autorizada (Androulaki et al., 2018; Polge et al., 2021).

A hierarquia de certificação adotada pelo Fabric é tipicamente de dois níveis. Uma CA raiz emite certificados para um conjunto de CAs intermédias, são estas que emitem certificados para os participantes da rede. Esta estrutura permite revogar certificados comprometidos sem impacto no conjunto de certificados emitidos por outras CAs intermédias, e alinha-se com as boas práticas de gestão de PKI publicadas em normas internacionais (Cooper et al., 2008).

2.4.7 *Autenticação X.509: paralelo entre MSP e Autenticação.Gov*

O modelo de identidade do Fabric, conforme descrito, é uma instanciação do modelo PKI X.509, normalizado pela *International Telecommunication Union, Telecommunication Standardization Sector* (ITU-T) e formalizado pela *Internet Engineering Task Force* (IETF) através do RFC 5280 (Cooper et al., 2008). Um certificado X.509 liga uma chave pública a um conjunto de atributos do seu titular; nome, organização, função, mediante assinatura digital de uma autoridade de certificação. A validação do certificado envolve o percurso da cadeia de confiança desde o certificado do titular até uma CA raiz considerada fidedigna, verificando, em cada passo, a validade da assinatura e o estado de revogação. Este modelo é a base de praticamente toda a autenticação forte utilizada em redes modernas, desde o protocolo *Transport Layer Security* (TLS) à assinatura de código e à autenticação em sistemas corporativos.

O Estado português adotou o mesmo modelo para a identidade digital dos cidadãos. O Cartão de Cidadão e, sobretudo, a CMD assentam numa hierarquia de certificados X.509 emitidos sob o controlo da Agência para a Modernização Administrativa (AMA), em conformidade com o Regulamento *electronic IDentification, Authentication and trust Services* (eIDAS) da União Europeia (Agência para a Modernização Administrativa (AMA), 2026; European Parliament and Council of the European Union, 2014). A CMD fornece autenticação forte e assinatura digital qualificada, com validade jurídica equivalente à assinatura manuscrita. A integração aplicacional é feita através do serviço Autenticação.Gov, que expõe os mecanismos de autenticação sobre OAuth 2.0 e OpenID Connect.

Ambos os sistemas assentam numa hierarquia de CAs raiz e intermédias, ambos emitem certificados X.509 para identidades de entidade final, ambos validam assinaturas contra cadeias de confiança e, decisivamente para o MedBlock, ambos coexistem tecnicamente sem fricção arquitetural. A identidade institucional dos nós da rede *blockchain*, fornecida pelos MSPs das organizações hospitalares, e a identidade dos pacientes, fornecida pela CMD, partilham o mesmo modelo criptográfico subjacente. Esta coerência permite uma integração nativa: a mesma plataforma usa X.509 para autenticar uma transação de um *peer* e X.509 para atestar a identidade do cidadão que autoriza essa transação.

2.5 CONTEXTO PORTUGUÊS

A primeira é a assimetria de governação entre prestadores públicos e privados. Os SPMS, criados em 2010, centralizam a infraestrutura digital de saúde do SNS e gerem, entre outros, o portal Área do Cidadão, com mais de 2,25 milhões de utilizadores registados, a prescrição eletrónica e a participação nacional no sistema MyHealth@EU (Martins, 2020; SPMS — Serviços Partilhados do Ministério da Saúde, 2026a). Os prestadores privados, que cobrem aproximadamente 20% da população portuguesa, operam sob regimes jurídicos independentes e não estão obrigados a partilhar os seus dados clínicos com o SNS (TEHDAS — Towards the European Health Data Space, 2023). Qualquer plataforma *blockchain* implementada em contexto nacional tem, assim, de acomodar dois regimes distintos de participação: institucional obrigatória, no setor público; voluntária e incentivada, no setor privado.

A segunda é a maturidade parcial da infraestrutura digital existente. A adoção do HL7 v2.5, a migração em curso para o FHIR, a Área do Cidadão e o sistema de receita eletrónica constituem pontos de contacto estabelecidos que qualquer nova plataforma tem de respeitar. Não se trata, neste caso, de implementar uma tecnologia num vazio digital, mas de acrescentar uma camada complementar de confiança, auditabilidade e gestão de consentimento aos fluxos de dados já operacionais (Martins, 2020; TEHDAS — Towards the European Health Data Space, 2023).

A terceira é a trajetória de adesão ao EHDS. O regulamento europeu, publicado em março de 2025 e com entrada em vigor prevista para 2027, impõe a criação de organismos nacionais de acesso a dados de saúde e exige suporte, tanto para utilização primária, como para utilização secundária, formulação de políticas, inovação (SPMS — Serviços Partilhados do Ministério da Saúde, 2023). Uma plataforma *blockchain* implementada no SNS tem de suportar os dois regimes de utilização, permitindo acesso controlado e anonimizado a dados secundários sem comprometer nem a privacidade do paciente nem as garantias de integridade de que depende a partilha clínica primária.

2.6 MOTIVAÇÃO PARA A INVESTIGAÇÃO

A convergência dos elementos apresentados nas secções anteriores, a maturidade da tecnologia *blockchain*, a disponibilidade de *frameworks* permissionadas robustas como o Hyperledger Fabric, a existência de uma infraestrutura nacional de identidade

digital baseada em X.509 e o contexto regulatório e institucional português desenha um espaço de investigação bem definido, no qual persistem lacunas concretas.

A presente tese propõe o MedBlock, uma plataforma baseada em *blockchain* para a partilha segura de dados de saúde, concebida especificamente para o contexto do sistema de saúde português. A arquitetura integra *smart contracts* para a gestão automatizada de consentimentos e controlo de acesso, técnicas criptográficas para a proteção de dados, e a infraestrutura nacional de identidade digital para a autenticação de pacientes. Ao cruzar conceção técnica, conformidade regulamentar e requisitos específicos do domínio, o trabalho visa contribuir simultaneamente com um protótipo funcional e com um conjunto de princípios de conceção transferíveis para outras jurisdições europeias com contextos institucionais semelhantes.

3

TRABALHOS RELACIONADOS

O Capítulo 1 estabeleceu a motivação da tese, enfatizando a natureza fragmentada dos dados de saúde em Portugal e o panorama de ameaças cada vez mais complexo, tal como descrito pela IBM em 2025 (IBM Security and Ponemon Institute, 2025). Além disso, estabelece os controlos regulamentares relevantes, especificamente os impostos do RGPD (Regulamento Geral sobre a Proteção de Dados da União Europeia) e do HIPAA (*Health Insurance Portability and Accountability Act* dos Estados Unidos). A Secção 2.1 apresenta a *blockchain*, um tipo de base de dados descentralizada que utiliza técnicas criptográficas, como solução potencial em termos de arquitetura.

Com base nas motivações e desafios do Capítulo 1, este capítulo aprofunda a análise da partilha de dados de saúde baseada em *blockchain*. Avaliam-se sistemas, estruturas e metodologias, analisando suas decisões arquitetónicas, a evidência empírica e as limitações.

A revisão está organizada em quatro secções principais, cada uma abordando um objetivo de investigação distinto e culminando num resumo geral. A secção 3.1 examina exaustivamente os sistemas de RES baseados em *blockchain*, centrando-se nas arquiteturas, nos paradigmas de *smart contracts*, nas estratégias de armazenamento de dados e nos resultados de avaliação publicados. A secção 3.2 apresenta um estudo comparativo de estruturas de *blockchain* permissionadas, com foco nas decisões arquitetónicas relevantes para a implementação na área da saúde. A secção 3.3 discute os resultados empíricos do Hyperledger Fabric, apoiando as abordagens em consideração.

3.1 SISTEMAS DE REGISTOS DE SAÚDE ELETRÓNICOS BASEADOS EM *BLOCKCHAIN*

Esta secção analisa os principais sistemas de *blockchain* para a gestão e a partilha de RESs, definindo os objetivos e o âmbito da análise. Cada subsecção avalia um

sistema quanto à arquitetura, ao modelo de *smart contract*, ao armazenamento de dados, ao protocolo de avaliação e às respectivas limitações.

3.1.1 *Blockchain como base para a partilha de dados na área da saúde*

Cyran defende a aplicação da *blockchain* na troca de dados na saúde. O estudo identificou falhas principais nos sistemas convencionais: fragmentação de dados, interoperabilidade insuficiente e salvaguardas de privacidade inadequadas ao acesso autorizado. A arquitetura proposta permite que *smart contracts* regulem o controlo de acesso, garantindo que apenas pessoal credenciado aceda a categorias clínicas específicas. A integridade dos dados é garantida pelo *ledger* descentralizado. A estrutura apoia a interoperabilidade por meio de uma camada unificada de troca de informações (Cyran, 2018).

A principal contribuição deste trabalho é justificar conceitualmente o uso da *blockchain* na saúde. A proposta é teórica, sem protótipo, avaliações de desempenho ou análise formal de segurança. Não apresenta uma estratégia concreta de armazenamento, o que suscita dúvidas sobre onde se encontram os registos. Questões importantes, como as limitações de armazenamento, não são abordadas. A conformidade regulatória é apenas mencionada, o que torna o estudo uma referência introdutória.

3.1.2 *MedRec: Gestão de registos centralizada do paciente em Ethereum*

MedRec, criado no *Massachusetts Institute of Technology* (MIT), destacando o controlo do paciente sobre os seus dados. Usa a *blockchain* pública Ethereum e três *smart contracts*: o de Registo gere identidades; o de Relação Paciente-Prestador (*Patient-Provider Relationship* (PPR)) controla as permissões de acesso e a localização dos registos; o de Resumo reune todas as referências do PPR do paciente. Assim, um novo prestador pode consultar o histórico do paciente sem contactar cada instituição (Azaria et al., 2016).

O MedRec separa os metadados na *blockchain* dos dados clínicos. A rede Ethereum armazena apenas a lógica dos *smart contracts*, as permissões e as inovações criptográficas, os registos médicos permanecem nas bases de dados locais dos prestadores. O acesso é controlado por um nó *Database Gatekeeper*, que aplica as permissões da *blockchain* antes de disponibilizar dados a utilizadores autorizados (Azaria et al.,

2016). Isso mantém a compatibilidade com sistemas existentes, evita custos de armazenamento em cadeia e suporta padrões de interoperabilidade, como HL7 e FHIR.

O MedRec cria uma nova versão de mineração. Em vez de recompensar com criptomoedas, o sistema oferece acesso a dados clínicos agregados e anonimizados. Essa abordagem aproxima os incentivos à manutenção da rede dos interesses da pesquisa médica (Azaria et al., 2016). Também aborda preocupações como o *free-rider*, que podem afetar a confiabilidade de sistemas baseados apenas na participação voluntária.

Apesar das suas contribuições inovadoras, o MedRec enfrenta limitações significativas. A utilização da rede pública Ethereum introduz restrições inerentes ao débito, à latência nas transações e às *gas fees* associadas, fatores que não se adaptam de forma eficiente aos sistemas nacionais de saúde. A estrutura carece de um mecanismo em conformidade com o RGPD para revogar o consentimento. A avaliação limitou-se a um protótipo de pequena escala utilizando dados sintéticos. O modelo de incentivo à mineração, embora conceptualmente inovador, deixa questões por resolver no que diz respeito à governação, tais como a resolução de litígios relativos à qualidade e ao acesso a esses dados.

3.1.3 *MedChain: Arquitetura híbrida de blockchain P2P*

MedChain apresenta uma arquitetura híbrida que resolve as limitações de soluções anteriores em saúde através da tecnologia de *blockchain*, com foco em fluxos de dados provenientes de dispositivos *Internet of Things* (IoT). O *design* separa artefactos imutáveis na *blockchain* dos dados mutáveis numa rede P2P de tabelas hash distribuídas (Shen et al., 2019). Isso reduz as transações na *blockchain* e aumenta a eficiência.

A MedChain usa partilha de dados em sessões para controlo granular de acesso. Ao contrário dos modelos estáticos, as sessões podem ser criadas, alteradas ou revogadas pelo paciente, permitindo controlo detalhado do acesso, das partes solicitantes e dos períodos (Shen et al., 2019).

Ao avaliar a eficiência, os autores demonstram menor sobrecarga em relação a sistemas baseados apenas no armazenamento na cadeia, devido à camada P2P. Mas os testes usaram apenas dados sintéticos e não avaliaram o desempenho real nem a interoperabilidade.

3.1.4 *Partilha de dados de saúde baseada em blockchain com consideração pelas normas regulamentares*

Theodouli et al. criaram um sistema *blockchain* que atende aos requisitos regulatórios de proteção de dados, alinhando-se melhor às exigências europeias. Usam *smart contracts* para controlo de acesso detalhado e criptografia para proteger a confidencialidade. Dados clínicos ficam fora da *blockchain*, que armazena apenas *hashes* e metadados de controlo de acesso (Theodouli et al., 2018). Essa abordagem busca equilibrar os limites da *blockchain* com as exigências legais.

Este trabalho destaca-se pela ênfase na conformidade regulatória. Ao contrário das abordagens anteriores, esta solução propõe uma arquitetura que facilita a minimização de dados, a limitação de finalidade e o controlo do consentimento. O armazenamento fora da cadeia permite a eliminação de dados, conforme o Artigo 17.º do RGPD. Porém, falta a validação jurídica formal ou o protótipo, o que torna incerta a aplicabilidade prática.

3.1.5 *Blockchain de consórcio com encriptação baseada em atributos*

Zhang e Lin propuseram uma arquitetura de *blockchain* de consórcio para a partilha segura de dados e a preservação da privacidade em sistemas de saúde eletrónica. A principal inovação técnica consistiu na integração da *Attribute-Based Encryption* (ABE) com a pesquisa por palavras-chave em dados cifrados. Os utilizadores autorizados podem consultar e recuperar registos de saúde relevantes sem expor o conteúdo em texto simples à rede *blockchain* ou a quaisquer nós intermediários. O modelo de consórcio restringiu a participação na rede a instituições de saúde verificadas. Cada instituição opera um nó de validação. Isto combinou a governação multipartidária com as vantagens de desempenho de uma topologia de rede controlada (Zhang e Lin, 2018).

Do ponto de vista regulatório, este modelo procura conciliar a transparência da *blockchain* com a privacidade dos pacientes. A adoção da ABE permite que as políticas de acesso sejam codificadas no texto cifrado, permitindo a decifragem apenas por utilizadores cujos atributos satisfazem as políticas relevantes. Esta abordagem corresponde naturalmente aos modelos de *Role-Based Access Control* (RBAC) estipulados pelo RGPD e pela HIPAA. Restringir a participação na rede a atores institucionais conhecidos apoia a conformidade com as disposições de salvaguardas físicas da HIPAA e com as restrições à transferência de dados do

RGPD (Zhang e Lin, 2018). No entanto, a escalabilidade das operações de ABE para políticas complexas ainda não foi avaliada empiricamente, nem foram abordadas as preocupações relativas à latência em contextos clínicos de elevado volume. Além disso, a interoperabilidade com as normas de dados de saúde existentes e a compatibilidade com a infraestrutura digital nacional continuam por ser exploradas, limitando a aplicabilidade direta destas conclusões a contextos como o SNS português.

3.2 PLATAFORMAS DE *BLOCKCHAIN* PERMISSIONADAS: AVALIAÇÃO COMPARATIVA

A seleção de uma plataforma de *blockchain* constitui uma das decisões arquitetónicas mais importantes em qualquer implementação no setor da saúde, uma vez que as propriedades da estrutura, relativas ao consenso, à privacidade, à execução de *smart contracts* e à gestão de identidades, refletem-se em todos os aspetos do comportamento do sistema. Polge et al. realizaram uma comparação sistemática de cinco das principais estruturas de *blockchain* permissionadas: Hyperledger Fabric, Ethereum, Quorum, MultiChain e R3 Corda. A sua análise avaliou cada estrutura em termos de conceção do protocolo de consenso, capacidades de *smart contracts*, mecanismos de preservação da privacidade, suporte a linguagens de programação, débito, latência e maturidade da adoção industrial. A principal conclusão foi que nenhuma estrutura domina todas as dimensões; a seleção da estrutura requer, invariavelmente, uma análise de compromissos específica ao contexto (Polge et al., 2021).

O Hyperledger Fabric, funciona como uma *blockchain* modular e permissionada com um protocolo de consenso. O Ethereum, na sua configuração permissionada, utiliza o consenso *Proof-of-Authority*. Os *smart contracts* são escritos em Solidity e executados na *Ethereum Virtual Machine* (EVM). No entanto, mesmo no modo permissionado, o Ethereum herda restrições arquitetónicas da sua origem como cadeia pública, incluindo um modelo de estado global que limita a privacidade ao nível da transação (Polge et al., 2021).

O Quorum, desenvolvido pela J.P. Morgan, ampliou o Ethereum com contratos privados visíveis apenas a participantes específicos e protocolos de consenso otimizados para consórcios, incluindo Raft e Istanbul *Byzantine Fault Tolerance* (BFT) (ConsenSys, 2023).

O R3 Corda adota uma filosofia fundamentalmente diferente, utilizando um serviço de notariado para validar a exclusividade das transações em vez da transmissão

global de blocos, o que maximiza a privacidade das transações, mas restringe a expressividade do seu modelo de contrato (R3, 2024).

O MultiChain oferece uma elevada configurabilidade das permissões de rede e suporta várias linguagens de programação, mas a sua comunidade e a adoção no domínio dos cuidados de saúde continuam a ser significativamente menores do que as do Fabric (Polge et al., 2021).

3.3 O HYPERLEDGER FABRIC NA ÁREA DA SAÚDE

Foi analisada a integração do Hyperledger Fabric com os *Clinical Decision Support Systems* (CDSSs) para responder ao problema generalizado de fadiga de alertas entre os profissionais de saúde. As implementações existentes de CDSS, normalmente integradas em sistemas de RES de uma única instituição, geram alertas com base em históricos de pacientes incompletos, uma vez que não têm acesso aos registos mantidos por outros prestadores de cuidados de saúde. A falta de informação resultante gera alertas irrelevantes ou redundantes a um ritmo que sobrecarrega os fluxos de trabalho clínicos, levando os médicos a ignorar a maioria das notificações, incluindo as clinicamente significativas (Gangula et al., 2021).

O modelo proposto aproveitou a arquitetura de canais do Hyperledger Fabric para permitir a troca de dados entre instituições. Quando um prestador de cuidados de saúde primários encaminhava um paciente a um especialista, ambas as organizações estabeleciam um canal privado no Hyperledger Fabric, através do qual o CDSS do especialista podia aceder ao historial completo de medicação do paciente e aos registos de resultados de diagnósticos anteriores. Ao enriquecer o contexto de dados disponível para o CDSS, esperava-se que o sistema melhorado produzisse alertas mais precisos e clinicamente relevantes, reduzindo tanto a carga cognitiva sobre os médicos, quanto a incidência de procedimentos de diagnóstico duplicados (Gangula et al., 2021). Este trabalho é significativo por ter ido além das propostas genéricas de *blockchain* para RES e ter identificado um problema clínico concreto — a fadiga de alertas — para o qual articulou uma solução assente em capacidades específicas do Hyperledger Fabric. A contribuição, no entanto, permaneceu teórica: não foi implementada nenhuma solução, e nenhuma medição empírica da redução da fadiga de alertas ou da melhoria da precisão do CDSS foi reportada.

3.3.1 *Hospital Provincial Frere: Implementação Empírica*

Oki et al. realizaram uma das poucas implementações empíricas de um sistema de RES baseado no Hyperledger Fabric, tendo implementado um protótipo no Hospital Provincial Frere, na África do Sul. O estudo empregou uma abordagem de métodos mistos, combinando casos de estudo, observação direta e entrevistas informais com o pessoal hospitalar. O sistema implementado definiu o controlo de acesso baseado em funções por meio de *chaincodes* do Hyperledger Fabric, estabelecendo níveis de permissão distintos para administradores, médicos e pacientes, e utilizou a rede para permitir a partilha segura de registos de pacientes entre os departamentos hospitalares (Oki et al., 2024).

Os resultados empíricos confirmaram vários benefícios hipotéticos da gestão de RES baseada em *blockchain*: o sistema impediu a modificação não autorizada dos registos dos pacientes por meio do fluxo de validação criptográfica do Fabric, manteve um histórico de transações auditável e permitiu a partilha de dados entre departamentos, o que antes exigia a transferência manual de registos. No entanto, o estudo também documentou limitações práticas significativas que as propostas teóricas tinham subestimado ou ignorado. Surgiram restrições de escalabilidade sob cargas elevadas de transações, o que levou à degradação do desempenho, tornando-o proibitivo à escala de um serviço nacional de saúde. A recuperação do histórico de transações tornou-se progressivamente mais lenta à medida que o *ledger* crescia, o que indica que a otimização das consultas a dados históricos no Hyperledger Fabric requer atenção arquitetónica para além das configurações padrão (Oki et al., 2024).

Talvez a conclusão mais relevante tenha sido a falta de adesão por parte dos utilizadores: a maioria dos pacientes inquiridos desconhecia a existência do sistema de RES, o que sugere que a implementação tecnológica, sem investimento simultâneo na educação dos pacientes e na formação dos profissionais de saúde, corre o risco de resultar num sistema tecnicamente funcional, mas operacionalmente subutilizado (Oki et al., 2024). O estudo identificou também uma lacuna em matéria de conformidade regulamentar, salientando a ausência de uma análise aprofundada sobre como os quadros jurídicos, tais como a HIPAA ou o RGPD, afetariam, na prática, as implementações de cuidados de saúde baseadas em *blockchain*. Estas conclusões são diretamente relevantes para o MedBlock, que deve abordar não só a conceção arquitetónica, mas também as dimensões de preparação para a implementação, relacionadas com a sensibilização dos utilizadores, a gestão da mudança institucional e a defensabilidade regulamentar, que o estudo do Hospital Frere revelou estar pouco investigada.

3.4 ANÁLISE COMPARATIVA E POSICIONAMENTO DO MEDBLOCK

A Tabela 3.1 sintetiza as principais características e limitações dos seis sistemas analisados neste capítulo, fornecendo uma base estruturada para identificar as lacunas de investigação que o MedBlock procura colmatar.

Tabela 3.1: Resumo comparativo dos sistemas revistos de partilha de dados de saúde baseados em *blockchain*.

Sistema	Tipo	Armaz.	Consent.	RGPD	Prot.	Aval.	Contexto
Cyran (2018)	Não espec.	Não espec.	Controlo acesso smart contract	Nenhuma	Não	Não	Nenhum
MedRec (Azaria et al., 2016)	Ethereum pública	Off-chain (BDs prestadores)	Contratos PPR, iniciado pelo paciente	Nenhuma (pré-RGPD)	Sim (limitado)	Apenas dados sintéticos	EUA/HIPAA
MedChain (Shen et al., 2019)	*Blockchain* híbrida + P2P	Off-chain (P2P DHT)	Baseado em sessão, revogável	Implícita (elim. off-chain)	Sim	Condições laboratoriais	Não espec.
Theodouli et al. (2018)	*Blockchain* (não espec.)	Off-chain + hashes on-chain	Controlo acesso smart contract	Eliminação off-chain	Não	Não	UE/RGPD
Zhang e Lin (2018)	Consórcio	On-chain cifrado + ABE	Acesso baseado em atributos	Controlo de chaves ABE	Parcial	Não	Não espec.
Oki et al. (2024)	Hyperledger Fabric	On-chain (ledger Fabric)	Chaincode baseado em papéis	Nenhuma	Sim	Ambiente hospitalar real	África do Sul
MedBlock	**Hyperledger Fabric 2.5**	**FHIR off-chain + consent. on-chain**	**Chaincode grant/revoke/query + identidade CMD**	**Elim. off-chain + revogação consent.**	**Sim (funcional)**	**Sintético + testes segurança/desemp.**	**Portugal (SNS, RGPD)**

A análise comparativa revela quatro lacunas persistentes que nenhuma das contribuições analisadas aborda de forma abrangente e que, em conjunto, definem o espaço de investigação ocupado pelo MedBlock.

Em primeiro lugar, a literatura empírica não apresenta casos específicos de implementação num sistema nacional de saúde europeu. O MedRec foi concebido de acordo com o quadro do HIPAA dos Estados Unidos (Azaria et al., 2016) O estudo do Hospital Frere foi realizado no âmbito da governação dos cuidados de saúde da África do Sul (Oki et al., 2024). O MedChain e Zhang e Lin não visaram nenhum contexto regulatório ou institucional específico (Shen et al., 2019; Zhang e Lin, 2018). Embora Theodouli et al. tenham considerado os princípios europeus de proteção de dados, o trabalho não produziu um protótipo nem uma avaliação no âmbito de um sistema nacional específico. O panorama dos cuidados de saúde portugueses, caracterizado pela estrutura de governação público-privada no âmbito do SPMS, pela coexistência da participação obrigatória do setor público com o envolvimento voluntário do setor privado e pela infraestrutura digital estabelecida (migração de HL7 V2.5 para FHIR, RSE Área do Cidadão, receitas eletrónicas),

apresenta constrangimentos que nenhuma solução existente aborda (TEHDAS — Towards the European Health Data Space, 2023).

Em segundo lugar, nenhum dos sistemas analisados integra um mecanismo nacional de identidade digital na camada de controlo de acesso da *blockchain*. O MedRec baseia-se em pares de chaves Ethereum para a identificação; o MedChain e o estudo de Zhang e Lin utilizam esquemas de identidade criptográfica personalizados; o estudo do Hospital Frere emprega o MSP do Fabric com credenciais ao nível da instituição. A integração do MedBlock com o Autenticação.Gov e a CMD como mecanismo de autenticação do paciente constitui uma combinação inovadora que fundamenta a identidade do paciente na infraestrutura nacional de identificação eletrónica, em vez de credenciais específicas da plataforma (Agência para a Modernização Administrativa (AMA), 2026).

Em terceiro lugar, a integração entre o armazenamento de dados clínicos baseado em FHIR e a gestão de consentimentos do Hyperledger Fabric ainda não foi demonstrada. O MedRec suporta o HL7 FHIR como padrão de interoperabilidade, mas não implementou um servidor FHIR como principal repositório de dados clínicos. O estudo do Hospital Frere armazenou os dados diretamente no *ledger* do Hyperledger Fabric. A arquitetura do MedBlock utiliza servidores HAPI FHIR R4 para dados clínicos fora da cadeia, ligados à rede Fabric por meio de canais protegidos por mTLS, com o código de cadeia restrito exclusivamente a operações de consentimento (conceder, revogar, consultar). Esta separação estabelece uma fronteira arquitetónica clara entre dados clínicos e metadados de consentimento, facilitando tanto a conformidade com o RGPD como a modularidade do sistema.

Em quarto lugar, os estudos empíricos analisados careciam de avaliação de segurança (Azaria et al., 2016; Shen et al., 2019) ou não realizaram testes de segurança estruturados conforme quadros estabelecidos (Oki et al., 2024). O desenho de avaliação do MedBlock incorpora uma avaliação de segurança estruturada em torno do NIST *Cybersecurity Framework* (National Institute of Standards and Technology, 2018) e do OWASP *Web Security Testing Guide* (OWASP Foundation, 2020), proporcionando uma metodologia replicável que atende às normas reconhecidas pela indústria.

3.5 RESUMO DO CAPÍTULO

Este capítulo analisou as principais contribuições e limitações dos sistemas existentes de partilha de dados de saúde baseados em *blockchain* e das estruturas de *blockchain*

autorizadas. A análise mostra que, embora o campo tenha produzido propostas arquitetonicamente diversas, que vão desde o modelo centrado no paciente em cadeia pública da MedRec (Azaria et al., 2016) até ao *design* híbrido de *blockchain*-P2P da MedChain (Shen et al., 2019) e ao esquema de encriptação em consórcio de Zhang e Lin (Zhang e Lin, 2018), nenhum sistema existente aborda simultaneamente as quatro dimensões necessárias para a implementação no contexto dos cuidados de saúde português: conformidade regulamentar nacional com o RGPD, integração com a infraestrutura digital e os sistemas de identidade existentes do SNS, gestão de consentimento baseada no Hyperledger Fabric com interoperabilidade FHIR e uma avaliação de segurança estruturada. Os capítulos seguintes descrevem como a plataforma MedBlock é concebida, implementada e avaliada para colmatar estas lacunas.

4

METODOLOGIA

Este capítulo explica os métodos utilizados para conceber, construir e avaliar a plataforma MedBlock. Os detalhes práticos, como o código-fonte e a implementação, encontram-se detalhados no Capítulo 5.

4.1 METODOLOGIA DE INVESTIGAÇÃO: DESIGN SCIENCE RESEARCH

Esta tese utilizou a *Design Science Research Methodology* (DSRM) descrita por Peffers et al. Ao contrário de outras abordagens, a DSRM visa construir uma ferramenta útil para resolver um problema real (Hevner et al., 2004). Isto é coerente com o objetivo deste trabalho: criar uma plataforma *blockchain* funcional, em vez de testar ou observar sistemas existentes.

A DSRM oferece uma forma clara de construir e testar uma potencial solução para um problema específico, evitando que essa solução seja utilizada numa organização real antes da avaliação (Peffers et al., 2007).

O processo DSRM tem seis etapas, cada uma correspondente a uma fase do projeto MedBlock. Inicialmente, foi identificado o problema através do estudo das lacunas dos dados nos cuidados de saúde portugueses, do conflito entre as funcionalidades da *blockchain* e o RGPD e da revisão de outras plataformas *blockchain* na área da saúde. Depois, foram definido os requisitos da solução, incluindo o acesso baseado em consentimento, a eliminação de dados conforme o RGPD, a interoperabilidade via FHIR e a integração com o sistema de identificação de Portugal. As etapas seguintes, de conceção e desenvolvimento, seguidas da demonstração da solução, estão abrangidas pela arquitetura descrita neste capítulo e pelo processo de construção detalhado no Capítulo 5. Posteriormente, a avaliação inclui as verificações de segurança. Finalmente, a comunicação concretiza-se na própria tese.

A DSRM é flexível e permite que os investigadores repitam etapas, se necessário. As etapas são apresentadas em ordem, mas podem ser revisitadas caso surjam dificuldades (Peffers et al., 2007).

4.2 IDENTIFICAÇÃO DO PROBLEMA E LEVANTAMENTO DE REQUISITOS

O problema de investigação foi identificado através de três vertentes de análise, cada uma delas fornecendo um conjunto de requisitos distintos para a plataforma. A seguir, cada vertente é apresentada em detalhe.

A primeira vertente envolveu uma análise sistemática do panorama de partilha de dados de saúde em Portugal. Apesar dos avanços relativos à prescrição eletrónica e no portal do RSE, ainda há muitos registos de dados e sistemas de informação distintos que funcionam em simultâneo apenas no setor público. Muitos assentam em normas de dados heterogéneas que abrangem diferentes gerações tecnológicas (TEHDAS — Towards the European Health Data Space, 2023). Os prestadores de cuidados de saúde privados atendem cerca de 20% da população e não têm qualquer obrigação legal de partilhar dados com o SNS (TEHDAS — Towards the European Health Data Space, 2023). Esta assimetria estrutural, entre instituições públicas regidas pelas políticas do SPMS e instituições privadas a operar de forma autónoma, definiu um requisito arquitetural central: a plataforma deve suportar modelos de participação, tanto obrigatórios como voluntários, numa única rede permissionada.

De seguida, analisámos o quadro legal. O RGPD estabelece regras para a utilização e proteção de dados pessoais na União Europeia. O Artigo 5.º estipula que apenas os dados necessários devem ser utilizados para fins específicos (European Parliament and Council of the European Union, 2016). O Artigo 9.º classifica os dados de saúde como dados sensíveis, sujeitos a uma proteção ainda maior. O Artigo 17.º consagra o direito ao esquecimento dos dados, ou seja, a possibilidade de eliminação dos dados pessoais a pedido do titular, o que é difícil de implementar em sistemas *blockchain* devido à natureza imutável dessas redes, constituindo um desafio central identificado por Xi et al. O Artigo 25.º determina que a privacidade deve ser integrada nos sistemas desde a sua conceção (*privacy by design*). O Artigo 32.º exige segurança proporcional ao risco. Todas estas regras implicam que o sistema deve manter os dados clínicos possíveis de apagar separados dos dados de consentimento permanentes, controlar o acesso ao nível da rede e fornecer prova do cumprimento das normas.

A terceira análise consistiu numa revisão dos sistemas *blockchain* existentes para dados de saúde (ver Capítulo 2). Examinámos seis sistemas principais: a *framework* de Cyran, o MedRec (Azaria et al., 2016), o MedChain (Shen et al., 2019), o *design* de Theodouli et al., a abordagem de consórcio de Zhang e Lin e o projeto Hyperledger Fabric de Oki et al. Verificámos que nenhum abordava, em simultâneo,

a conformidade com o RGPD, o apagamento de dados e a interoperabilidade FHIR. A Tabela 3.1 do Capítulo 3 apresenta estas lacunas. É nelas que o MedBlock visa atuar.

Estes requisitos foram divididos em requisitos funcionais (como a gestão do consentimento, a utilização de dados FHIR e o início de sessão do paciente) e não funcionais (como a conformidade com o RGPD, a segurança, a auditoria, a confidencialidade e o suporte à elevada carga). Para as necessidades não funcionais, a *framework* S3EF-HBCA agrupou-as em segurança (proteção, autenticação, redução de risco), privacidade (restrição da utilização de dados, consentimento e prazos de conservação de dados) e confiança (fiabilidade, conformidade, confiança do utilizador final)(Ramachandran, 2023).

A escolha da *framework blockchain* foi uma decisão de *design* fundamental. *Blockchain* define os limites de privacidade, a abordagem de consenso, os *smart contracts* e a gestão de identidade na plataforma. Foram considerados cinco *frameworks blockchain* permissionados: Hyperledger Fabric, Ethereum, Quorum, R3 Corda e MultiChain.

O Hyperledger Fabric 2.5 foi escolhido por três principais razões. Em primeiro lugar, a sua funcionalidade de canal permite que as organizações vejam apenas os dados específicos do canal. Isso contribui para o cumprimento dos requisitos de minimização de dados do RGPD (Artigo 5.º). Na área da saúde, significa que os prestadores podem partilhar registos de consentimento de forma segura. Em segundo lugar, o MSP gere as identidades digitais de todos os utilizadores, facilita a ligação com autoridades de certificação, sem necessidade de ferramentas de identidade adicionais. Em terceiro lugar, o Fabric não utiliza criptomoeda, pelo que não há taxas de transação. (Polge et al., 2021). As *frameworks* rejeitadas apresentavam limitações específicas em relação aos requisitos do projeto.

4.2.1 *Seleção da Norma de Dados Clínicos*

A escolha da norma de representação de dados clínicos determina como a informação de saúde é estruturada, partilhada e consultada na camada *off-chain* de dados. O HL7 FHIR Release 4 (R4) foi adotado como padrão para todos os dados clínicos armazenados fora da *blockchain* (HL7 International, 2019). Três fatores justificam esta escolha.

Em primeiro lugar, a arquitetura RESTful do FHIR é consistente com as necessidades contemporâneas de serviços *web*, permitindo operações *Hypertext Transfer Protocol* (HTTP) padrão (GET, PUT, POST, DELETE) sobre recursos clínicos discretos como Patient, Appointment, DiagnosticReport e MedicalRecord. Esta granularidade permite um controlo de acesso ao nível do recurso. Em segundo lugar, o FHIR R4 é a direção para a qual o SPMS está a migrar a sua infraestrutura de saúde digital (Martins, 2020; TEHDAS — Towards the European Health Data Space, 2023). A adoção do FHIR posiciona a arquitetura MedBlock como complementar, e não concorrente, ao ecossistema de dados de saúde português existente. Em terceiro lugar, o regulamento do EEDS, com aplicação prevista a partir de 2027, estabelece normas de interoperabilidade que convergem no formato FHIR como formato de troca preferencial tanto para dados de saúde de utilização primária quanto para a secundária (SPMS — Serviços Partilhados do Ministério da Saúde, 2023). Ao adotar o FHIR R4, a plataforma antecipa a conformidade regulatória em vez de exigir adaptações futuras.

4.2.2 *Seleção de Identidade e Autenticação*

A autenticação dos pacientes constitui uma fronteira crítica para a confiança em qualquer plataforma de partilha de dados de saúde. A abordagem adotada integra a CMD de Portugal, o mecanismo de identidade digital nacional gerido pela AMA (Agência para a Modernização Administrativa (AMA), 2026). A CMD fornece autenticação multifator através de uma combinação de número de telemóvel, número de identificação pessoal e código de segurança de utilização única e já se encontra amplamente adotada nos serviços digitais públicos portugueses, incluindo portais de saúde, administração fiscal e segurança social.

A justificação para integrar a CMD, em vez de um mecanismo de autenticação proprietário, baseia-se em três razões. Em primeiro lugar, utiliza uma infraestrutura de identidade que os cidadãos portugueses já possuem e conhecem, eliminando o atrito de acesso que poderia inibir a adoção da plataforma. Em segundo lugar, a CMD fornece uma identidade digital juridicamente reconhecida, ao abrigo da lei portuguesa, reforçando, consequentemente, o valor probatório das transações de consentimento registadas na *blockchain*. O consentimento de um paciente autenticado via CMD tem o mesmo peso legal que um documento assinado digitalmente. Em terceiro lugar, a infraestrutura baseada em certificados da CMD articula-se naturalmente com a camada MSP do Hyperledger Fabric: após uma autenticação

CMD bem-sucedida, a camada de aplicação pode mapear a identidade do cidadão autenticado para as credenciais de inscrição no Hyperledger Fabric correspondentes, ligando o domínio de identidade nacional ao domínio de identidade *blockchain*, sem criar um repositório de credenciais separado.

4.3 DECISÕES DE *DESIGN* ARQUITETURAL

Esta secção documenta as decisões de *design* que estruturam a arquitetura da plataforma MedBlock.

4.3.1 *Justificação da adoção de blockchain*

A literatura propõe diversos esquemas formais para esta decisão, sendo o de Wüst e Gervais o mais citado em contextos de partilha de dados sensíveis (Wüst e Gervais, 2018). Trata-se de uma árvore de decisão, reproduzida na Figura 2.1, composta por seis questões encadeadas que discriminam, de forma progressiva, entre base de dados convencional, *blockchain* permissionada pública e *blockchain* permissionada privada. Esta árvore foi caracterizada na Secção 2.1.4 e é aplicada nesta secção ao caso concreto do MedBlock.

A primeira questão pergunta se o sistema necessita de manter estado persistente. A resposta é afirmativa: as autorizações organizacionais para acesso a dados clínicos, os registos imutáveis dos eventos de acesso e a evolução do consentimento do paciente ao longo do tempo constituem estado que tem de ser preservado entre interações sucessivas.

A segunda questão pergunta se existem múltiplos escritores. No contexto do SNS, a resposta é também afirmativa. Hospitais públicos, hospitais privados e centros de saúde, cada um deles uma organização independente, poderão escrever autorizações no sistema. Esta diversidade de participantes é precisamente o que torna inviável a centralização da autoridade num único operador.

A terceira questão pergunta se está disponível um intermediário de confiança permanentemente *online*. A resposta é negativa. A assimetria de governação entre prestadores públicos e privados, descrita na Secção 2.5, torna inviável a designação de uma única entidade como árbitro permanente em quem todas as partes confiem. Ainda que os SPMS desempenhem um papel central na infraestrutura digital pública,

os prestadores privados operam sob regimes jurídicos independentes e não estão obrigados a delegar a custódia dos seus dados clínicos numa autoridade central.

A quarta questão pergunta se os participantes na escrita do *ledger* são conhecidos. A resposta é afirmativa. As organizações hospitalares são entidades reguladas, com identidade institucional verificável; os pacientes autenticam-se através da CMD, com base em certificados X.509 emitidos pela infraestrutura nacional de identidade eletrónica.

A quinta questão pergunta se todos os escritores são confiáveis. A resposta é negativa. A negação não significa que se atribua má-fé deliberada às organizações participantes; significa, antes, que a confiança institucional varia entre prestadores e que o histórico de incidentes de segurança no setor de saúde justifica desconfiança operacional.

A sexta e última questão pergunta se é necessária verificabilidade pública. A resposta é negativa. Os dados envolvidos são clínicos e sensíveis, sujeitos ao RGPD, pelo que a sua leitura tem de estar restrita a participantes autorizados.

A combinação destas seis respostas conduz, segundo a árvore de Wüst e Gervais, a uma *blockchain* permissionada privada. Esta conclusão fundamenta tecnicamente a opção pelo Hyperledger Fabric.

4.3.2 *Estratégia de Separação de Dados On-Chain e Off-Chain*

A decisão arquitetural mais determinante recaiu sobre a distribuição dos dados entre o *ledger* da *blockchain* e os sistemas de armazenamento externos. Foram consideradas duas estratégias gerais: armazenar todos os dados (clínicos e de consentimento) *on-chain* ou adotar um modelo híbrido, em que os metadados de consentimento são armazenados *on-chain*, mantendo os recursos clínicos FHIR *off-chain*.

Uma abordagem totalmente *on-chain* foi rejeitada por duas razões. Em primeiro lugar, armazenar dados clínicos diretamente no *ledger* do Hyperledger Fabric tornaria tecnicamente inviável a conformidade com o Artigo 17.º do RGPD. A estrutura *append-only* da *blockchain* impossibilita a eliminação retroativa de registos individuais sem comprometer a integridade criptográfica da cadeia (Xi et al., 2022). Em segundo lugar, os registos clínicos, que podem incluir estudos de imagiologia, resultados laboratoriais e históricos de pacientes, podem ter grande volume digital. Colocar estes dados *on-chain* degradaria o desempenho do *ledger* e aumentaria os requisitos de armazenamento em todos os *peer nodes*, o que é uma preocupação pelas

limitações de escalabilidade identificadas na implementação no Hospital Provincial de Frere (Oki et al., 2024).

A arquitetura híbrida adotada armazena na *blockchain* apenas os metadados de consentimento e autorização: os pacientes e as autorizações concedidas às organizações para aceder aos seus dados, quando a autorização foi concedida e quando (se alguma vez) foi revogada. Os dados clínicos residem em servidores HAPI por organização, cada um operando como um repositório de dados *off-chain* independente. Quando um prestador de cuidados de saúde solicita dados de um paciente, a camada de aplicação consulta primeiro a *blockchain* para verificar a existência de um consentimento válido e não revogado; apenas após a verificação bem-sucedida, recupera os recursos FHIR correspondentes do servidor *off-chain* relevante.

Esta separação resolve diretamente a tensão decorrente do direito de esquecimento previsto no RGPD. Quando um paciente exerce o direito ao esquecimento, os registos clínicos *off-chain* poderão ser eliminados do servidor FHIR recorrendo de operações de base de dados padrão, cabendo à responsabilidade de cada organização de saúde. O registo de consentimento *on-chain* está marcado como revogado, mas não eliminado; permanece como prova auditável de que o consentimento existiu e foi posteriormente revogado. Uma vez que o registo *on-chain* não contém dados clínicos (apenas identificadores organizacionais, identificadores de inscrição do paciente e marcas temporais), a sua manutenção não constitui tratamento continuado de dados de saúde do Artigo 9.º (European Parliament and Council of the European Union, 2016).

4.3.3 *Design da Topologia de Rede*

A rede Hyperledger Fabric foi concebida com quatro organizações: HospitalPublico (`HospitalpublicoMSP`), HospitalPrivado (`HospitalprivadoMSP`), CentrodeSaude (`CentrodesaudeMSP`) e Medblock (`MedblockMSP`). Este modelo de quatro organizações não foi uma configuração arbitrária; foi selecionado para representar os três principais níveis do sistema de saúde português: hospitais públicos, hospitais privados e centros de saúde, acrescidos de uma organização de administração da plataforma, responsável pela governação da rede, pela gestão do ciclo de vida do *chaincode* e pela coordenação interorganizacional. O modelo de quatro organizações é a topologia mínima que exercita a assimetria público-privado (HospitalPublico e CentrodeSaude representam instituições governadas publicamente; HospitalPri-

vado representa uma entidade privada de participação voluntária), incluindo uma entidade de governação (Medblock).

Todas as quatro organizações participam num único canal, designado `medblock-chain`, representado na Figura 4.1. Uma arquitetura de canal único foi adotada no protótipo porque o modelo de consentimento exige que todas as organizações participantes tenham visibilidade do estado de autorização. As arquiteturas multicanal, em que os dados dos pacientes permanecem isolados por canal, foram consideradas para a escalabilidade futura, mas consideradas prematuras para a fase de protótipo funcional.

O serviço de ordenação utiliza um *orderer* Solo que processa as transações através de um único *ordering node*, sem tolerância a falhas bizantinas. O ordenamento Solo foi selecionado para a fase de prototipagem porque minimiza a complexidade da infraestrutura e evita a sobrecarga operacional de configurar e manter um *cluster* Raft ou BFT. Esta escolha é metodologicamente adequada para um protótipo de *design* cujo objetivo principal é demonstrar a viabilidade funcional, e não a resiliência em nível de produção. Cada organização opera um único *anchor peer* configurado para a comunicação interorganizacional através do protocolo *gossip*, o que permite a sincronização do *ledger*.

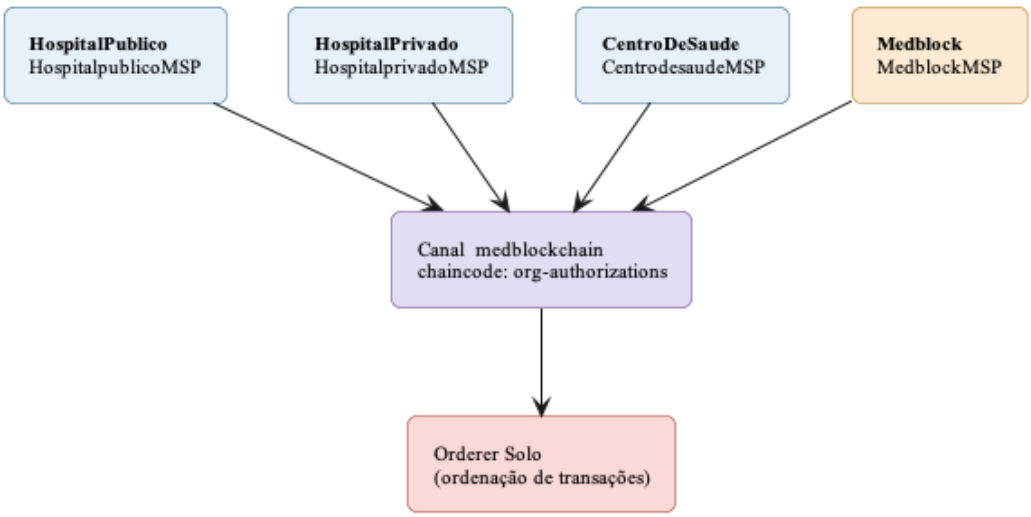


Figura 4.1: Topologia da rede da plataforma MedBlock.

4.3.4 *Estratégia de Canal de Staging*

Um canal Hyperledger Fabric paralelo foi provisionado como ambiente de *staging* para todas as atividades de validação funcional, testes de segurança e *benchmarking*

de desempenho. Este canal de *staging*, distinto do canal de produção (medblockchain), replica a implementação idêntica de *chaincode*, políticas de *endorsement* e membros do ambiente de produção.

A justificação desta decisão de *design* baseia-se em uma propriedade fundamental das *blockchains* permissionadas: a imutabilidade do *ledger*. Duas das três funções de *chaincode*, `AddAuthorizedOrg` e `RemoveAuthorizedOrg`, executam operações de escrita que alteram permanentemente o *world state* e registam transações na *blockchain*. O *benchmarking* de desempenho via Hyperledger Caliper gera centenas ou milhares dessas transações em rápida sucessão. Executar estas cargas de trabalho diretamente no canal de produção contaminaria o *ledger* com dados fictícios de teste, tornando impossível distinguir registos de consentimento genuínos e artefactos de *benchmarking*. Numa plataforma cuja conformidade regulamentar depende da auditabilidade, esta contaminação comprometeria, sobretudo, a integridade do registo de consentimento.

O canal de *staging* resolve esta preocupação ao oferecer um ambiente isolado, porém estruturalmente idêntico. Uma vez que os canais do Hyperledger Fabric mantêm *ledgers*, *world states* e históricos de transações de forma independente, a atividade no canal de *staging* não afeta os dados do canal de produção. No entanto, o canal de *staging* partilha a mesma infraestrutura de pares, autoridades de certificação e configuração de rede, o que significa que as características de desempenho observadas no canal de *staging* são representativas do comportamento em produção, uma vez que as transações percorrem o mesmo tipo de *endorsement* e *ordering*.

Esta abordagem assemelha-se ao fluxo de passagem de *staging* para produção, prevalente na entrega de *software* empresarial, e é adaptada às restrições da tecnologia de *ledger* distribuído, em que, ao contrário das bases de dados convencionais, as operações de *rollback* e de limpeza de dados são arquiteturalmente impossíveis. Todas as atividades de avaliação foram conduzidas exclusivamente no canal de *staging*. Apenas após a verificação dos resultados e a aplicação de iterações corretivas, o canal de produção foi utilizado para a demonstração final da plataforma validada.

4.3.5 *Estratégia de Design do Chaincode*

A camada de *smart contract*, denominada *chaincode* na terminologia do Hyperledger Fabric, foi concebida com base no conjunto mínimo de operações necessárias para implementar um modelo de controlo de acesso baseado no consentimento.

Foram definidas três funções: `AddAuthorizedOrg`, que regista o consentimento de um paciente para que uma organização específica aceda aos seus dados clínicos; `RemoveAuthorizedOrg`, que revoga um consentimento previamente concedido; e `GetAuthorizedOrgs`, que consulta o estado atual de autorização de um determinado paciente. Este *design* reflete o ciclo de vida do consentimento, concessão, revogação e verificação e exclui deliberadamente as operações de dados clínicos do âmbito do *chaincode*, em consonância com a estratégia de separação *on-chain*/*off-chain*.

Importa reconhecer uma limitação assumida no decurso deste projeto: a estratégia inicial previa a implementação adicional de *chaincode* orientado às organizações, complementar ao conjunto de funções centradas no paciente, mas tal componente não chegou a ser concretizado devido às restrições temporais do projeto. Esse *chaincode* permitiria a cada organização listar os pacientes que lhe concederam. Esta funcionaldade constitui uma extensão a desenvolver em trabalho futuro.

O Hyperledger Fabric 2.5 suporta o desenvolvimento de *chaincode* em Go, Java e TypeScript. O TypeScript foi preferido por duas razões: em primeiro lugar, permite a partilha de código com a camada da aplicação *web*, que também opera em Node.js, reduzindo a sobrecarga de manter contextos de linguagem separados na plataforma; em segundo lugar, as bibliotecas `fabric-contract-api` e `fabric-shim` para Node.js fornecem uma abstração bem documentada do protocolo de comunicação entre *peers* do Hyperledger Fabric.

A política de *endorsement* foi configurada por MAJORITY no nível da aplicação, exigindo que a maioria dos pares das organizações fizesse o *endorsement* de cada transação antes de ser confirmada no *ledger*. Esta política equilibra dois requisitos concorrentes. Um requisito de unanimidade maximizaria a confiança, embora tornasse a rede vulnerável à indisponibilidade de uma única organização que bloqueasse todas as operações de consentimento, uma fragilidade num contexto de saúde, onde o acesso aos registos dos pacientes é crítico. O *endorsement* de uma única organização maximizaria a disponibilidade, mas enfraqueceria o modelo de confiança ao permitir que uma organização comprometida forjasse, unilateralmente, registos de consentimento. O *endorsement* por maioria ocupa o meio-termo, garantindo que nenhuma organização isolada possa manipular o *ledger*, ao mesmo tempo em que tolera a indisponibilidade temporária de uma minoria de *peers*.

4.3.6 *Design da Arquitetura de Segurança*

A arquitetura de segurança foi concebida como um modelo de defesa em camadas, incluindo controlos criptográficos ao nível da rede e da aplicação.

Na camada criptográfica, foi estabelecida uma PKI em dois níveis. Uma *Certificate Authority* (CA) Raiz atua como raiz de confiança para toda a rede, emitindo certificados para CAs intermédias operadas por cada organização (`ca.hospitalpublico`, `ca.hospitalprivado`, `ca.centrodesaude` e a CA do `ca.orderer`). Esta estrutura hierárquica permite a gestão independente de certificados por organização; cada entidade pode inscrever e revogar os seus próprios utilizadores e *peers*, mantendo uma cadeia de confiança unificada sob uma única autoridade. O certificado da CA raiz é distribuído a todos os pares e *orderers* como raiz de confiança, garantindo que qualquer *peer* possa verificar a identidade de qualquer outro participante da rede, independentemente da sua afiliação organizacional.

O certificado do cliente deve estar encadeado à CA raiz, garantindo que apenas entidades com credenciais válidas na rede Hyperledger Fabric possam aceder aos *endpoints* de dados clínicos. Esta aplicação de mTLS impede o acesso externo não autorizado, mesmo que o endereço de rede do servidor FHIR venha a ser descoberto.

Como camada secundária, a validação por chave de *Application Programming Interface* (API) complementa o mecanismo mTLS nos *proxies* FHIR. A cada organização é atribuída uma chave de API única que deve ser apresentada no cabeçalho HTTP `X-Api-Key` em cada pedido. Esta medida de defesa em profundidade fornece um controlo ao nível da aplicação que opera independentemente da camada TLS; mesmo um cliente com um certificado de rede válido não pode aceder aos recursos FHIR sem a chave de API correspondente. O duplo requisito (certificado válido e chave de API válida) garante que o comprometimento de um único tipo de credencial não é suficiente para obter acesso aos dados.

Na camada de aplicação, a aplicação *web* implementa uma gestão de sessões encriptada para proteger o estado de autenticação do utilizador. Os *tokens* de sessão são assinados criptograficamente e encriptados em repouso, o que previne o sequestro de sessão através do roubo de *tokens* ou da manipulação de *cookies*.

4.4 ESTRATÉGIA DE CONFORMIDADE REGULAMENTAR

A conformidade regulamentar na arquitetura MedBlock não é uma consideração posterior aplicada a um *design* concluído; constitui, antes, um conjunto de restrições de primeira ordem que moldaram as decisões arquiteturais desde o início. O Artigo 25.º do RGPD exige explicitamente esta abordagem prospectiva, determinando que a proteção de dados seja integrada no desenvolvimento, e não introduzida como medida corretiva após a implementação (European Parliament and Council of the European Union, 2016). Esta secção descreve como cada disposição relevante do RGPD foi traduzida numa decisão de *design* concreta.

O princípio da minimização de dados (Artigo 5.º, n.º 1, alínea c)) é aplicado recorrendo a dois mecanismos. Ao nível da *blockchain*, os registos de consentimento *on-chain* contêm apenas os dados mínimos necessários para estabelecer e verificar uma autorização: o identificador do paciente, o identificador da organização autorizada e um *timestamp*. Nenhum dado clínico, pessoal ou qualquer informação além do estritamente necessário à gestão do consentimento é armazenado no *ledger*.

As proteções reforçadas para categorias especiais de dados pessoais (Artigo 9.º) são implementadas através da combinação de acesso mTLS aplicado aos servidores FHIR, verificação de consentimento mediada por *chaincode* antes de qualquer recuperação de dados e a ausência de dados de saúde na própria *blockchain*. Uma vez que o *ledger* não contém dados ao abrigo do Artigo 9.º, as condições de tratamento rigorosas que se aplicam aos dados de saúde apenas precisam de ser cumpridas na camada de acesso *off-chain*, onde podem ser aplicados controlos padrão no nível da base de dados.

O direito ao esquecimento (Artigo 17.º) é assegurado recorrendo da arquitetura híbrida *on-chain*/*off-chain*. Os dados clínicos armazenados nos servidores HAPI podem ser eliminados através de operações convencionais de base de dados. O registo de consentimento *on-chain* é revogado em vez de eliminado, criando um registo auditável que demonstra tanto a existência de consentimento prévio como a sua retirada subsequente.

O requisito de segurança do tratamento (Artigo 32.º) é implementado pela arquitetura de segurança em camadas: gestão de identidade baseada em PKI, aplicação de mTLS, validação por chave de API e gestão de sessões encriptadas. Estas medidas são proporcionais à natureza de elevado risco do tratamento de dados de saúde e destinam-se a demonstrar a conformidade técnica com o requisito de que

os responsáveis pelo tratamento implementem medidas organizacionais e técnicas adequadas.

4.5 ESTRATÉGIA DE AVALIAÇÃO

Esta secção define a estratégia de avaliação aplicada ao protótipo MedBlock. A avaliação cobre duas dimensões complementares: o desempenho da rede *blockchain* sob carga e a postura de segurança da plataforma face a ameaças conhecidas.

4.5.1 *Princípio de Testes com Prioridade para o Staging*

Todas as atividades de avaliação descritas nesta secção - validação funcional, avaliação de segurança e *benchmarking* de desempenho - foram conduzidas no canal de *staging* antes de qualquer interação com o canal de produção.

4.5.2 *Abordagem à Avaliação de Desempenho*

A avaliação de desempenho utilizou o Hyperledger Caliper, uma *framework* de *benchmarking* da comunidade Hyperledger, como instrumento de medição. O Hyperledger Caliper foi selecionado porque se integra nativamente ao *Software Development Kit* (SDK) do Hyperledger Fabric, suporta definições de carga de trabalho configuráveis e reporta métricas padronizadas que permitem a comparação com resultados publicados na literatura académica (Hyperledger Foundation, 2026).

Foram definidas duas categorias de métricas para a avaliação do MedBlock. O débito (*throughput*), medido em *Transactions Per Second* (TPS), quantifica a taxa a que a rede pode processar operações de consentimento sob carga sustentada. A latência, medida em milissegundos, captura o tempo decorrido entre a submissão de uma transação e a sua confirmação, tanto para operações de invocação (`AddAuthorizedOrg`, `RemoveAuthorizedOrg`) quanto para operações de consulta (`GetAuthorizedOrgs`). A distinção entre a latência de invocação e de consulta é metodologicamente importante: as transações de invocação percorrem o *pipeline* completo de *endorsement-ordering*-confirmação e modificam o *world state*, ao passo que as transações de consulta são resolvidas localmente por um único *peer* sem modificar o *ledger*.

4.5.3 *Abordagem à Avaliação de Segurança*

A avaliação de segurança foi estruturada em torno de dois *frameworks* complementares: o NIST Cybersecurity Framework, versão 1.1 (National Institute of Standards and Technology, 2018), e o OWASP Web Security Testing Guide, versão 4.2 (OWASP Foundation, 2020).

O NIST Cybersecurity Framework oferece uma perspetiva organizacional de alto nível estruturada em torno de cinco funções: Identificar, Proteger, Detetar, Responder e Recuperar. Para a avaliação do MedBlock, as funções Identificar e Proteger são as mais diretamente relevantes, abrangendo a gestão de ativos, o controlo de acesso, a segurança de dados e a tecnologia de proteção. A avaliação mapeia cada controlo de segurança do MedBlock (hierarquia PKI, mTLS, validação por chave de API, verificação de consentimento baseada em *chaincode*, sessões encriptadas) para as subcategorias NIST correspondentes e avalia se os controlos implementados satisfazem as expectativas da *framework* para um sistema que trata dados de saúde de elevada sensibilidade.

O OWASP Testing Guide fornece uma metodologia de avaliação granular e tecnicamente orientada, focada em vulnerabilidades de aplicações *web*. Uma vez que a plataforma MedBlock inclui uma camada de aplicação *web* através da qual pacientes e profissionais de saúde interagem com o sistema, os testes orientados pelo OWASP foram essenciais. As categorias avaliadas incluem testes de autenticação (verificação de que a autenticação baseada em CMD não pode ser contornada), testes de autorização (verificação de que os limites de consentimento aplicados pelo *chaincode* não podem ser contornados através de manipulação direta da API), testes de gestão de sessão (verificação de que os *tokens* de sessão resistem a sequestro, fixação e ataques de *replay*) e testes de validação de entrada (verificação de que os parâmetros do *chaincode* não podem ser explorados através de injeção ou malformação).

4.6 AMBIENTE DE DESENVOLVIMENTO E FERRAMENTAS

Esta secção cataloga as principais ferramentas, *frameworks* e componentes de infraestrutura utilizados ao longo do projeto. Os números de versão e as características de configuração são especificados para garantir a reprodutibilidade por parte de outros investigadores.

Todos os componentes Fabric são contentorizados com Docker e orquestrados usando Docker Compose, permitindo que toda a rede multiorganização seja instanciada, desativada e recriada de forma determinística a partir de um único ficheiro de configuração.

A camada de dados clínicos *off-chain* é composta por instâncias do servidor HAPI, uma por organização de saúde, e é suportada por bases de dados PostgreSQL 16. O HAPI é a implementação de referência *open source* da especificação HL7 FHIR e foi selecionado pela sua conformidade com o FHIR R4 e pela manutenção ativa pela comunidade do FHIR. Cada servidor FHIR é acedido através de um *proxy* reverso Nginx (imagem baseada em Alpine), configurado para aplicar mTLS e validar com a chave da API.

O *chaincode* e a camada de aplicação são implementados em JavaScript a correr num *runtime* do Node.js. O *chaincode* utiliza as bibliotecas fabric-contract-api (versão 2.5.0) e fabric-shim (versão 2.5.0) do SDK do Hyperledger Fabric.

4.7 CRONOGRAMA DO PROJETO

O desenvolvimento do projeto MedBlock foi estruturado em cinco fases sequenciais, embora parcialmente sobrepostas, ao longo de um período de dezoito meses compreendido entre novembro de 2024 e abril de 2026. A Figura 4.2 apresenta o diagrama de Gantt que sintetiza a distribuição temporal destas fases. As sobreposições entre determinadas fases não são acidentais; refletem a natureza iterativa da metodologia DSRM adotada, em que os resultados de uma atividade alimentam e por vezes obrigam à revisão de atividades anteriores. A primeira fase, a revisão da literatura, decorreu entre novembro de 2024 e fevereiro de 2025. Neste período foram identificados, analisados e sintetizados os trabalhos académicos, os quadros regulamentares e as iniciativas institucionais relevantes para a partilha segura de dados de saúde baseada em *blockchain*. A revisão abrangeu as plataformas existentes, as *frameworks* de *blockchain* permissionada, os requisitos do RGPD e da HIPAA, e o estado da infraestrutura digital de saúde em Portugal. Os resultados desta fase alimentaram diretamente a identificação do problema e a elicitação de requisitos, bem como a análise comparativa que fundamentou a seleção tecnológica. A segunda fase, o *design* da plataforma, estendeu-se de janeiro a agosto de 2025 e constituiu a fase mais prolongada do projeto. A sobreposição inicial com a revisão da literatura foi intencional: à medida que as lacunas dos sistemas existentes se tornavam evidentes na análise bibliográfica, as decisões arquiteturais começaram a ser formuladas em

paralelo. Esta fase compreendeu a definição da topologia da rede Fabric com quatro organizações, a estratégia de separação de dados *on-chain*/*off-chain*, o *design* do *chaincode* de gestão de consentimento, a hierarquia de PKI com CA raiz e CAs intermédias, e o mapeamento sistemático dos requisitos do RGPD para decisões concretas de *design*. A duração alargada desta fase reflete as múltiplas iterações que a arquitetura sofreu, nomeadamente a revisão da política de endosso após testes preliminares no canal de *staging* e o refinamento da estratégia de autenticação após a análise das restrições da integração com a CMD. A terceira fase, o desenvolvimento da plataforma, decorreu entre junho e dezembro de 2025. A sobreposição com a fase de *design* é consequência direta do modelo iterativo: determinados componentes cuja arquitetura estava estabilizada, como a rede Fabric base, os servidores FHIR e a infraestrutura de contentorização, foram implementados enquanto outros elementos de *design*, como os controlos de segurança perimetral via Cloudflare *Web Application Firewall* (WAF) e a configuração definitiva do *proxy* Nginx com mTLS, continuavam a ser refinados. Esta fase materializou o protótipo funcional completo, incluindo o *chaincode*, o *backend* em TypeScript/Express, a integração com os servidores HAPI FHIR, a autenticação via CMD e a aplicação web. A quarta fase, os testes de segurança e *benchmarking*, ocupou o período de dezembro de 2025 a fevereiro de 2026. A ligeira sobreposição com o final da fase de desenvolvimento é explicada pelo facto de os primeiros ciclos de testes terem sido executados sobre componentes já concluídos enquanto os últimos refinamentos da aplicação web ainda estavam em curso. Todas as atividades de avaliação, nomeadamente testes funcionais, avaliação de segurança segundo o NIST *Cybersecurity Framework* e o OWASP *Web Security Testing Guide*, e *benchmarking* de desempenho com o Hyperledger Caliper, foram conduzidas exclusivamente no canal de *staging*, em conformidade com a estratégia de isolamento de testes adotada. A quinta e última fase, a documentação e elaboração da tese, decorreu progressivamente entre dezembro de 2024 e abril de 2026.

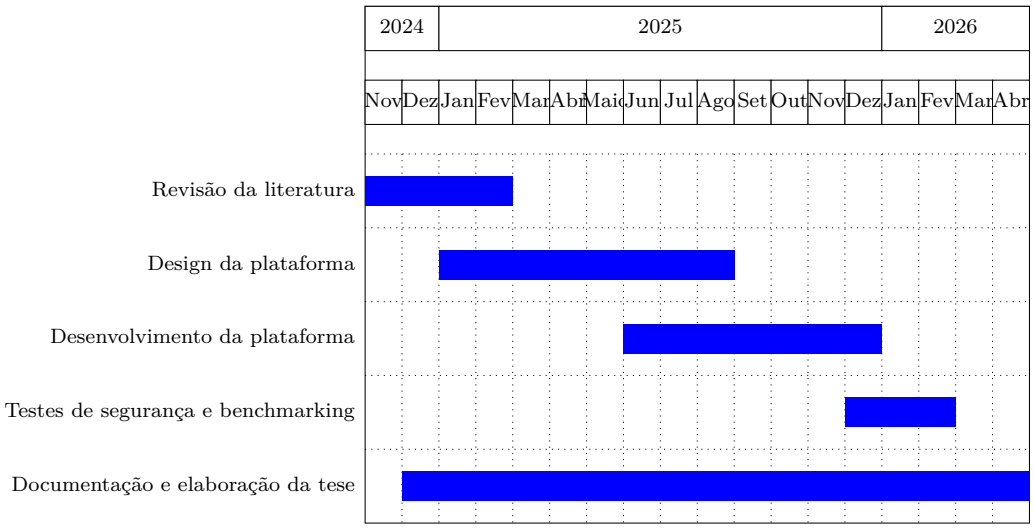


Figura 4.2: Diagrama de Gantt com o cronograma das fases da elaboração desta tese.

5

DESENVOLVIMENTO

Este capítulo apresenta a implementação da plataforma MedBlock. Traduz as decisões arquiteturais e a fundamentação de *design* estabelecidas no Capítulo 4 num protótipo funcional. Enquanto no capítulo anterior foi descrito o que cada componente deveria realizar e por que razão determinadas tecnologias foram selecionadas, este capítulo documenta como cada componente foi concretizado na prática. Especifica parâmetros de configuração, decisões ao nível do código, padrões de comunicação entre serviços e procedimentos operacionais.

A infraestrutura de rede e a contentorização surgem em primeiro lugar, formando a base para as restantes camadas. A PKI e a hierarquia de CAs seguem-se, uma vez que as identidades criptográficas por elas emitidas constituem pré-requisitos tanto para a *blockchain* como para as camadas de dados *off-chain*. A configuração de canal e a implementação de *chaincode* são abordadas em seguida, assim como os servidores de dados clínicos *off-chain* e a sua arquitetura de *reverse proxy*. O *backend* e o *frontend* da aplicação *web* são então documentados, demonstrando como as camadas de *blockchain* e FHIR se integram num sistema unificado orientado para o utilizador. Os detalhes de implementação de segurança, como o perímetro da Cloudflare, são consolidados na secção 5.8.

Ao longo deste capítulo, as referências a ficheiros, variáveis de ambiente e diretivas de configuração refletem o repositório do projeto na validação final do protótipo. Todas as versões das dependências externas estão registadas para garantir a reprodutibilidade.

5.1 INFRAESTRUTURA DE REDE E CONTENTORIZAÇÃO

Esta secção descreve a infraestrutura física e lógica sobre a qual o protótipo MedBlock opera. São abordadas as decisões de alojamento, a sua justificação no contexto de um protótipo e a organização dos serviços em *containers* Docker.

5.1.1 *Infraestrutura de Alojamento*

O protótipo MedBlock está alojado numa única instância de *Virtual Private Server* (VPS), executando o sistema operativo `Ubuntu Server 25.04`. Esta instância aloja simultaneamente dois componentes principais: a rede Hyperledger Fabric completa (incluindo todos os *peers*, o *orderer*, as autoridades de certificação e os servidores FHIR, contentorizados via Docker) e a aplicação *web* Node.js/Express. Ambos os serviços partilham, portanto, o mesmo endereço IP público, os mesmos recursos de processamento e de memória e o mesmo sistema de ficheiros.

A Figura 5.1 apresenta a topologia de *containers*, incluindo as ligações entre serviços na rede `hospital-net` e a comunicação entre camadas.

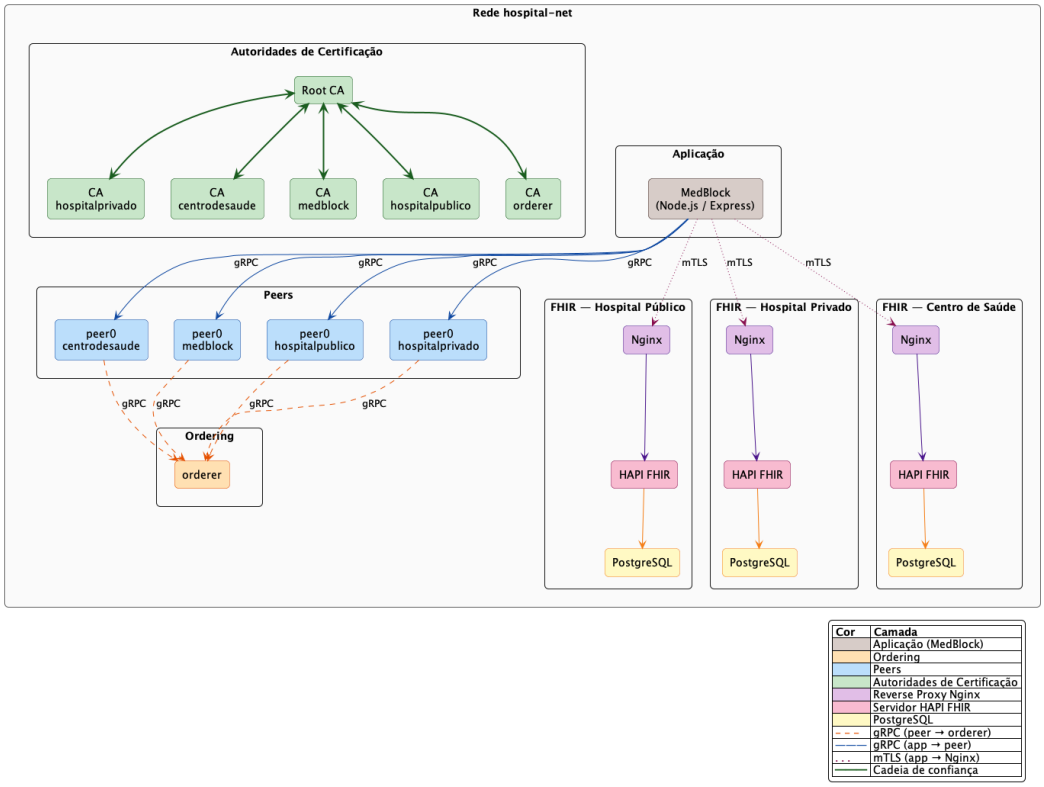


Figura 5.1: Topologia de *containers* Docker da rede MedBlock.

A decisão de consolidar todos os componentes numa única máquina virtual foi condicionada por um requisito funcional do protótipo: a integração com a CMD exige que a aplicação *web* esteja acessível num domínio público com um certificado TLS válido, uma vez que o fluxo *Open Authorization 2.0* (OAuth 2.0) da Autenticação.Gov redireciona o navegador para um *Uniform Resource Identifier* (URI) de retorno que deve ser acessível pela infraestrutura da AMA. O alojamento num VPS com o domínio `medblock.pt` satisfaz este requisito de forma direta,

evitando as complexidades de expor um ambiente de desenvolvimento local à *internet* pública.

Esta topologia de servidor único é explicitamente uma concessão ao caráter prototípico do projeto e não deve ser replicada em cenário de produção. Num sistema implementado no contexto real do SNS, a arquitetura deveria separar, no mínimo, três camadas em servidores distintos: a rede *blockchain* (*peers*, *orderer* e autoridades de certificação), os servidores de dados clínicos FHIR e a aplicação *web* orientada ao utilizador. Esta separação proporcionaria isolamento de falhas; uma indisponibilidade na camada de aplicação *web* não afetaria a operação da rede *blockchain* e permitiria a aplicação de políticas de segurança de rede, como regras de *firewall* e segmentação de rede específicas por camada. Adicionalmente, os requisitos de escalabilidade num cenário nacional implicariam a distribuição dos *peers* em múltiplas máquinas (idealmente geridas pelas respetivas organizações de saúde) e o balanceamento de carga na camada de aplicação *web*.

5.1.2 *Contentorização*

O protótipo MedBlock funciona como um ambiente *multi-container* orquestrado por Docker Compose (versão 2), unificado sob uma única rede Docker *bridge* denominada `hospital-net`. Esta rede permite a descoberta de serviços baseada em *Domain Name System* (DNS) entre *containers*, permitindo que cada serviço se comunique com outros serviços pelo *hostname*, em vez da atribuição manual de endereços IP. A utilização de uma rede única na fase de protótipo simplifica a comunicação entre serviços, preservando, contudo, a opção de introduzir segmentação de rede em produção.

O ambiente completo compreende 21 *containers*, agrupados em 5 camadas funcionais. A Tabela 5.1 lista todos os *containers*, as respetivas imagens Docker, os portos expostos e camada de contexto.

A camada do *ordering service* contém um nó Solo *orderer* (`orderer.example.com`) que executa a imagem do *orderer* do Hyperledger Fabric, na versão 2.5.10. O *orderer* escuta no porto 7050, vinculada exclusivamente à interface *loopback* (`127.0.0.1:7050`), para impedir o acesso externo direto. Toda a comunicação legítima com o *orderer* é originada dos *containers peers* na rede Docker. A camada de *peers* inclui quatro nós *peer*, um por organização: `peer0.hospitalpublico.example.com` (porto 7051), `peer0.hospitalprivado.example.com` (porto 9051), `peer0.centrodesaude.example.com` (porto 11051) e `peer0.medblock.example`

Tabela 5.1: Resumo dos *containers* Docker que compõem a rede MedBlock, agrupados por camada funcional.

Nome do Container	**Imagem Docker**	**Porto(s)**	**Camada**
cli	hyperledger/fabric-tools:2.5.10	—	CLI
orderer	hyperledger/fabric-orderer:2.5.10	7050	*Ordering*
peer0.hospitalpublico	hyperledger/fabric-peer:2.5.10	7051–7052	*Peers*
peer0.hospitalprivado	hyperledger/fabric-peer:2.5.10	9051–9052	*Peers*
peer0.centrodesaude	hyperledger/fabric-peer:2.5.10	11051–11052	*Peers*
peer0.medblock	hyperledger/fabric-peer:2.5.10	12051–12052	*Peers*
rca	hyperledger/fabric-ca:1.5	7040	CAs
ca.hospitalpublico	hyperledger/fabric-ca:1.5	7054	CAs
ca.orderer	hyperledger/fabric-ca:1.5	8054	CAs
ca.hospitalprivado	hyperledger/fabric-ca:1.5	9054	CAs
ca.centrodesaude	hyperledger/fabric-ca:1.5	11054	CAs
ca.medblock	hyperledger/fabric-ca:1.5	12054	CAs
fhir.hospitalpublico	hapiproject/hapi:latest	—	FHIR
fhir.hospitalprivado	hapiproject/hapi:latest	—	FHIR
fhir.centrodesaude	hapiproject/hapi:latest	—	FHIR
postgres.hospitalpublico	postgres:16	5433	FHIR
postgres.hospitalprivado	postgres:16	5434	FHIR
postgres.centrodesaude	postgres:16	5435	FHIR
nginx.hospitalpublico	nginx:alpine	8081	FHIR
nginx.hospitalprivado	nginx:alpine	8082	FHIR
nginx.centrodesaude	nginx:alpine	8083	FHIR

`.com` (porto 12051). Cada *peer* está configurado com TLS ativado, o *gossip* apontando para si próprio (topologia de *peer* único por organização) e a execução de *chaincode*.

A camada de CA compreende seis *containers*: uma CA raiz (`rca.example.com`, porto 7040) e cinco CAs intermediária, uma para cada uma das quatro organizações mais a organização do *orderer*. As imagens de CA utilizam a versão 1.5 do Hyperledger Fabric CA. Um *container* *Command Line Interface* (CLI) fornece acesso administrativo à rede Fabric para operações nos canais (configuração da rede), gestão do ciclo de vida de *chaincode* e consultas de diagnóstico (verificações do sistema).

A camada de dados clínicos *off-chain* implementa três *stacks* paralelos, um para cada organização de saúde. Cada *stack* é composta por uma base de dados PostgreSQL 16, um servidor HAPI FHIR R4 e um *reverse proxy* Nginx Alpine. As instâncias PostgreSQL expõem portos de *host* distintas (5433, 5434, 5435) para acesso administrativo. Comunicam-se com os respetivos servidores HAPI através da rede interna Docker no porto 5432. Os servidores HAPI FHIR funcionam como aplicações Spring Boot, com a configuração via variáveis de ambiente. Os *reverse proxies* Nginx comunicam-se por TLS nas portos 8081, 8082 e 8083 para o Hospital Público, o Hospital Privado e o Centro de Saúde, respetivamente.

5.2 *PUBLIC KEY INFRASTRUCTURE* E HIERARQUIA DE *CERTIFICATE AUTHORITY*

A PKI do MedBlock implementa uma hierarquia de certificados de duas camadas, centralizada numa única CA raiz e ramificando-se em cinco CAs intermediárias: uma para cada organização participante e outra para o *orderer*. Esta estrutura permite que cada organização faça a gestão, de forma independente, do ciclo de vida das suas próprias identidades, registando utilizadores, inscrevendo *peers* e revogando credenciais, mantendo, simultaneamente, uma cadeia de confiança unificada ancorada a um único certificado raiz. O significado arquitetural deste *design* foi delineado no Capítulo 4; esta secção documenta a sua implementação concreta. A Figura 5.2 apresenta visualmente a hierarquia, mapeando cada CA aos tipos de identidade que emite.

A CA raiz, configurada com um ficheiro `fabric-ca-server-config.yaml` na versão 1.5.15, TLS ativado e uma identidade de administrador. O certificado da CA raiz e o certificado TLS são os artefactos criptográficos de confiança: o primeiro é distribuído a cada *peer*, *orderer* e *proxy* Nginx como raiz das cadeias de confiança

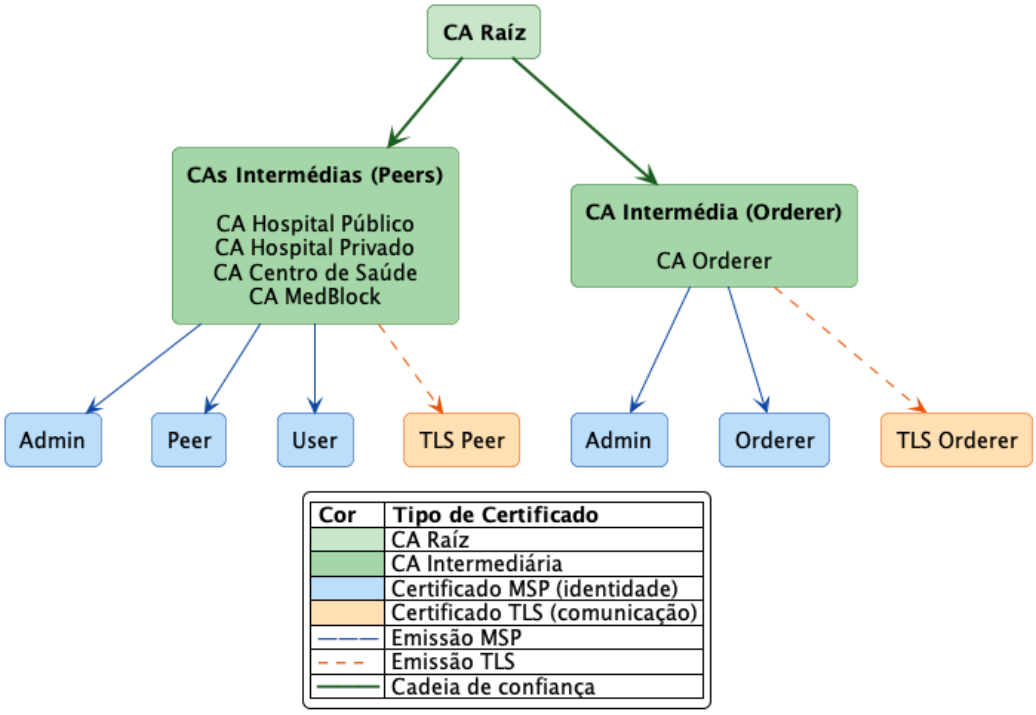


Figura 5.2: Hierarquia da infraestrutura de chaves públicas das entidades de certificação.

MSP e TLS, enquanto o segundo é utilizado pela ferramenta `fabric-ca-client` para estabelecer ligações seguras à CA raiz durante a inicialização.

O registo de identidade do *orderer* segue um procedimento paralelo, produzindo identidades Admin e *orderer* com credenciais MSP e TLS.

Um detalhe de implementação crítico envolve a criação de ficheiros de cadeia de certificados. A verificação TLS do Hyperledger Fabric exige que a cadeia completa de certificados, desde o certificado folha, passando pela CA intermediária até à CA raiz, seja apresentada durante os *handshakes* TLS. A implementação constrói o ficheiro `server-chain.crt` concatenando o certificado da CA intermediária e o certificado da CA raiz nessa ordem. Estes ficheiros de cadeia são montados tanto nos *containers* dos *peers* como no *container* do *orderer*, contendo certificados TLS, e são igualmente fornecidos aos *proxies* Nginx FHIR para verificação de mTLS. Apenas o certificado da CA raiz (`ca-cert.pem`) é distribuído como `ssl_client_certificate` na configuração do *proxy* Nginx, com `ssl_verify_depth` definido em 2 para certificar a cadeia de dois níveis (folha → intermediária → raiz).

5.3 CONFIGURAÇÃO DO CANAL E GERAÇÃO DO *GENESIS BLOCK*

A topologia dos canais do MedBlock é definida no ficheiro `configtx-ca.yaml`, que especifica cinco organizações, as respetivas configurações MSP, a definição do *consortium* e dois perfis de canais. Esta secção documenta cada elemento e os procedimentos utilizados para instanciar o canal na rede em funcionamento.

Cada organização especifica o seu identificador MSP, o caminho no sistema de ficheiros para o seu diretório MSP (que contém os certificados de CA, os certificados de administrador e os certificados TLS) e três tipos de políticas. As políticas *Readers*, *Writers* e *Admins*, ao nível da organização, utilizam o tipo de política *Signature*, com regras como `OR('HospitalpublicoMSP.member')` para *Readers* e *Writers*, e `OR('HospitalpublicoMSP.admin')` para *Admins*. Uma política de *Endorsement* é definida separadamente para cada organização, utilizando o tipo *Signature* com o qualificador `.member`. Cada organização de *peers* declara, adicionalmente, um *anchor peer* com o respetivo *hostname* e a respetivo porto. A organização do *orderer* (`OrdererMSP`) segue uma estrutura análoga, mas sem declaração de *anchor peer*.

Dois perfis de canais são definidos, e a Figura 5.3 ilustra a sua relação com as organizações e o *consortium*. O perfil *MedblockChain* define o canal de aplicação `medblockchain`, incluindo as quatro organizações *peer* no grupo *Application* e referenciando o *HospitalConsortium* definido no perfil *HospitalNetworkGenesis*. O *orderer* participa no *System Channel* enquanto os quatro *peers* se ligam ao canal de aplicação.

O perfil `HospitalNetworkGenesis` constrói o *system channel*, incorporando a configuração do *orderer* (tipo Solo, endereço único em `orderer.example.com:7050`) e um *consortium* denominado `HospitalConsortium` que inclui todas as quatro organizações de *peers* (Hospital Público, Hospital Privado, Centro de Saúde e MedBlock). O perfil `MedblockChain` define o canal da aplicação, refere-se ao `HospitalConsortium` e inclui todas as quatro organizações de *peers* no grupo *Application*, com políticas `MAJORITY Admins` e `MAJORITY Endorsement` ao nível da aplicação.

O tipo de *orderer* Solo foi selecionado para o protótipo por não introduzir *overhead* de consenso, sendo adequado para um ambiente de desenvolvimento e avaliação em que a tolerância a falhas bizantinas não é necessária. O Capítulo 4 discutiu esta escolha e notou que um *deployment* de produção dirigido ao SNS exigiria a migração para o protocolo de consenso Raft.

A geração de artefactos do canal é executada pela ferramenta `configtxgen` em dois passos.

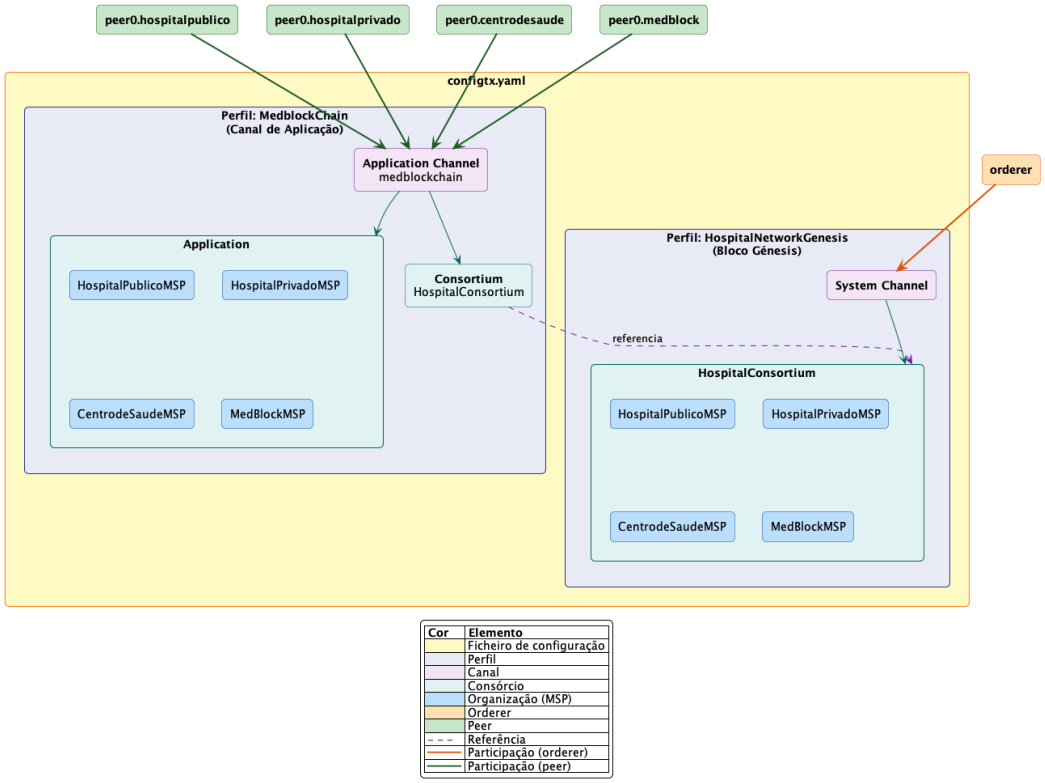


Figura 5.3: Estrutura dos perfis definidos no ficheiro `configtx.yaml` da rede MedBlock.

A primeira produz o *Genesis Block*:

Listagem 1: Geração do bloco génesis com `configtxgen`.

```
configtxgen -profile HospitalNetworkGenesis \
    -channelID system-channel \
    -outputBlock channel-artifacts/genesis.block
```

A segunda gera a transação de criação do canal para o canal `medblockchain`:

Listagem 2: Geração da transação de criação do canal de aplicação `medblockchain`.

```
configtxgen -profile MedblockChain \
    -channelID medblockchain \
    -outputCreateChannelTx channel-artifacts/medblockchain.tx
```

Após o *orderer* iniciar com o *Genesis Block* e todos os *peers* estiverem em funcionamento, as operações de canal são executadas a partir do CLI. A transação de criação do canal é submetida ao *orderer*, resultando no *genesis block* do canal (`medblockchain.block`). Cada um dos quatro *peers* junta-se então ao canal, obtendo e processando esse bloco. As atualizações de *anchor peer* são submetidas

a cada organização de modo a permitir a comunicação *Google Remote Procedure Call* (gRPC) entre organizações. Por fim, verifica-se o funcionamento do canal, consultando a lista de canais de cada *peer* e confirmando que `medblockchain` consta nos resultados.

5.4 IMPLEMENTAÇÃO DE *CHAINCODE*: GESTÃO DE CONSENTIMENTO

O *chaincode* `org-authorizations` é o único *smart contract* implementado na rede MedBlock. O seu âmbito é deliberadamente restrito: gere o mapeamento entre as identidades de pacientes e as organizações de saúde autorizadas a aceder aos seus dados clínicos. Nenhum dado clínico, informação pessoal ou recurso FHIR é armazenado no *ledger* ou passa por ele. Esta separação de metadados de consentimento *on-chain* e de dados clínicos *off-chain* é uma opção arquitetural descrita no Capítulo 4, e o *design* minimalista do *chaincode* é uma consequência direta dessa decisão.

O *chaincode* é implementado em JavaScript, utilizando as bibliotecas `fabric-contract-api`, versão 2.5.0, e `fabric-shim`, versão 2.5.0, conforme declarado nos respetivos ficheiros `package.json`. O ponto de entrada (`index.js`) exporta uma única classe de contrato, `OrgAuthorizationsContract`, que estende a classe base Fabric Contract. A Listagem 3 apresenta uma parte simplificada do código responsável pela implementação completa do contrato. O contrato expõe três funções: `GetAuthorizedOrgs` para consultar as organizações autorizadas por um dado `enrollmentId`, `AddAuthorizedOrg` para conceder acesso a uma organização, e `RemoveAuthorizedOrg` para revogar esse acesso. O código completo poderá ser visto no Apêndice A.1.

Listagem 3: Amostra doódigo-fonte do *smart contract* `OrgAuthorizationsContract`.

```
'use strict';

const { Contract } = require('fabric-contract-api');

class OrgAuthorizationsContract extends Contract {
  _key(enrollmentId) {
    return `orgAuth:${enrollmentId}`;
  }

  async GetAuthorizedOrgs(ctx, enrollmentId) {
    ...
    return existing;
  }

  async AddAuthorizedOrg(ctx, enrollmentId, org) {
```

```
    ...
    return payload;
  }

  async RemoveAuthorizedOrg(ctx, enrollmentId, org) {
    ...
    return payload;
  }
}

module.exports = OrgAuthorizationsContract;
```

O modelo de dados armazena um par chave-valor por paciente no *ledger*. A chave segue um padrão composto: `orgAuth:{enrollmentId}`, em que `enrollmentId` é uma *hash* encriptada derivada do NSNS do paciente. O valor é um objeto *JavaScript Object Notation* (JSON) com dois campos: `enrollmentId` e `orgs` (um *array* de *strings* com as organizações que o paciente autorizou, cada uma com um identificador de MSP, como `HospitalpublicoMSP`). Quando um paciente ainda não concedeu quaisquer autorizações, o *array* `orgs` está vazio.

Três funções de transação, visíveis na Listagem 3, implementam o ciclo de vida completo do consentimento:

`GetAuthorizedOrgs(ctx, enrollmentId)` é uma transação de consulta que recebe o identificador do paciente e retorna o registo atual das autorizações do paciente. Se não houver qualquer registo indicando que o paciente nunca concedeu consentimento, a função retorna um objeto predefinido com um *array* `orgs` vazio, em vez de lançar um erro. Caso o paciente tenha dado autorizações a alguma organização, a transação retorna um *array* com as organizações a que deu autorização.

`AddAuthorizedOrg(ctx, enrollmentId, org)` é uma transação de invocação que recebe o identificador do paciente e a *string* MSP da organização e acrescenta o identificador MSP da organização ao *array* de `orgs` autorizadas pelo paciente. A função primeiro recupera o registo existente (ou inicializa um novo), verifica a existência de entradas duplicadas, acrescenta a nova organização e escreve o registo atualizado de volta ao *ledger*. A verificação de duplicados previne entradas desnecessárias no *ledger* decorrentes de concessões de consentimento repetidas.

`RemoveAuthorizedOrg(ctx, enrollmentId, org)` é uma transação de invocação que recebe o identificador do paciente e a *string* MSP da organização e remove uma organização específica do *array* `orgs`. Filtra o *array* para excluir a organização alvo e escreve o resultado de volta no *ledger*. Se o paciente não tiver registo de autorizações, a função retorna um objeto de autorização vazio, sem erro.

O *deployment* de *chaincode* segue o ciclo de vida do Hyperledger Fabric 2.x, que requer acordo em múltiplos passos entre as organizações do canal antes que um *chaincode* se torne ativo. O código-fonte do *chaincode* é primeiro empacotado num ficheiro `tar.gz` com o comando `peer lifecycle chaincode package`. O pacote é então instalado nos quatro nós *peer* (Hospital Público, Hospital Privado, Centro de Saúde e MedBlock), com cada instalação a retornar um identificador de pacote que inclui um *hash* do conteúdo do *chaincode*. O administrador de cada organização deve então aprovar a definição de *chaincode* no canal, especificando o *package ID*, o nome do *chaincode* (`org-authorizations`), a versão e a política de *endorsement*. Dada a política de *Endorsement* `MAJORITY` configurada no nível da aplicação, pelo menos três das quatro organizações devem aprovar antes que a definição possa ser submetida. Uma vez submetido, o *chaincode* é instanciado e fica disponível para invocação pelo Fabric Gateway.

A política de *endorsement* tem uma implicação prática no fluxo de consentimento: quando um paciente submete uma transação `AddAuthorizedOrg` ou `RemoveAuthorizedOrg` pela aplicação *web*, o Fabric Gateway recolhe *endorsements* de uma maioria dos *peers* do canal antes de submetê-la ao *orderer*. Isto assegura que nenhuma organização pode, unilateralmente, forjar ou revogar consentimento, uma propriedade de segurança alinhada com o requisito do RGPD para a gestão de consentimento verificável e auditável, nos termos do Artigo 7.

5.5 CAMADA DE DADOS CLÍNICOS *OFF-CHAIN*: SERVIDORES FHIR E ARQUITETURA DE *REVERSE PROXY* NGINX

Esta secção descreve a camada de dados clínicos *off-chain*, que materializa a decisão de *design* estabelecida no Capítulo 4 de manter os recursos clínicos fora da *blockchain*.

5.5.1 *Deployment do Servidor HAPI FHIR*

A camada de dados clínicos *off-chain* implementa a decisão de *design* estabelecida no Capítulo 4 de armazenar todos os recursos clínicos fora da *blockchain*, utilizando HL7 FHIR R4 como padrão de interoperabilidade. Três instâncias independentes de servidores HAPI FHIR são implementadas, uma por organização de saúde: `fhir.hospitalpublico.example.com`, `fhir.hospitalprivado.example.com` e `fhir.centrodesaude.example.com`. HAPI é a implementação de referência *open source* da especificação FHIR, mantida pela comunidade HL7 FHIR.

Cada servidor HAPI funciona como um *container* Docker a partir da imagem `hapiproject/hapi:latest`, configurado através de variáveis de ambiente do Spring Boot para se conectar à sua base de dados PostgreSQL 16 dedicada. A configuração de *datasource* especifica a URI *Java Database Connectivity* (JDBC) que aponta para o *container* PostgreSQL (por exemplo, `jdbc:postgresql://postgres.hospitalpublico.example.com:5432/hapi_fhir`). A separação de bases de dados por organização assegura que os dados clínicos de cada prestador de cuidados de saúde permanecem fisicamente isolados, mesmo no ambiente de protótipo.

Quatro tipos de recursos FHIR são utilizados ao longo do protótipo MedBlock. O recurso Patient representa o registo do paciente, identificado pelo NSNS através do sistema de identificadores `http://interop.gov.pt/MDC/Cidadao/NSNS`. O ID do recurso segue a convenção `nsns-{número_NSNS}` (por exemplo, `nsns-111111111`), permitindo consultas determinísticas pelo identificador nacional de saúde. O recurso Appointment regista encontros clínicos agendados, associando a referência ao paciente, o tipo de serviço, as horas de início e fim e a instalação física. O recurso `DiagnosticReport` representa resultados de exames clínicos, associando um paciente, um estado do relatório, um código de tipo de exame e uma data efetiva. O recurso Location identifica os locais físicos em cada organização (por exemplo, `"Hospital Público - Room 201"`).

Um *shell script* (`fhir-mockup-data.sh`) insere dados fictícios nos três servidores FHIR para um único paciente, criando o recurso `Patient` e as entradas `Location`, `Appointments` e `DiagnosticReports` em todas as três organizações. O *script* utiliza `curl` com certificados de cliente mTLS e *headers* de API key para autenticar contra os *proxies* Nginx, replicando exatamente o caminho de segurança que a aplicação MedBlock percorre em produção. Estes dados fictícios simulam um cenário em que os registos clínicos de um paciente estão distribuídos por múltiplos prestadores de cuidados de saúde, o que é o caso de uso descentralizado para o qual o MedBlock foi concebido para responder. Num *deployment* de produção no SNS, estes servidores FHIR seriam substituídos por ou integrados aos sistemas de informação hospitalares existentes, e o MedBlock leria e não escreveria os dados clínicos neles armazenados.

5.5.2 *Configuração do Reverse Proxy Nginx para Endpoints FHIR*

Cada servidor FHIR é precedido por um *reverse proxy* Nginx que impõe dois mecanismos de autenticação independentes antes que qualquer pedido alcance o *backend* HAPI. Três *containers* Nginx: `nginx.hospitalpublico`, `nginx.hospitalprivado`

e `nginx.centrodesaude` partilham um único *template* de configuração (`fhir-proxy.conf.template`), parametrizado com variáveis de ambiente do Docker, para a chave de API específica da organização (`FHIR_API_KEY`) e para o endereço do *backend* (`FHIR_BACKEND`). A Figura 5.4 ilustra as duas camadas de autenticação implementadas no *reverse proxy* Nginx: autenticação por certificados mútuos com verificação do certificado de cliente contra a CA raiz com profundidade 2, e validação do *header* `X-Api-Key` contra a variável de ambiente `$FHIR_API_KEY`. Apenas após ambas as validações o pedido é reencaminhado via `proxy_pass` para o servidor HAPI FHIR interno.

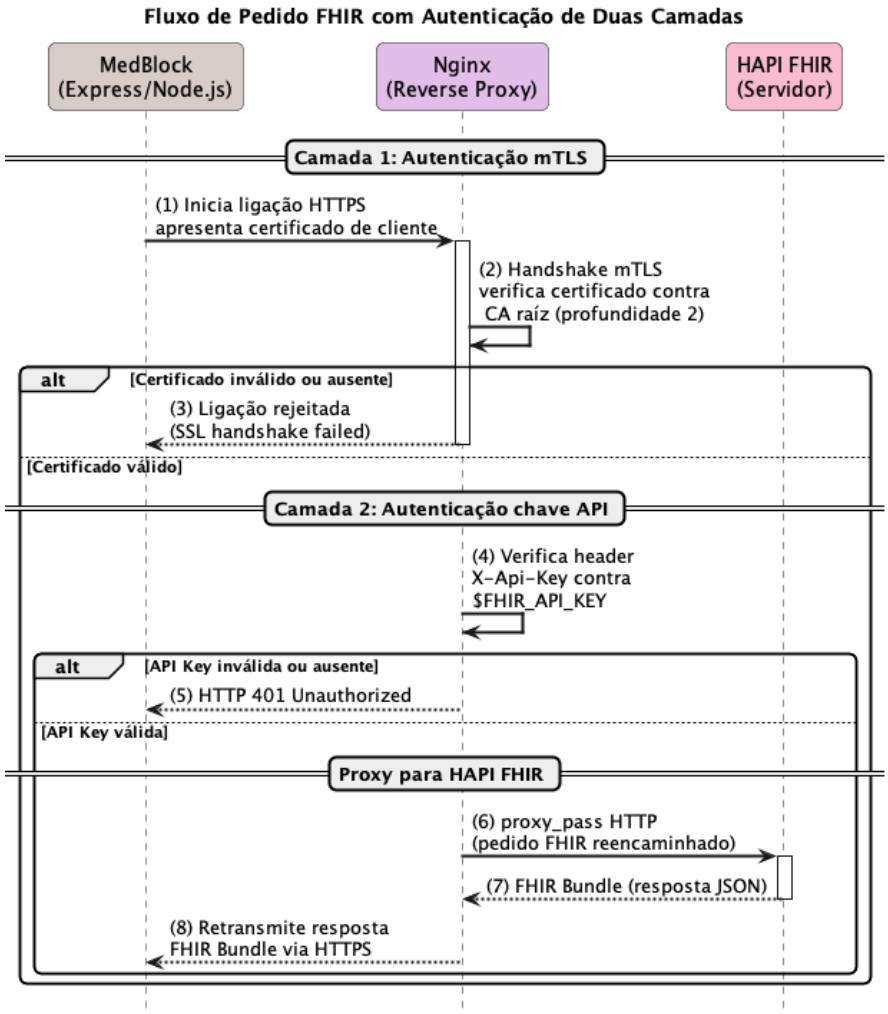


Figura 5.4: Diagrama de sequência do fluxo de um pedido clínico desde a aplicação MedBlock até ao servidor HAPI FHIR.

O *proxy* Nginx escuta no porto 443 com TLS ativado. O certificado do servidor e a chave privada são gerados a partir do material TLS do *peer* da organização correspondente. Estes certificados são assinados pela CA intermediária dessa organização

e encadeiam até à CA raiz, assegurando que os clientes podem verificar a identidade do *proxy* confiando apenas no certificado da CA raiz.

A segunda camada de autenticação valida uma chave de API no nível do HTTP. Uma diretiva do Nginx verifica o *header* do pedido `X-Api-Key` contra a chave por organização injetada via variável de ambiente `FHIR_API_KEY`. Cada organização possui uma chave única. Pedidos com chaves ausentes ou incorretas recebem uma resposta 401 com um corpo de erro em JSON. Esta segunda camada opera independentemente da camada TLS: mesmo um cliente que possua um certificado de rede válido não pode aceder aos recursos FHIR sem a chave de API correta da organização-alvo. O inverso também é verdadeiro: a posse da chave da API, por si só, é insuficiente sem um certificado de cliente válido.

Uma vez que ambas as verificações sejam concluídas, conforme ilustrado na Figura 4.4, o *proxy* reencaminha o pedido ao servidor HAPI FHIR.

A lógica de *defense-in-depth* desta arquitetura é relevante para a avaliação de segurança apresentada no Capítulo 6. O comprometimento de um único tipo de credencial, seja um certificado de cliente roubado ou uma chave de API divulgada, é insuficiente para violar o perímetro de dados do FHIR. Um atacante deve comprometer ambos, de forma independente, para obter acesso, o que eleva substancialmente a dificuldade de um ataque bem-sucedido contra a camada de dados clínicos.

5.6 APLICAÇÃO *WEB*: IMPLEMENTAÇÃO DO *BACKEND*

A aplicação *web* MedBlock é implementada como um servidor Node.js, utilizando a *framework* Express.js, versão 5.1, e TypeScript. A aplicação serve como a camada de integração que liga o sistema de gestão de consentimento *blockchain*, os servidores de dados clínicos FHIR e a infraestrutura de identidade nacional portuguesa (CMD) numa plataforma unificada acessível através de um navegador *web*. O código-fonte do servidor está organizado em um ponto de entrada principal (`server.ts`) e em um conjunto de módulos utilitários sob `src/utils/`.

5.6.1 *Gateway Connection e Integração do Fabric SDK*

A comunicação entre a aplicação *web* e a rede Hyperledger Fabric é estabelecida através do Fabric Gateway SDK (`@hyperledger/fabric-gateway`, versão 1.10.0), que disponibiliza uma API para submeter e avaliar transações num canal do Hyperledger

Fabric. O módulo `fabric-gateway.ts` implementa a função `createGateway()`, que constrói e retorna uma instância de Gateway, bem como o cliente gRPC correspondente.

Um segundo módulo utilitário, `fabric-listOrgs.ts`, permite a descoberta dinâmica das organizações participantes no canal aplicacional. A função `listChannelApplicationOrgs()` consulta o *Configuration System Chaincode* (`cscc`) ao avaliar a transação `GetConfigBlock`, que retorna o bloco de configuração do canal. A função desserializa este bloco utilizando a biblioteca `@hyperledger/fabric-protos`, navegando pela estrutura do *protobuf*: `Block` → `Envelope` → `Payload` → `ConfigEnvelope` → `Config` → `ChannelGroup` → grupo `Application`. As chaves do mapa do grupo `Application` correspondem aos identificadores MSP das organizações participantes no canal. A função filtra a própria organização MedBlock (pois é o operador da plataforma, não um prestador de cuidados de saúde) e retorna os identificadores de organização restantes, ordenados alfabeticamente. Este mecanismo de descoberta dinâmica assegura que a aplicação *web* não codifica listas de organizações de forma estática; se um novo prestador de cuidados de saúde aderir ao canal, aparece automaticamente nas listagens de organizações da aplicação sem alteração de código.

5.6.2 *Autenticação: Integração CMD e Provisionamento de Identidade*

A autenticação de pacientes no MedBlock segue um processo em duas fases. A primeira fase autentica o cidadão pelo sistema de identidade nacional português (CMD); a segunda fase atribui uma identidade de rede Fabric ao cidadão autenticado. A Figura 4.5 ilustra o fluxo de autenticação completo, desde o *redirect* OAuth 2.0 inicial, passando pela obtenção de atributos até ao provisionamento de identidade na Fabric e à criação de sessão.

A integração com a CMD está implementada no *endpoint* `/api/fetch-attributes`. Quando um paciente inicia o *login*, o *frontend* redireciona para o *endpoint* de autorização OAuth 2.0 da Autenticação.Gov. Após a autenticação bem-sucedida, o fluxo OAuth 2.0 redireciona para a página de *callback* do MedBlock, enviando um *token* de acesso. O *frontend* extrai este *token* e faz um `POST` para `/api/fetch-attributes`.

O *endpoint* executa então uma troca em dois passos com a API AttributeManager da AMA, alojada no ambiente de pré-produção (`https://preprod.autenticacao.gov.pt/OAuthResourceServer/Api/AttributeManager`). No primeiro passo, um

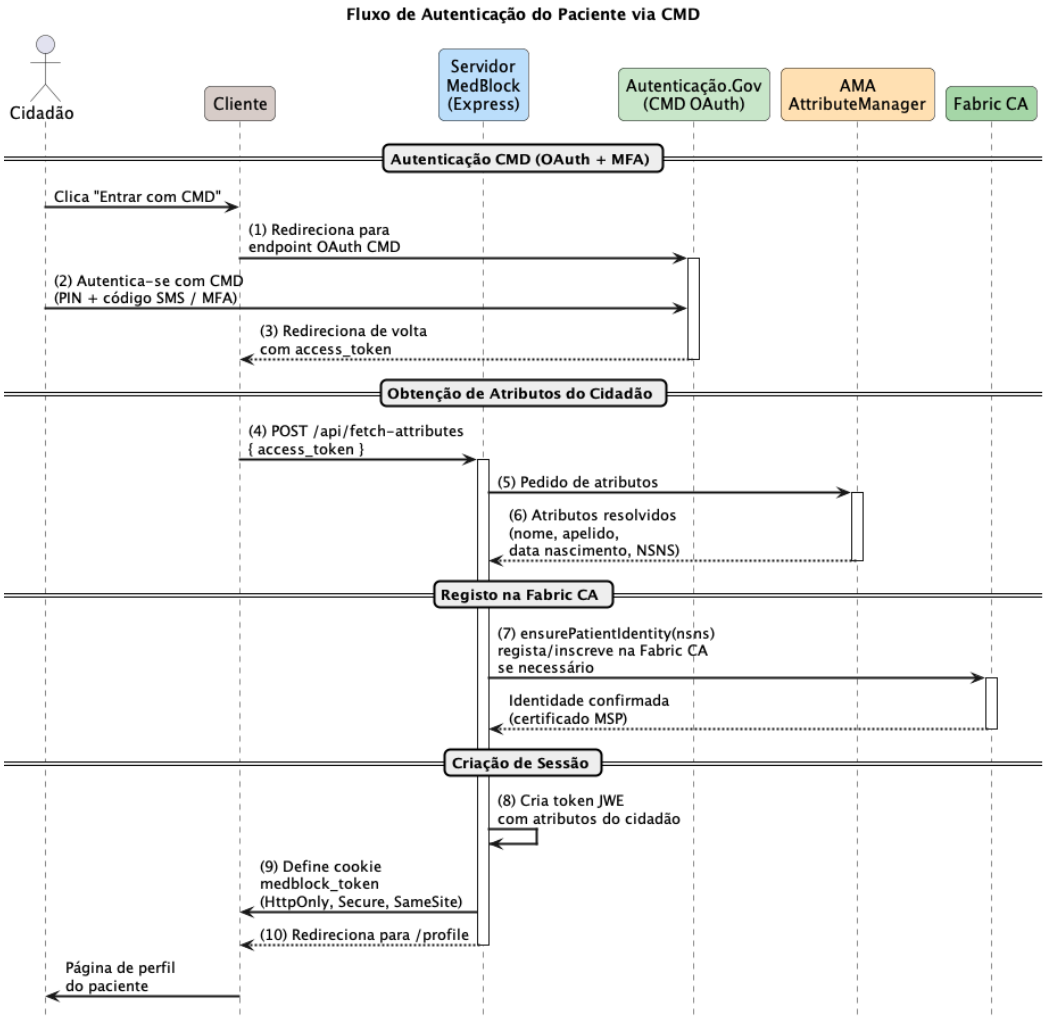


Figura 5.5: Diagrama de sequência do fluxo de autenticação do paciente na plataforma MedBlock.

pedido `POST` envia o *access token* e uma lista de quatro URIs de atributos solicitados: `http://interop.gov.pt/MDC/Cidadao/NomeProprio` (nome próprio), `http://interop.gov.pt/MDC/Cidadao/NomeApelido` (apelido), `http://interop.gov.pt/MDC/Cidadao/DataNascimento` (data de nascimento) e `http://interop.gov.pt/MDC/Cidadao/NSNS` (número serviço nacional de saúde). A API retorna um `authenticationContextId` e inicia a resolução assíncrona dos atributos; o cidadão deverá confirmar a libertação desses atributos no seu telemóvel.

Antes de emitir um *token* de sessão, o *endpoint* invoca a função `ensurePatientIdentity(nsns)` do módulo `fabric-identity.ts`. Esta função gera um `enrollmentId` determinístico a partir do NSNS e verifica se uma identidade correspondente já existe na Hyperledger Fabric CA. Se a identidade não existir, a função regista e inscreve uma nova identidade de cliente na CA intermediária do MedBlock, emitindo um certificado MSP e uma chave privada associados ao NSNS do paciente. Este provisionamento a pedido elimina a necessidade de um passo pré-registo; qualquer cidadão português que se autentique via CMD pode interagir imediatamente com a camada de consentimento da *blockchain*. O `enrollmentId` é derivado deterministicamente a partir do NSNS, assegurando que o mesmo cidadão mapeia sempre para a mesma identidade Fabric entre sessões.

Paralelamente ao fluxo de autenticação via CMD, a aplicação disponibiliza um *endpoint* alternativo de teste acessível em `/login-patient`, cuja interface é apresentada na Figura 5.6. Este *endpoint* solicita ao utilizador a introdução manual dos quatro atributos que o fluxo CMD obtém automaticamente da infraestrutura da AMA: nome próprio, apelido, data de nascimento e número de saúde (NSNS). Após a submissão, o *endpoint* `/api/login-patient` processa estes valores de forma idêntica ao percurso da CMD, invocando `ensurePatientIdentity()` para provisionar a identidade Fabric, gerando um *JSON Web Encryption* (JWE) com os atributos do cidadão e estabelecendo a sessão autenticada recorrendo à *cookie* `medblock_token`.

A existência deste *endpoint* justifica-se por uma limitação operacional do ambiente de pré-produção da Autenticação.Gov: o acesso ao serviço OAuth 2.0 da CMD em pré-produção está restrito a um conjunto limitado de utilizadores de teste registados junto da AMA, não estando disponível para utilizadores arbitrários. Durante o desenvolvimento e a avaliação do protótipo, esta restrição impediria a demonstração e o teste iterativo. O *endpoint* `/login-patient` contorna esta limitação ao permitir a simulação de autenticações com dados fictícios, preservando todo o percurso de processamento desde o provisionamento de identidade no Fabric CA até à verificação de consentimento via *chaincode* e à recuperação de dados clínicos dos servidores

FHIR. Este *endpoint* destina-se exclusivamente ao ambiente de desenvolvimento. Num cenário de produção no SNS, o *endpoint* `/login-patient` deveria ser removido.

O fluxo de *login* da organização segue um caminho diferente. O *endpoint* `/api/login-organization` recebe um identificador MSP de organização do *frontend*, verifica se a organização existe no canal (chamando `listChannelApplicationOrgs()`) e emite um *token* de sessão com `OrganizationAttributes`, contendo o `organizationMSPId` e o campo `role` definido como `"organization"`. Não é necessária autenticação via CMD no portal da organização, uma vez que as organizações se autenticam ao nível institucional, e não ao nível individual. Este *endpoint* foi criado apenas no contexto de desenvolvimento, pois foi necessário simular a interação das organizações de saúde com a *blockchain*, e um processo de autenticação mais robusto deverá ser implementado para uma eventual migração para um sistema de produção.

5.6.3 *Gestão de Sessões: Implementação de Tokens JWE*

O estado de sessão no MedBlock é gerido por *tokens* cifrados armazenados em *cookies* HTTP, implementados com o módulo utilitário `jwe.ts` e a biblioteca `jose`. A escolha de JWE em detrimento do mais comum *JSON Web Token* (JWT) apenas com assinatura foi motivada por um requisito de segurança específico: o *payload* do *token* contém atributos do cidadão (nome, NSNS, data de nascimento) que constituem dados pessoais nos termos do Artigo 4(1) do RGPD. A cifragem do *token* em repouso assegura que estes atributos não podem ser lidos ao inspecionar o valor do *cookie*, mesmo que o *cookie* seja intercetado ou extraído do armazenamento do navegador.

A função `createEncryptedToken()` aceita um objeto `CitizenAttributes` ou `OrganizationAttributes`, serializa-o como *payload* de JWE e retorna uma *string* de *token* cifrada. A função `decryptToken()` inverte este processo, retornando os atributos originais ou *null* caso a descifragem falhe (indicando adulteração ou expiração). O *middleware* `verifyEncryptedToken` é registado em todas as rotas protegidas: extrai o *token* do *cookie* `medblock_token`, decifra-o e anexa os atributos resultantes ao objeto de pedido como `req.user`. Se o *cookie* estiver ausente, expirado ou não puder ser decifrado, o *middleware* retorna uma resposta 401.

A `requireRole()` proporciona controlo de acesso baseado em *roles*. Aceita uma *string* (`"patient"` ou `"organization"`) e retorna uma função de *middleware* que verifica se `req.user.role` corresponde ao valor requerido. Esta cadeia de dois

middlewares, primeiro `verifyEncryptedToken`, depois `requireRole()`, é aplicada a todos os *endpoints* protegidos, assegurando que a autenticação e a autorização são impostas de forma consistente. O *cookie* é configurado com `httpOnly: true` (impedindo o acesso por JavaScript), `secure: true` (requerendo *Hypertext Transfer Protocol Secure* (HTTPS)), `sameSite: "strict"` (prevenindo *cross-site request forgery*) e um `maxAge` de 2 horas, após o qual a sessão expira e o paciente deve reautenticar-se.

5.6.4 *Camada de Obtenção de Dados FHIR*

O módulo `fhir.ts` encapsula toda a comunicação entre a aplicação MedBlock e os servidores FHIR *off-chain*. A sua responsabilidade principal é traduzir as decisões de consentimento registadas na *blockchain* e obter os dados das organizações de saúde com autorização do paciente.

O módulo implementa um agente HTTPS com mTLS. Na primeira invocação, lê três variáveis de ambiente: `FHIR_CLIENT_CERT_PATH`, `FHIR_CLIENT_KEY_PATH` e `FHIR_CA_CERT_PATH` e constrói um `https.Agent` com o certificado do cliente, a chave privada e o certificado de CA da rede Fabric. Este agente é reutilizado em todos os pedidos FHIR subsequentes, estabelecendo ligações mTLS com os *proxies* Nginx que precedem cada servidor FHIR. Uma função auxiliar, `fhirRequestConfig(org)`, constrói a configuração de pedido Axios para uma dada organização, anexando tanto o agente mTLS como o *header* `X-Api-Key` apropriado, obtido a partir do módulo `organization-config.ts`, que mapeia cada identificador MSP para o respetivo URL FHIR e chave de API.

Três funções específicas de obtenção por recurso tratam dos principais tipos de dados. `getPatientByNSNS(nsns, org)` constrói um URL de recurso FHIR `Patient` na convenção `nsns-{NSNS}` e retorna o nome próprio, o apelido, a data de nascimento e o número de saúde do paciente. `getAppointmentsByPatient(nsns, org)` consulta o *endpoint* `Appointment` com um filtro de referência ao paciente e, em seguida, iterando sobre as entradas do FHIR, extrai a data, o estado, o tipo de serviço e a localização de cada consulta. `getExamsByPatient(nsns, org)` consulta o *endpoint* `DiagnosticReport` de forma semelhante, extraindo a data efetiva, estado, tipo de exame e organização.

Duas funções de agregação, `getAppointmentsFromAllOrgs(nsns, authorizedOrgs)` e `getExamsFromAllOrgs(nsns, authorizedOrgs)`, implementam a obtenção de dados entre organizações autorizadas pelo paciente. Estas funções aceitam

um *array* de identificadores MSP de organizações autorizadas (obtidos da *blockchain* via `GetAuthorizedOrgs`) e distribuem pedidos em paralelo ao servidor FHIR de cada organização autorizada, utilizando `Promise.allSettled()`. A utilização de `allSettled` em vez de `Promise.all` é uma decisão deliberada de resiliência: se o servidor FHIR de uma organização estiver indisponível, os resultados restantes são ainda assim recolhidos e devolvidos ao utilizador, em vez de todo o pedido falhar. Os resultados de todas as organizações são fundidos num único *array*, proporcionando ao paciente ou ao profissional de saúde uma visão unificada dos dados clínicos entre prestadores.

5.6.5 *Design da REST API e Estrutura de Rotas*

O servidor Express expõe uma *Representational State Transfer* (REST) API que mapeia de forma clara para dois papéis de utilizador: paciente e organização. Todos os *endpoints* seguem um padrão de segurança consistente, com rotas protegidas pela cadeia de *middleware* `verifyEncryptedToken` e `requireRole()`. A Tabela 5.2 resume a superfície completa da API.

A API orientada ao paciente inclui: `GET /api/profile` (retorna os atributos do paciente autenticado a partir do *token* de sessão), `GET /api/authorizations` (avalia a transação de *chaincode* `GetAuthorizedOrgs` para o NSNS do paciente atual), `POST /api/authorizations` (submete uma transação `AddAuthorizedOrg` com a organização alvo a partir do corpo do pedido), `DELETE /api/authorizations/:org` (submete uma transação `RemoveAuthorizedOrg`), `GET /api/fhir/patient/:nsns` (obtém o registo do paciente do primeiro servidor FHIR autorizado), `GET /api/fhir/appointments/:nsns` (agrega consultas de todas as organizações autorizadas) e `GET /api/fhir/exams/:nsns` (agrega relatórios de diagnóstico de todas as organizações autorizadas). Cada *endpoint* de obtenção FHIR consulta primeiro a *blockchain* para as organizações autorizadas do paciente e, em seguida, distribui pedidos FHIR apenas a essas organizações, impondo o perímetro de consentimento em cada acesso aos dados.

A API orientada à organização inclui: `POST /api/login-organization` (autentica a organização e emite um *token* de sessão), `GET /api/orgs` (retorna a lista de todas as organizações no canal, descobertas dinamicamente a partir da configuração do canal) e `POST /api/org-dashboard/lookup` (executa uma pesquisa de paciente na perspetiva da organização). O *endpoint* de *lookup* extrai o identificador MSP da organização requerente a partir do *token* de sessão, consulta a *blockchain* para

Tabela 5.2: Resumo dos *endpoints* da interface aplicacional *REST* da plataforma MedBlock.

Método	**Caminho**	**Descrição**
Autenticação		
POST	/api/fetch-attributes	Obtém atributos do cidadão via AMA após autenticação pela Chave Móvel Digital
Endpoints do Paciente		
GET	/api/profile	Retorna perfil do paciente a partir do *token* JWE
GET	/api/authorizations	Lista organizações autorizadas pelo paciente
POST	/api/authorizations	Concede acesso a uma organização
DELETE	/api/authorizations/:org	Revoga acesso de uma organização
GET	/api/fhir/patient/:nsns	Obtém dados do paciente via FHIR
GET	/api/fhir/appointments/:nsns	Lista consultas do paciente via FHIR
GET	/api/fhir/exams/:nsns	Lista exames do paciente via FHIR
Endpoints da Organização		
POST	/api/login-organization	Autenticação da organização com credenciais
GET	/api/orgs	Lista organizações disponíveis na rede
POST	/api/org-dashboard/lookup	Pesquisa dados de paciente por número nacional de saúde (vista organização)
Rotas de Páginas Estáticas		
GET	/	Página inicial
GET	/profile	Página de perfil
GET	/appointments	Página de consultas
GET	/exams	Página de exames
GET	/authorizations	Página de autorizações
GET	/org-dashboard	Painel da organização
GET	/logout	Termina sessão e limpa *cookie*

obter as organizações autorizadas do paciente e verifica se a organização requerente consta da lista autorizada antes de prosseguir com a obtenção de dados FHIR. Se a organização não estiver autorizada, uma resposta 403 é retornada. Esta verificação implementa a verificação de consentimento na camada aplicacional, complementando o papel da *blockchain* como o *ledger* autoritativo de consentimento.

As rotas de páginas estáticas servem o *frontend* *HyperText Markup Language* (HTML) para cada portal: o caminho raiz (`/`) serve a página de *login* (com *redirect* para `/profile` se o utilizador já possuir uma sessão válida), `/profile`, `/appointments`, `/exams` e `/authorizations` servem as páginas do portal do paciente (todas protegidas para o papel de paciente), e `/org-dashboard` serve o portal da organização (protegido para o papel de organização). A rota `/logout` limpa o *cookie* de sessão e redireciona para o caminho raiz.

5.7 APLICAÇÃO *WEB*: IMPLEMENTAÇÃO DO *FRONTEND*

O *frontend* do MedBlock é implementado como um conjunto de páginas HTML estáticas, cada uma acompanhada de ficheiros *Cascading Style Sheets* (CSS) dedicados e JavaScript, servidos a partir do diretório `public/`.

A aplicação apresenta dois portais distintos, diferenciados pelo papel do utilizador.

O portal do paciente compreende quatro páginas. A página de *login* (`login-patient.html`), apresentada na Figura 5.6, disponibiliza campos de entrada para nome próprio, apelido, data de nascimento e NSNS.

Figura 5.6: Página de *login* de testes do paciente.

Existe um caminho de *login* separado para autenticação baseada em CMD via Autenticação.Gov, com a página *callback* (`callback.html`) a tratar do *redirect* do OAuth 2.0 mostrado na Figura 5.7. O JavaScript de *callback* extrai o `access_token` e faz um `POST` para `/api/fetch-attributes`, redirecionando para `/profile` em caso de sucesso.

Figura 5.7: Página de autenticação via Chave Móvel Digital no portal do paciente.

A página de perfil (`profile.html`), apresentada na Figura 5.8, exibe as informações do paciente autenticado (nome próprio, apelido, data de nascimento e número nacional de saúde), obtidas de `/api/profile`.

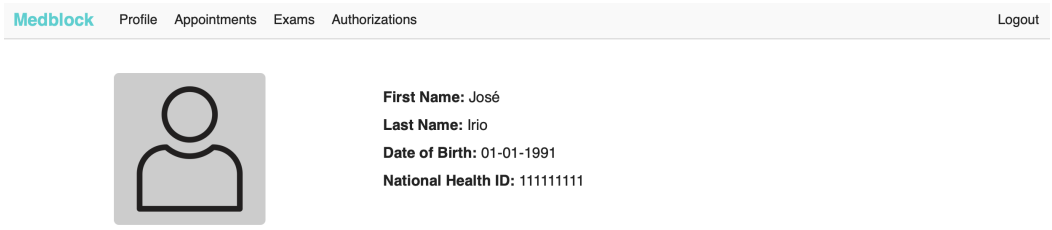


Figura 5.8: Página de perfil do paciente no portal da plataforma MedBlock.

A página de consultas (`appointments.html`) renderiza uma tabela de consultas clínicas agregadas de todas as organizações autorizadas, com colunas para data, estado, tipo de serviço e localização, apresentado na Figura 5.9.

A página de exames (`exams.html`) utiliza o mesmo *layout* que a página de consultas, apresentado na Figura 5.10.

A página de autorizações (`authorizations.html`), apresentada na Figura 5.11, é a interface de gestão de consentimento: exibe um *dropdown* de organizações disponíveis (povoado dinamicamente a partir de `/api/orgs`), um botão "adicio-

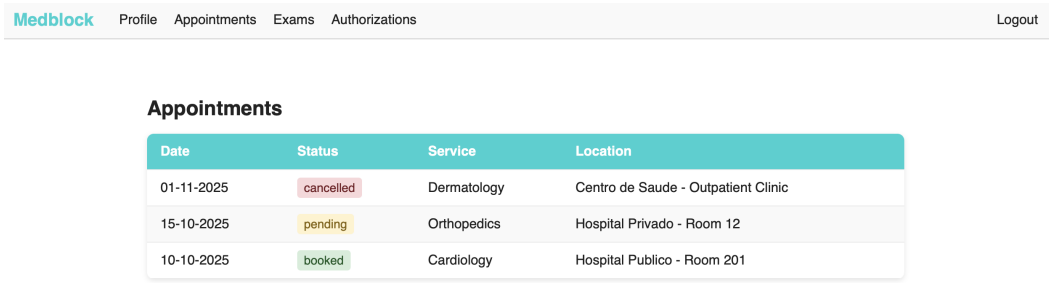


Figura 5.9: Página de consultas clínicas do paciente no portal da plataforma MedBlock.

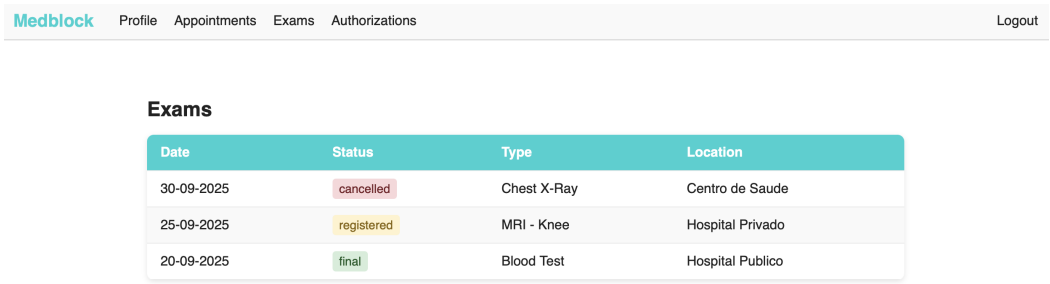


Figura 5.10: Página de exames do paciente no portal da plataforma MedBlock.

nar"para conceder acesso e uma lista de organizações atualmente autorizadas, cada uma com um botão de remoção para revogar acesso. As ações de adição e remoção invocam `POST /api/authorizations` e `DELETE /api/authorizations/:org`, respetivamente, acionando as transações de *chaincode* correspondentes.

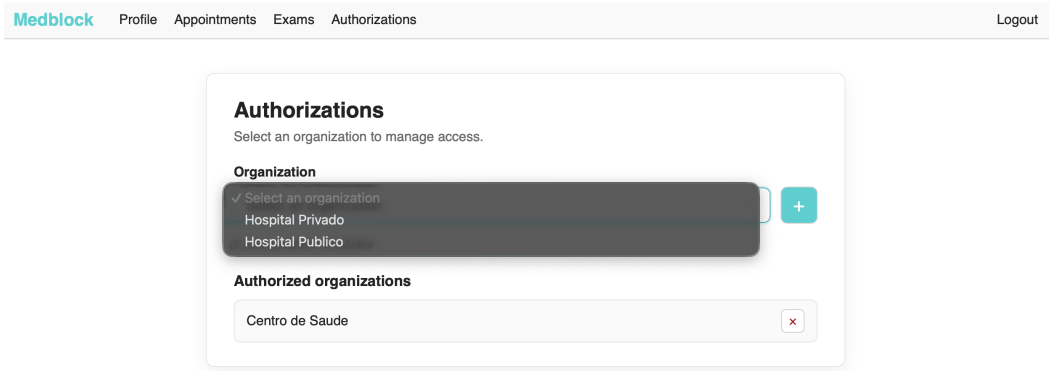


Figura 5.11: Página de gestão de autorizações do paciente no portal da plataforma MedBlock.

O portal da organização disponibiliza um *dashboard* numa única página (`org-dashboard.html`). A página de *login* (`login-organizations.html`), apresentada na Figura 5.12, exibe um *dropdown* de organizações disponíveis, preenchido com os resultados da requisição `/api/orgs`.

Após o *login*, o utilizador da organização é direcionado para o *dashboard*, que apresenta uma interface de pesquisa de pacientes. O profissional de saúde envia

Figura 5.12: Página de *login* da organização no portal da plataforma MedBlock.

um NSNS e o *frontend* faz um `POST` para `/api/org-dashboard/lookup`. Se a organização estiver autorizada para o paciente, o *dashboard* exibe as informações do paciente, consultas e exames numa interface com separadores, conforme apresentado na Figura 5.13.

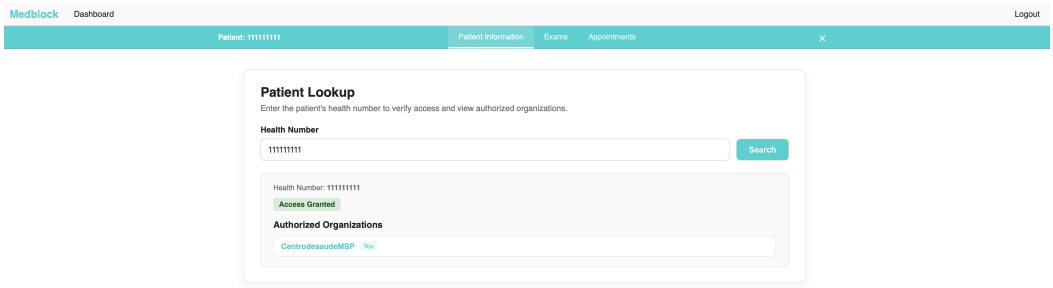


Figura 5.13: Página de pesquisa de paciente por número nacional de saúde no portal da organização, com acesso concedido.

Se a organização não estiver autorizada, é exibida a mensagem "Acesso Negado", apresentada na Figura 5.14. Este fluxo demonstra diretamente o mecanismo de imposição de consentimento: o profissional de saúde não pode visualizar quaisquer dados clínicos, a menos que o paciente tenha explicitamente concedido autorização à sua organização na *blockchain*.

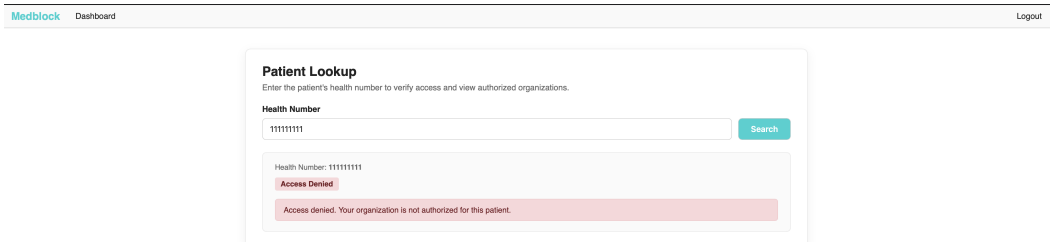


Figura 5.14: Página de pesquisa de paciente por número nacional de saúde no portal da organização, com acesso negado.

Ambos os portais partilham uma barra de navegação consistente exibindo a marca MedBlock, ligações específicas do portal e uma ação de *logout*. O *design* visual prioriza a clareza e o minimalismo funcional em detrimento da elaboração estética, consistente com o papel do protótipo como demonstrador técnico e não como produto finalizado.

5.8 DETALHES DE IMPLEMENTAÇÃO DE SEGURANÇA

Esta secção consolida os controlos de segurança concretos implementados na plataforma MedBlock. Enquanto o Capítulo 4 descreveu a estratégia de segurança em camadas ao nível arquitetural, esta secção documenta cada controlo implementado, fornecendo a especificidade necessária para as avaliações do *Cybersecurity Framework* do NIST e do *OWASP Web Security Testing Guide* no Capítulo 5.

5.8.1 *Controlos de Segurança na Camada Aplicacional*

O servidor Express.js utiliza a biblioteca de *middleware* Helmet.js para definir cabeçalhos HTTP de segurança em cada resposta. A *Content Security Policy* (CSP) está configurada com `script-src` restrito a `'self'`, impedindo a execução de *scripts inline* ou de *scripts* carregados de domínios externos, uma mitigação direta contra ataques de *Cross-Site Scripting* (XSS). A diretiva `connect-src` permite ligações a `'self'` e `https://medblock.pt:8080`, sendo esta última necessária para o fluxo OAuth 2.0 da CMD. Todas as restantes configurações predefinidas do Helmet estão ativadas, incluindo `X-Content-Type-Options: nosniff`, `X-Frame-Options` e `Referrer-Policy`.

Os *cookies* de sessão estão configurados com três *flags* de proteção. A *flag* `httpOnly` impede que JavaScript do lado do cliente aceda ao *cookie*, mitigando o roubo de sessão via XSS. A *flag* `secure` assegura que o *cookie* só é transmitido através de ligações HTTPS. A *flag* `sameSite: "strict"` impede o navegador de enviar o *cookie* em pedidos *cross-origin*, proporcionando proteção robusta contra ataques de *Cross-Site Request Forgery* (CSRF). A combinação de *payloads* de *token* cifrados com JWE e estas *flags* de *cookie* cria uma abordagem *defense-in-depth* para a segurança de sessão: mesmo que um atacante intercete o *cookie*, o *payload* cifrado não pode ser lido nem modificado sem a chave de cifragem do servidor.

A validação de entrada é aplicada no limite da API antes de qualquer interação com a rede Fabric ou com os servidores FHIR. O campo NSNS é validado por uma expressão regular, rejeitando qualquer valor que não corresponda exatamente a 9 dígitos. Os identificadores de organização são validados como *strings* não vazias e verificados contra a configuração do canal antes de serem utilizados. Estas verificações impedem que entradas malformadas alcancem o *chaincode* ou sejam injetadas nos URLs de consulta FHIR.

5.8.2 *Segurança de Perímetro: Integração Cloudflare WAF*

A aplicação *web* MedBlock está implementada com proteção da WAF da Cloudflare no plano gratuito, com o domínio `medblock.pt` configurado para *proxying* DNS completo na rede da Cloudflare. Esta decisão adiciona uma camada de defesa de perímetro que opera independentemente de todos os controlos de segurança ao nível aplicacional e ao nível de *blockchain*. A Tabela 5.3 resume a configuração completa da Cloudflare.

A configuração de DNS encaminha todo o tráfego público para a Cloudflare, definindo os registos A para `medblock.pt` e `www.medblock.pt` com o endereço IP do servidor de origem com o estado de *proxy* definido como `"Proxied"`. Esta configuração mascara o verdadeiro endereço IP do servidor de origem das consultas DNS públicas, prevenindo ataques diretos à origem que contornariam completamente a WAF. O modo de configuração do DNS é `Full`, o que significa que a Cloudflare resolve todas as consultas do domínio.

A configuração TLS está definida para o modo de cifragem `Full`, o que significa que o tráfego é cifrado tanto entre o navegador e a *edge* da Cloudflare e entre a Cloudflare e o servidor de origem. A configuração `"Always Use HTTPS"` está ativada, fazendo com que a Cloudflare emita um *redirect* 301 para qualquer pedido HTTP em texto claro. O *HTTP Strict Transport Security* (HSTS) está ativado com um `Max-Age` de seis meses, com inclusão de subdomínios e *preload*; isto instrui navegadores compatíveis a recusarem qualquer futura ligação em texto claro ao domínio, mesmo que o utilizador escreva explicitamente `http://` na barra de endereço. A versão TLS mínima está definida como TLS 1.2, rejeitando ligações de clientes que apenas suportem os protocolos TLS 1.0 ou 1.1 obsoletos. Estas configurações da camada de transporte estão alinhadas com as melhores práticas atuais para aplicações *web* e com a recomendação do NIST de evitar versões do TLS inferiores a 1.2.

Tabela 5.3: Resumo das configurações de segurança aplicadas na plataforma MedBlock através da firewall de aplicações Web da Cloudflare no plano gratuito.

Funcionalidade	**Configuração**	**Fundamentação**
Modo SSL/TLS	Full	Cifra tráfego entre Cliente-Cloudflare e Cloudflare-Servidor
Always Use HTTPS	Ativado	Redireciona todo o tráfego HTTP para HTTPS
HSTS	Ativado, Max-Age 6 meses, incluir subdomínios, *preload*	Força futuras ligações do navegador a utilizarem exclusivamente HTTPS
Versão TLS mínima	TLS 1.2	Rejeita clientes TLS 1.0/1.1 obsoletos e vulneráveis
Regra personalizada 1 *"Portugal"*	Bloquear se país não for PT	Restrição geográfica a endereços IP portugueses
Regra personalizada 2 *"Block specified user agents"*	Bloquear ∼50 assinaturas de ferramentas	Impede reconhecimento automatizado por *scanners* e ferramentas de vulnerabilidade
Rate Limiting	5 req/10s por IP em caminhos `/auth` + `/api`, bloquear por 10s	Mitiga ataques de força bruta e abuso de API
Proteção DDoS	3 conjuntos de regras geridas sempre ativos	Mitiga DDoS nas camadas SSL/TLS, rede e HTTP
Esquema da API	23 *endpoints* abrangidos	Visibilidade de *endpoints* e deteção de anomalias

Duas regras de *firewall* personalizadas estão configuradas. A primeira regra, denominada `"Portugal"`, bloqueia todos os pedidos recebidos cujo código de país não seja `PT`. Esta restrição geográfica limita o acesso a endereços IP portugueses, em conformidade com o público-alvo do MedBlock no SNS e com o âmbito do protótipo como plataforma de saúde portuguesa. A segunda regra, denominada `"Block specified user agents"`, bloqueia pedidos cujo *header* `User-Agent` corresponda a qualquer uma das aproximadamente cinquenta assinaturas conhecidas de ferramentas de segurança ofensiva e de reconhecimento. As assinaturas bloqueadas abrangem quatro categorias: *scanners* de vulnerabilidades e *frameworks* de exploração (`sqlmap`, `nikto`, `burpsuite`, `nessus`, `openvas`, etc.); e *web crawlers*, bibliotecas de *scraping* e *bots* comerciais não essenciais (`scrapy`, `python-requests`, etc.). Esta lista de bloqueio abrangente aborda o reconhecimento automatizado no perímetro, impedindo que estas ferramentas mapeiem os *endpoints* da aplicação, descubram caminhos ocultos ou sondem vulnerabilidades.

Uma regra de *rate limiting*, denominada `"Rate Limiting Auth & API Endpoints"`, visa os caminhos mais sensíveis da aplicação, aqueles que tratam de autenticação e de acesso a dados. A regra corresponde a pedidos para os seguintes caminhos de URI: `/api/fetch-attributes`, `/api/login-patient`, `/api/authorizations`, `/api/login-organization` e `/api/org-dashboard/lookup`, bem como para qualquer caminho que comece por `/api/fhir/`. Quando um único endereço IP excede cinco pedidos numa janela de dez segundos para qualquer destes *endpoints*, a Cloudflare bloqueia todos os pedidos subsequentes correspondentes a esse IP durante dez segundos. O limiar de cinco pedidos por dez segundos foi calibrado para estar bem acima da taxa de interação humana legítima, permanecendo, ao mesmo tempo, bem abaixo das taxas de pedidos geradas por ataques automatizados de força bruta, por ferramentas ou por *scripts* de reconhecimento.

A camada de proteção *Distributed Denial of Service* (DDoS) compreende três conjuntos de regras geridas pela Cloudflare, sempre ativos: proteção contra ataques DDoS via SSL/TLS, proteção contra ataques DDoS na camada de rede e proteção contra ataques DDoS via HTTP. Todos os três conjuntos de regras operam com configurações predefinidas.

O plano gratuito impõe limitações que devem ser reconhecidas com transparência. O *OWASP Core Rule Set* e outros conjuntos de regras WAF avançados geridos estão disponíveis apenas em planos pagos (Pro e superiores). Consequentemente, a defesa de perímetro do MedBlock assenta nas regras personalizadas descritas acima e nas proteções de base da Cloudflare. Para um sistema de produção, a atualização para um plano pago com regras WAF geridas proporcionaria uma camada adicional

de proteção contra ataques e ameaças descritas no OWASP Top 10. Não obstante, o plano gratuito proporciona uma defesa significativa para um protótipo: restrição geográfica, bloqueio de assinaturas de ferramentas, *rate limiting* em *endpoints* sensíveis e mitigação de DDoS sempre ativa reduzem, coletivamente e de forma substancial, a superfície de ataque. A Figura 5.15 captura, na sua totalidade, esta arquitetura de segurança em camadas.

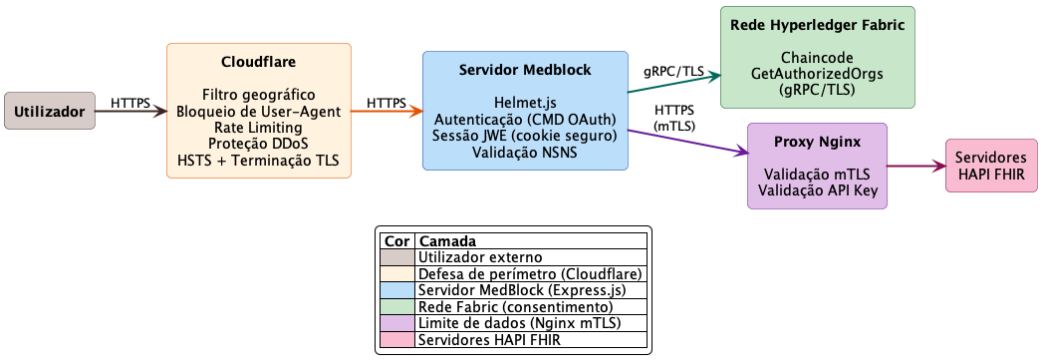


Figura 5.15: Diagrama de segurança em camadas ilustrando o perímetro de segurança completo da plataforma MedBlock.

5.9 RESUMO DO CAPÍTULO

Este capítulo documentou a implementação completa do protótipo MedBlock, abrangendo a infraestrutura de rede contentorizada, a hierarquia de PKI, o canal e o *chaincode* do *blockchain*, a camada de dados FHIR *off-chain*, a aplicação *web* e a segurança de perímetro. O ambiente implementado compreende vinte e um *containers* Docker operando em uma rede, implementando uma rede Hyperledger Fabric 2.5 com quatro organizações e uma PKI de duas camadas, três servidores HAPI FHIR R4, precedidos por *reverse proxies* Nginx, com imposição de mTLS, e uma aplicação *web* TypeScript/Express.js integrando autenticação nacional via CMD, gestão de sessões cifrada com JWE e verificação de consentimento mediada por *blockchain*. A Cloudflare WAF proporciona defesa de perímetro com restrição geográfica, bloqueio de ferramentas de reconhecimento, *rate limiting* de *endpoints* de API e mitigação de DDoS.

O fluxo de dados através do sistema completo pode ser resumido da seguinte forma: um paciente autentica-se via CMD, o que provisiona uma identidade na rede Hyperledger Fabric; o paciente concede ou revoga acesso organizacional através de transações de *chaincode* registadas no *ledger* imutável; quando o paciente ou um profissional de saúde autorizado solicita dados clínicos, a aplicação consulta a

blockchain para o estado de consentimento atual e distribui então pedidos FHIR, seguros por mTLS e validação da chave de API, apenas para as organizações autorizadas, agregando os resultados numa visão unificada. Em nenhum momento os dados clínicos tocam o *ledger blockchain*, e em nenhum momento podem ser obtidos sem um registo de consentimento correspondente *on-chain*.

A avaliação desta implementação, incluindo *benchmarking* de desempenho com o Hyperledger Caliper e avaliação de segurança conforme o *Cybersecurity Framework* do NIST e o *OWASP Web Security Testing Guide*, é apresentada no Capítulo 6.

6

TESTES E RESULTADOS

Este capítulo apresenta os resultados da avaliação da plataforma MedBlock. Abrange as duas estratégias de avaliação: desempenho e segurança. A avaliação de desempenho quantifica o *throughput* e a latência da rede Hyperledger Fabric sob cinco níveis progressivos de carga. Foi utilizado o Hyperledger Caliper como instrumento de medição (Hyperledger Foundation, 2026). A avaliação de segurança, mostra resultados de três ferramentas: análise estática de código com o SonarQube (SonarSource, 2024), análise dinâmica de vulnerabilidades *web* com o OWASP ZAP (Checkmarx, 2024) e análise de vulnerabilidades de infraestrutura com o Tenable Nessus Professional (Tenable, Inc., 2024).

Em conformidade com o princípio de testes com prioridade para o *staging*, todas as atividades de *benchmarking* foram executadas num canal dedicado (`calipertest`), isolado do canal de produção `medblockchain`. Este isolamento garante que as transações de teste, que modificam o *world state* com dados sintéticos, não comprometem a integridade dos dados do canal de produção. Os testes de segurança, por sua vez, foram realizados na aplicação e servidor *web*.

6.1 AVALIAÇÃO DE DESEMPENHO: *BENCHMARK* HYPERLEDGER CALIPER

Esta secção apresenta os resultados do *benchmarking* de desempenho conduzido sobre a rede Hyperledger Fabric do MedBlock.

6.1.1 *Configuração do Ambiente de Teste*

O *benchmarking* foi realizado com o Hyperledger Caliper, uma *framework* de avaliação de desempenho desenvolvido pela comunidade Hyperledger. A *framework* foi configurada para interagir com a rede Hyperledger Fabric 2.5 do MedBlock por meio do *peer* da organização `HospitalpublicoMSP`. O perfil de ligação especifica a comunicação via gRPC com o *peer* `peer0.hospitalpublico.example.com` e com o *orderer* `orderer.example.com`. Os *timeouts* de *endorsement* e de *ordering* foram

configurados com 300 segundos. Este valor é deliberadamente elevado para evitar falhas artificiais em níveis de carga extremos.

O *chaincode* avaliado, `org-authorizations`, é o mesmo *smart contract* utilizado em produção para gerir o consentimento dos pacientes, com três operações: `AddAuthorizedOrg` (concessão de acesso), `RemoveAuthorizedOrg` (revogação de acesso) e `GetAuthorizedOrgs` (consulta de autorizações). As duas primeiras são operações de invocação que percorrem o *pipeline* completo de *endorsement*, *ordering* e confirmação, enquanto a última é uma operação de consulta resolvida localmente pelo *peer* sem modificar o *ledger*.

A estratégia de avaliação baseou-se em cinco níveis de carga progressivamente mais exigentes. O nível 1 (Base) estabelece uma linha de referência com 25 TPS de escrita e 50 TPS de leitura, utilizando 5 *workers* e um total de 2 000 transações. Os níveis 2 a 4 duplicam iterativamente a pressão sobre a rede, aumentando proporcionalmente a taxa de envio, o número de *workers* e o volume de transações, com o objetivo de identificar o ponto em que a latência começa a degradar-se e o *orderer* atinge a saturação. O nível 5 (Saturação) impõe uma taxa-alvo de 500 TPS para escrita e 1 000 TPS para leitura, com 25 *workers* e 20 000 transações, visando determinar o teto absoluto de débito e verificar se a rede apresenta falhas sob carga extrema. Todos os níveis utilizam o controlador de taxa `fixed-rate` do Caliper, que submete as transações a uma cadência constante. A Tabela 6.1 sintetiza a configuração dos cinco níveis.

Tabela 6.1: Configuração dos cinco níveis de carga utilizados no *benchmark* Hyperledger Caliper.

Nível	**Designação**	**TPS Escrita** (alvo)	**TPS Leitura** (alvo)	**Workers**	**Transações**
1	Base	25	50	5	2 000
2	Moderado	50	100	10	4 000
3	Elevado	100	200	15	8 000
4	Stress	200	400	20	16 000
5	Saturação	500	1 000	25	20 000

6.1.2 *Resultados de Throughput*

A Tabela 6.2 consolida as métricas de desempenho obtidas ao longo dos cinco níveis de carga, e a Figura 6.1 apresenta a evolução do débito (*throughput*) por operação.

Tabela 6.2: Resultados do *benchmark* Hyperledger Caliper ao longo de cinco níveis de carga.

Nível	**AddAuthorizedOrg**			**GetAuthorizedOrgs**		**RemoveAuthorizedOrg**			**Falhas**
	TPS	Méd. (s)	Máx. (s)	TPS	Méd. (s)	TPS	Méd. (s)	Máx. (s)	
1 – Base	25,1	0,25	1,16	50,2	0,01	25,1	0,21	0,37	0
2 – Moderado	49,6	1,18	3,52	100,3	0,02	50,0	0,29	0,86	0
3 – Elevado	49,7	2,57	8,14	158,9	0,11	55,0	2,54	10,64	0
4 – Stress	61,3	3,80	13,34	263,0	0,12	63,9	2,94	8,27	0
5 – Saturação	54,5	3,27	13,33	228,8	0,18	56,4	3,37	11,18	0

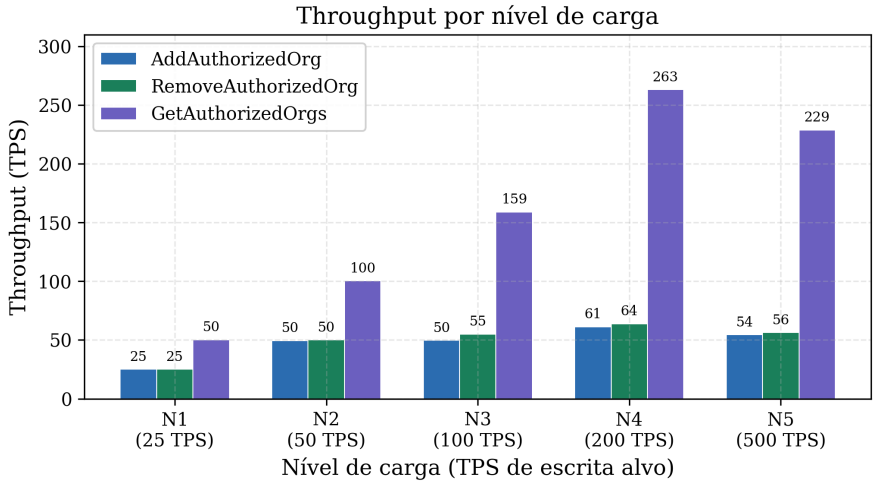


Figura 6.1: Throughput obtido ao longo de cinco níveis progressivos de carga para o *chaincode* `org-authorizations` na rede Hyperledger Fabric 2.5 do MedBlock.

No nível 1, as três operações atingem o débito-alvo com precisão: `AddAuthorizedOrg` regista 25,1 TPS, `GetAuthorizedOrgs` atinge 50,2 TPS e `RemoveAuthorizedOrg` atinge 25,1 TPS. A rede comporta-se de forma previsível, processando todas as transações sem acumulação de atrasos. No nível 2 (Moderado), o débito de escrita duplica efetivamente para aproximadamente 50 TPS (`AddAuthorizedOrg`: 49,6 TPS; `RemoveAuthorizedOrg`: 50,0 TPS), e o de leitura acompanha proporcionalmente (100,3 TPS). A rede continua a absorver a carga imposta sem sinais de congestionamento.

A partir do nível 3 (Elevado), observa-se uma diferença significativa entre o débito-alvo e o débito efetivo. Embora a taxa de envio configurada seja de 100 TPS para escrita, o débito observado estabiliza em 49,7 TPS para `AddAuthorizedOrg` e 55,0 TPS para `RemoveAuthorizedOrg`. Isso indica que a rede atingiu seu limite de processamento de escrita. As operações de leitura, por não dependerem do *pipeline* de *ordering*, continuam a escalar e chegam a 158,9 TPS. Esse padrão fica mais evidente nos níveis 4 e 5. O débito de escrita oscila entre 50 e 64 TPS, independentemente da taxa de escrita-alvo (200 ou 500 TPS). Já as leituras atingem um pico de 263,0 TPS no nível 4, antes de recuar para 228,8 TPS no nível 5.

A ligeira redução no débito de leitura entre os níveis 4 e 5 pode ser atribuída à contenção de recursos no *peer*: com 25 *workers* concorrentes a submeter simultaneamente operações de escrita e leitura, o *peer* partilha ciclos de CPU entre a validação de blocos provenientes do *orderer*. Ainda assim, mesmo sob saturação, o débito de leitura mantém-se acima de 228 TPS.

O resultado mais expressivo é a ausência total de falhas em todos os cinco níveis. As 50 000 transações submetidas ao longo dos testes foram processadas com uma taxa de sucesso de 100%. Isso mostra que o *chaincode* `org-authorizations` e a configuração da rede Hyperledger Fabric não introduzem erros, mesmo quando a taxa de submissão excede substancialmente a capacidade de processamento do *orderer*.

6.1.3 *Resultados de Latência*

A Figura 6.2 apresenta a evolução da latência média e máxima ao longo dos cinco níveis de carga.

As operações de leitura apresentam comportamento estável. A latência média de `GetAuthorizedOrgs` mantém-se entre 0,01 s no nível 1 e 0,18 s no nível de Saturação,

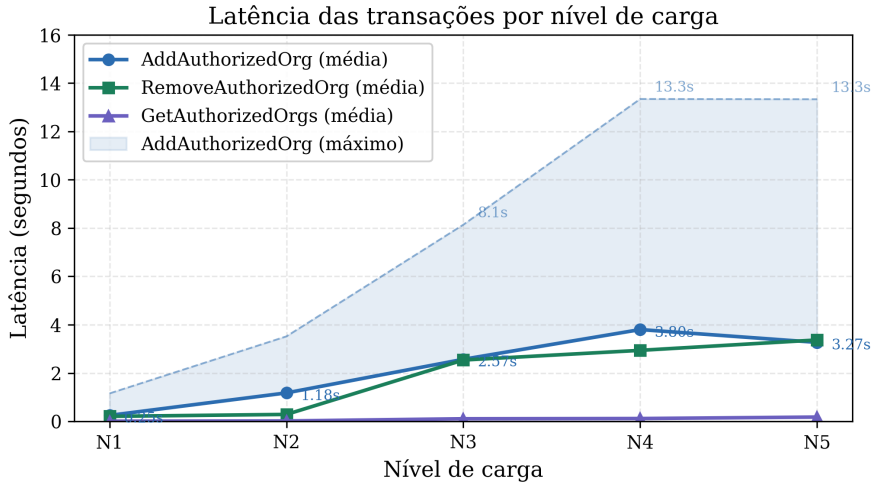


Figura 6.2: Latência média e máxima das transações ao longo de cinco níveis de carga.

uma variação de apenas 170 milissegundos ao longo de um aumento de carga. Este resultado é expectável: as consultas são resolvidas a partir do *world state* local do *peer*, sem envolver o *orderer* nem o protocolo de consenso.

As operações de escrita apresentam um perfil distinto. No nível 1, a latência média de `AddAuthorizedOrg` é de 0,25 s e a de `RemoveAuthorizedOrg` é de 0,21 s, valores compatíveis com o tempo de um ciclo completo de *endorsement*-*ordering*-confirmação numa rede com carga reduzida. No nível 2, a latência média de `AddAuthorizedOrg` quadruplica para 1,18 s e a máxima atinge 3,52 s, indicando o início da acumulação de transações na fila do *orderer*.

No nível 3 (Elevado), a latência média de escrita salta para 2,57 s (`AddAuthorizedOrg`) e 2,54 s (`RemoveAuthorizedOrg`), enquanto a latência máxima de `AddAuthorizedOrg` escala para 8,14 s e a de `RemoveAuthorizedOrg` para 10,64 s. A partir deste ponto, a taxa de submissão de transações excede consistentemente a capacidade de *ordering*, e as transações acumulam-se em filas que aumentam o tempo de espera de forma não linear. Nos níveis 4 e 5, a latência máxima de escrita estabiliza-se em cerca de 13 segundos.

A assimetria entre a latência de leitura e de escrita confirma a análise arquitetónica delineada: o *bottleneck* de desempenho da rede reside exclusivamente no *pipeline* de *ordering* e validação, não na camada de consulta.

6.1.4 *Discussão e Contextualização dos Resultados*

Os resultados do *benchmark* permitem extrair três conclusões operacionais para um possível cenário de uso do MedBlock no SNS.

Em primeiro lugar, o teto de escrita de aproximadamente 50 a 64 TPS é adequado ao padrão de utilização previsto. As operações de escrita no MedBlock correspondem exclusivamente a ações de concessão e revogação de acesso organizacional, atos deliberados que um paciente realiza pontualmente ao longo da sua interação com o sistema de saúde. Mesmo num cenário hipotético de adoção massiva, em que milhares de pacientes ajustassem simultaneamente as suas autorizações, a taxa de operações de consentimento por segundo permaneceria ordens de grandeza abaixo do teto observado. A título de comparação, Oki et al. reportaram restrições de escalabilidade no protótipo do Hospital Provincial Frere sob cargas elevadas de transações, mas não quantificaram os limites observados por meio de *benchmarking* padronizado (Oki et al., 2024). A presente avaliação colmata essa lacuna ao estabelecer empiricamente os limites de desempenho com uma metodologia reprodutível.

Em segundo lugar, o débito de leitura de até 263 TPS proporciona uma margem confortável para o fluxo operacional dominante: profissionais de saúde e pacientes que consultam o estado de consentimento. Estas consultas constituem a operação mais frequente no sistema e beneficiam da resolução local no *peer*, sem necessidade de consenso.

Em terceiro lugar, a taxa de sucesso de 100% em todos os níveis, inclusive o de saturação, demonstra resiliência sob pressão. A rede não rejeita transações nem gera erros quando sobrecarregada; em vez disso, aumenta a latência, preservando a integridade transacional. Este comportamento é particularmente relevante num contexto de saúde, em que a perda de uma transação de consentimento poderia ter implicações para a continuidade do acesso a dados clínicos.

Importa, contudo, reconhecer as limitações desta avaliação. O *benchmark* foi executado num ambiente com um único *peer* (`peer0.hospitalpublico.example.com`), sem carga concorrente proveniente das restantes três organizações da rede. Num cenário de produção com múltiplos *peers* a participar no *endorsement*, o comportamento de latência poderia variar, particularmente se a política de *endorsement* exigir assinaturas de várias organizações. Adicionalmente, os testes utilizaram um *orderer* Solo, que não reproduz o comportamento de um serviço de *ordering* Raft com tolerância a falhas. Estas condições representam o cenário mais favorável e os

resultados devem ser interpretados como limites superiores de desempenho para a configuração testada.

6.2 AVALIAÇÃO DE SEGURANÇA

A avaliação de segurança da plataforma MedBlock foi conduzida com recurso a três ferramentas complementares, cada uma orientada para uma camada ou perspetiva distinta do sistema. A análise estática de código, realizada com o SonarQube, avalia a qualidade e a postura de segurança do código-fonte antes da execução. A análise dinâmica com o OWASP ZAP examina o comportamento da aplicação *web* em tempo de execução, ao simular interações maliciosas. A análise com o Tenable Nessus Professional examina a infraestrutura do servidor e da aplicação *web* ao nível da rede, identificando vulnerabilidades em serviços, protocolos e *software* instalado.

6.2.1 *Análise Estática de Código: SonarQube*

A análise estática foi realizada com o SonarQube Community Edition na versão v26.2.0.119303, aplicada a dois projetos distintos: a aplicação *web* MedBlock (projeto `medblock`) e o *smart contract* de gestão de autorizações (projeto `org-authorizations`).

O projeto `medblock` obteve a classificação *Passed*. Na dimensão de segurança, o SonarQube não identificou nenhuma vulnerabilidade nem *security hotspot*, atribuindo a classificação A (0 *issues*). Na dimensão de manutenção, a classificação foi igualmente A, embora com 54 *issues* abertos, predominantemente relacionados com convenções de código e à sua complexidade. A percentagem de duplicações registada foi de 4,1% em 3 800 linhas analisadas. Na dimensão de fiabilidade, o projeto obteve a classificação D com 8 *issues* abertos. Estes *issues* de fiabilidade correspondem a padrões de código que o SonarQube classifica como potencialmente problemáticos em termos de comportamento em *runtime* e constituem uma limitação reconhecida que não afeta diretamente a postura de segurança do sistema, mas que poderá ser avaliado em iterações futuras do desenvolvimento.

O projeto `org-authorizations`, correspondente ao *chaincode*, obteve a classificação *Passed* com resultados A em todas as três dimensões: segurança (0 *issues*), fiabilidade (0 *issues*) e manutenção (0 *issues*). Não foram identificados *security hotspots* e a percentagem de duplicações é de 0,0%. A dimensão reduzida do *chaincode*

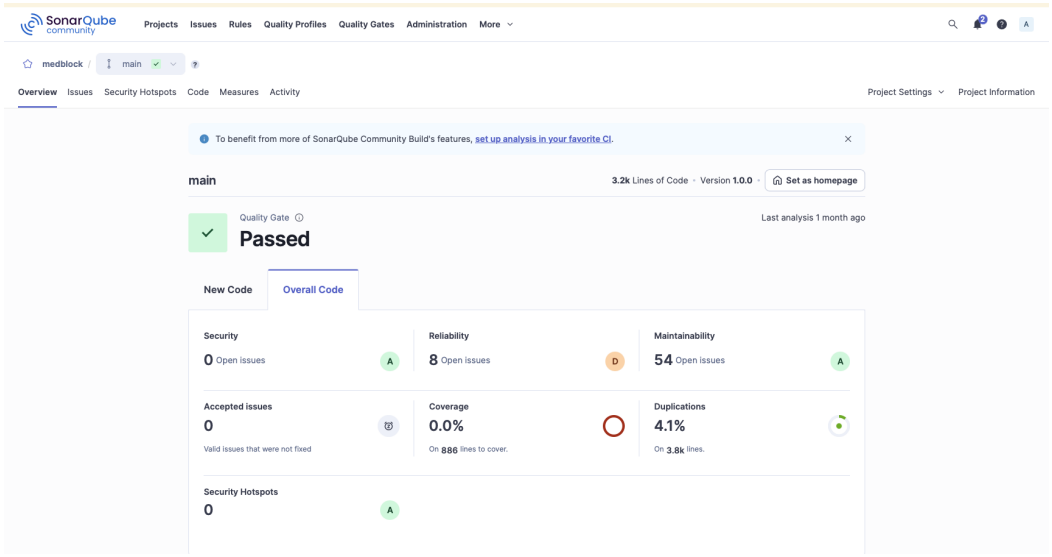


Figura 6.3: Resultados da análise estática SonarQube do projeto `medblock`, correspondente à aplicação *web*.

é uma consequência deliberada da decisão arquitetónica de limitar o *smart contract* exclusivamente às operações de consentimento (concessão, revogação e consulta), excluindo qualquer lógica adicional. Esta simplicidade reduz a superfície de ataque do código que opera diretamente sobre o *ledger* imutável.

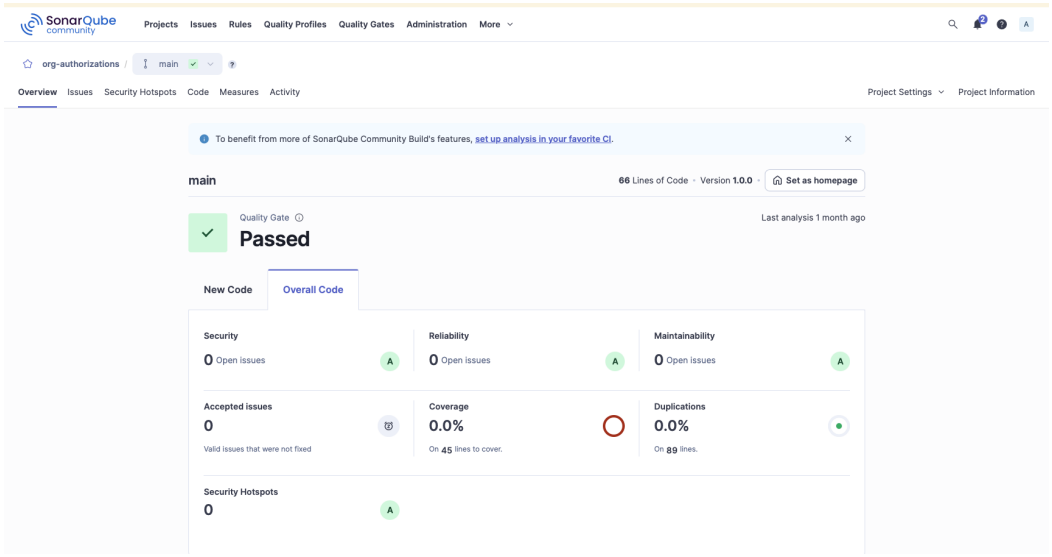
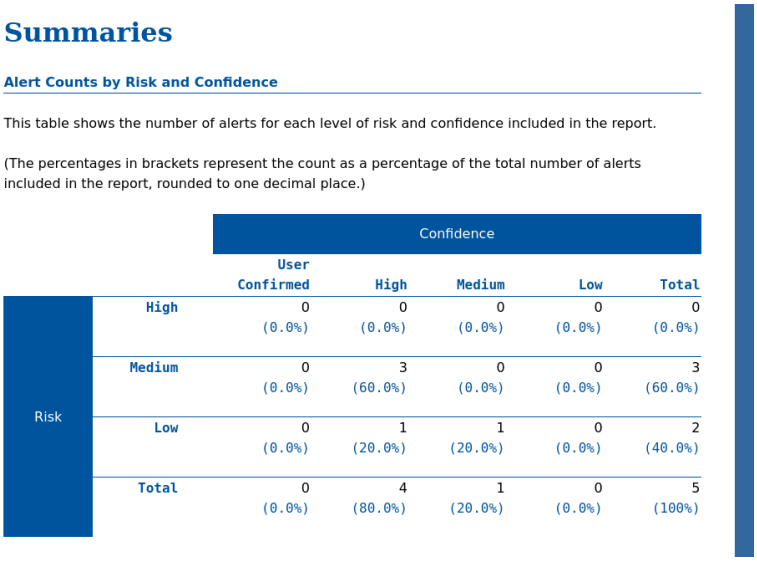


Figura 6.4: Resultados da análise estática SonarQube do projeto `org-authorizations`, correspondente ao *chaincode*.

6.2.2 *Análise Dinâmica: OWASP ZAP*

A análise dinâmica da aplicação *web* foi realizada com o OWASP ZAP na versão 2.17.0, que executou uma análise automatizada no domínio `medblock.pt`. O relatório gerado pela ferramenta, apresentado na Figura 6.5, identificou um total de 5 alertas, distribuídos em 3 de risco médio e 2 de risco baixo, sem alertas de risco elevado.



Summaries

Alert Counts by Risk and Confidence

This table shows the number of alerts for each level of risk and confidence included in the report.

(The percentages in brackets represent the count as a percentage of the total number of alerts included in the report, rounded to one decimal place.)

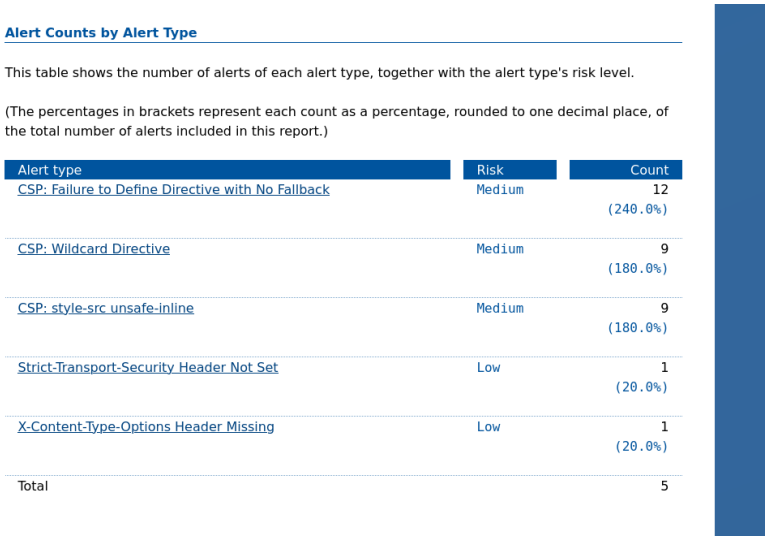
		Confidence				
		User Confirmed	High	Medium	Low	Total
Risk	High	0 (0.0%)	0 (0.0%)	0 (0.0%)	0 (0.0%)	0 (0.0%)
	Medium	0 (0.0%)	3 (60.0%)	0 (0.0%)	0 (0.0%)	3 (60.0%)
	Low	0 (0.0%)	1 (20.0%)	1 (20.0%)	0 (0.0%)	2 (40.0%)
	Total	0 (0.0%)	4 (80.0%)	1 (20.0%)	0 (0.0%)	5 (100%)

Figura 6.5: Matriz de alertas por nível de risco e confiança resultante do *scan* automatizado OWASP ZAP à aplicação *web* MedBlock.

Os três alertas de risco médio estavam relacionados à configuração da CSP. O primeiro, *CSP: Failure to Define Directive with No Fallback*, foi registado em 12 ocorrências e sinaliza diretivas CSP que não definem um valor de *fallback*, resultando na herança do valor `default-src`. O segundo, *CSP: Wildcard Directive*, foi detetado em 9 ocorrências e identifica diretivas que utilizam padrões excessivamente permissivos. O terceiro, *CSP: style-src unsafe-inline*, igualmente com 9 ocorrências, alerta para a permissão de estilos *inline* na política CSP.

Os dois alertas de risco baixo referiam-se à ausência de cabeçalhos de segurança HTTP: *Strict-Transport-Security Header Not Set* (1 ocorrência) e *X-Content-Type-Options Header Missing* (1 ocorrência). Estes alertas indicavam que determinados *endpoints* da aplicação não incluíam os cabeçalhos HSTS e `X-Content-Type-Options` nas respostas HTTP.

Nenhum alerta de risco elevado foi identificado, o que significa que o OWASP ZAP não detetou vulnerabilidades de injeção *Structured Query Language* (SQL), *cross-site scripting*, nem falhas de autenticação ou autorização exploráveis por meio



Alert Counts by Alert Type

This table shows the number of alerts of each alert type, together with the alert type's risk level.

(The percentages in brackets represent each count as a percentage, rounded to one decimal place, of the total number of alerts included in this report.)

Alert type	Risk	Count
CSP: Failure to Define Directive with No Fallback	Medium	12 (240.0%)
CSP: Wildcard Directive	Medium	9 (180.0%)
CSP: style-src unsafe-inline	Medium	9 (180.0%)
Strict-Transport-Security Header Not Set	Low	1 (20.0%)
X-Content-Type-Options Header Missing	Low	1 (20.0%)
Total		5

Figura 6.6: Distribuição dos alertas por risco e número de ocorrências resultante do *scan* automatizado OWASP ZAP à aplicação *web* MedBlock.

da análise automatizada. A ausência de alertas nas categorias de risco mais elevadas é consistente com os controlos implementados na camada aplicacional.

Após a remediação dos alertas identificados pelo OWASP ZAP, a configuração dos cabeçalhos de segurança HTTP foi verificada com a ferramenta shcheck, *open source* disponível no GitHub (Lammerts, 2023), concebida para auditar a presença desses cabeçalhos em respostas HTTP. A verificação foi realizada no domínio `https://www.medblock.pt`.

O resultado, apresentado na Figura 6.7, confirmou a presença de 10 cabeçalhos de segurança e a ausência de 0, o que corresponde a uma pontuação perfeita segundo os critérios da ferramenta.

```
[*] Analyzing headers of https://www.medblock.pt
[*] Effective URL: https://www.medblock.pt
[*] Header X-XSS-Protection is present! (Value: 0)
[*] Header X-Content-Type-Options is present! (Value: nosniff)
[*] Header Strict-Transport-Security is present! (Value: max-age=15552000; includeSubDomains; preload)
[*] Header Content-Security-Policy is present!
Value:
        default-src: 'self'
        base-uri: 'self'
        object-src: 'none'
        frame-ancestors: 'self'
        form-action: 'self' https://autenticacao.gov.pt
        script-src: 'self'
        script-src-attr: 'none'
        style-src: 'self'
        style-src-attr: 'none'
        img-src: 'self' data:
        font-src: 'self' https: data:
        connect-src: 'self' https://medblock.pt:8080
        upgrade-insecure-requests
[*] Header X-Permitted-Cross-Domain-Policies is present! (Value: none)
[*] Header Referrer-Policy is present! (Value: no-referrer)
[*] Header Permissions-Policy is present! (Value: camera=(), microphone=(), geolocation=(), payment=(), usb=(), magnetometer=(), gyroscope=(), accelerometer=())
[*] Header Cross-Origin-Embedder-Policy is present! (Value: credentialless)
[*] Header Cross-Origin-Resource-Policy is present! (Value: same-origin)
[*] Header Cross-Origin-Opener-Policy is present! (Value: same-origin)
-------------------------------------------------------------
[!] Analyzing headers for https://www.medblock.pt
[+] 10 security header(s) present
[-] 0 security header(s) missing
```

Figura 6.7: Resultado da verificação de cabeçalhos de segurança HTTP com a ferramenta shcheck

A presença da totalidade destes cabeçalhos confirma que os controlos implementados pelo *middleware* Helmet.js, estão efetivamente a ser aplicados a todas as respostas HTTP servidas pela aplicação. A diretiva `form-action` inclui explicitamente `https://autenticacao.gov.pt` para permitir o redirecionamento do formulário de autenticação para o serviço nacional Autenticação.Gov, e a diretiva `connect-src` inclui `https://medblock.pt:8080` para suportar o fluxo OAuth 2.0 da CMD. Estas exceções são necessárias para a integração com a infraestrutura de identidade nacional e não comprometem a postura global de segurança da política CSP, uma vez que se limitam a domínios controlados e confiáveis.

6.2.3 *Análise de Vulnerabilidades: Nessus Professional*

A análise de vulnerabilidades de infraestrutura foi conduzida com o Tenable Nessus Professional na versão 10.11.3 (edição *trial*), utilizando a política *Advanced Scan*, com severidade baseada no CVSS v3.0. Foram realizados dois *scans* distintos, um direcionado à aplicação *web* (Medblock - *webapp*) e outro ao servidor (Medblock - *server*), seguidos de *scans* de verificação pós-remediação. Os resultados foram filtrados, utilizando o filtro representado na Figura 6.8, para excluir entradas de severidade `None` (informacionais), concentrando a análise exclusivamente em vulnerabilidades classificadas como *Low*, *Medium*, *High* ou *Critical*.

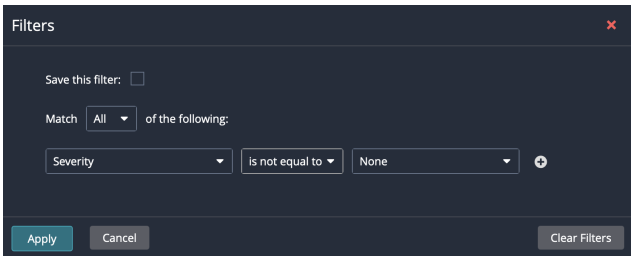


Figura 6.8: Filtro aplicado aos resultados dos *scans* Tenable Nessus Professional.

O *scan* inicial da aplicação *web* identificou 2 vulnerabilidades, como pode ser visto na Figura 6.9, ambas de severidade média (CVSS 6.5). A primeira, *TLS Version 1.0 Protocol Detection*, detetou que o servidor aceitava ligações com o protocolo TLS 1.0, obsoleto desde 2020 devido a vulnerabilidades conhecidas nos mecanismos criptográficos. A segunda, *TLS Version 1.1 Deprecated Protocol*, identificou o mesmo problema no TLS 1.1, também obsoleto. Cada uma destas vulnerabilidades foi registada em 6 ocorrências, correspondentes aos diferentes serviços expostos pelo servidor que aceitavam negociações com versões antigas do protocolo TLS.

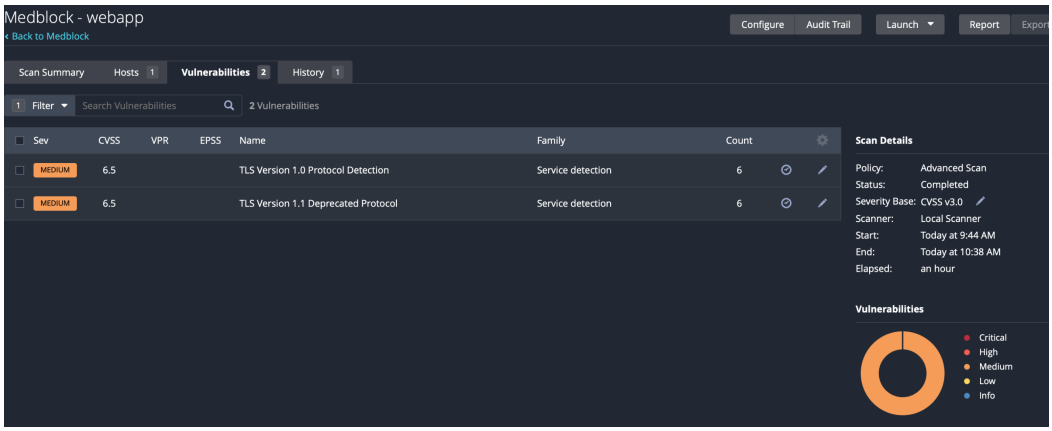


Figura 6.9: Resultado do *scan* Tenable Nessus Professional inicial à aplicação *web* MedBlock

O *scan* inicial ao servidor identificou 7 vulnerabilidades, como pode ser visto na Figura 6.10, com uma distribuição de severidade mais ampla: 2 de risco elevado (CVSS 7.8), 4 de risco médio (CVSS entre 4.4 e 6.6) e 1 de risco baixo (CVSS 2.2). Todas as vulnerabilidades estavam associadas ao editor de texto Vim instalado no servidor. As duas vulnerabilidades de risco elevado correspondiam a uma falha de injeção de comandos e a um *heap-based buffer overflow*, ambas exploráveis apenas localmente por um utilizador com acesso ao sistema.

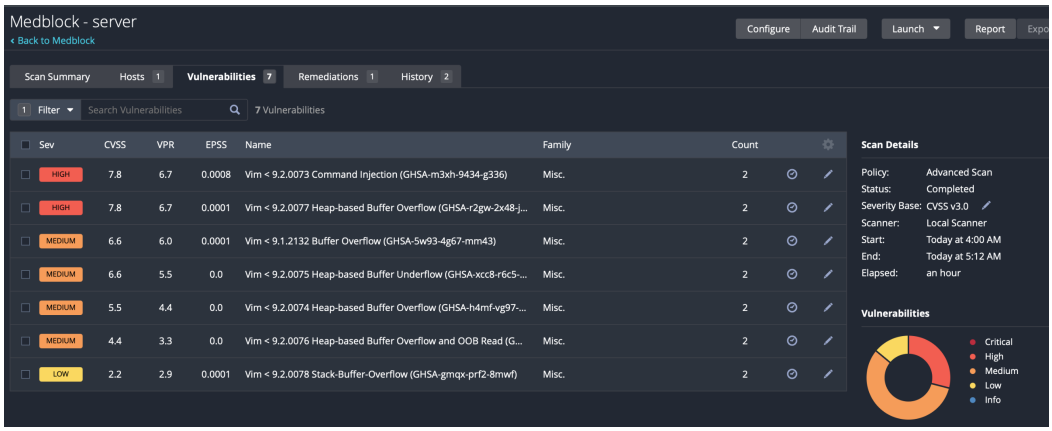


Figura 6.10: Resultado do *scan* Tenable Nessus Professional inicial ao servidor.

O processo de remediação envolveu duas ações: a desativação dos protocolos TLS 1.0 e 1.1 na configuração do servidor e na WAF de Perímetro, restringindo a negociação de TLS exclusivamente às versões 1.2 e 1.3, e a atualização do Vim para uma versão que corrige todas as CVEs identificadas. Ambas as ações foram implementadas e verificadas por meio de *scans* de reavaliação.

Os *scans* pós-remediação, representados nas Figuras 6.11 e 6.12, confirmaram a resolução de todas as vulnerabilidades identificadas. O *scan* da aplicação *web* (Medblock - *webapp* - PostFix) registou 0 vulnerabilidades, e o *scan* do servidor

(Medblock - *server* - PostFix) registou igualmente 0 vulnerabilidades. Ambos os *scans* pós-remediação foram concluídos com sucesso utilizando a mesma política *Advanced Scan* e o mesmo filtro de severidade, garantindo a comparabilidade direta com os resultados iniciais.

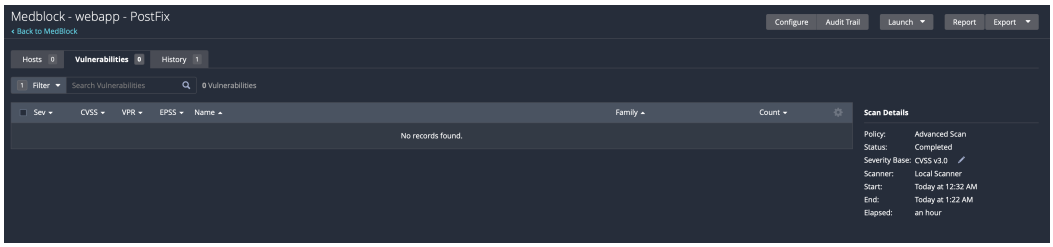


Figura 6.11: Resultado do *scan* Tenable Nessus Professional pós-remediação à aplicação *web* MedBlock

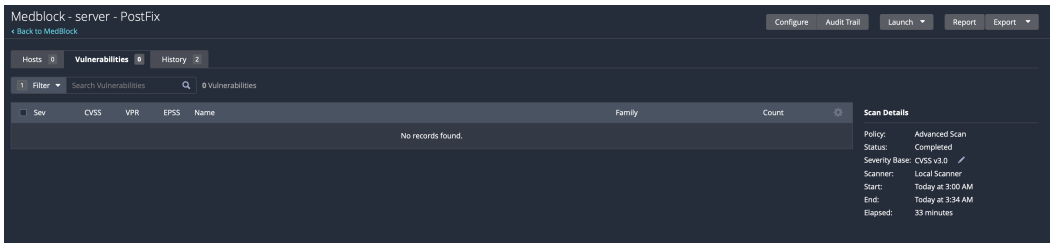


Figura 6.12: Resultado do *scan* Tenable Nessus Professional pós-remediação ao servidor.

6.3 RESUMO DO CAPÍTULO

Este capítulo documentou a avaliação de desempenho e segurança da plataforma MedBlock, produzindo um conjunto de evidências empíricas que permite avaliar a adequação do sistema ao cenário de utilização previsto no contexto do SNS.

Na dimensão de desempenho, o *benchmark* com o Hyperledger Caliper demonstrou que a rede Fabric 2.5 do MedBlock suporta um débito de escrita entre 50 e 64 TPS e um débito de leitura de até 263 TPS, com uma taxa de sucesso de 100% ao longo de 50 000 transações distribuídas em cinco níveis de carga progressivamente mais exigentes. O ponto de inflexão de latência foi identificado no nível 3 (100 TPS de escrita-alvo), a partir do qual a latência média de escrita ultrapassa 2,5 segundos e a latência máxima atinge de 8 a 13 segundos. O débito de leitura mantém-se estável em todos os níveis, com latências médias inferiores a 0,18 segundos mesmo sob saturação. Estes valores são adequados ao padrão de utilização do MedBlock, em que as operações de consentimento são menos frequentes e as consultas de autorização constituem o fluxo dominante.

Na dimensão de segurança, a avaliação abrangeu três camadas complementares. A análise estática com o SonarQube confirmou a ausência de vulnerabilidades e de *security hotspots* tanto na aplicação *web* quanto no *chaincode*. A análise dinâmica com o OWASP ZAP não identificou vulnerabilidades de risco elevado, limitando-se a alertas de risco médio e baixo relacionados à configuração da CSP e a outros cabeçalhos HTTP, todos posteriormente corrigidos e verificados com a ferramenta shcheck, que confirmou a presença de 10 cabeçalhos de segurança. A análise de infraestrutura com o Tenable Nessus Professional identificou vulnerabilidades nos protocolos TLS obsoletos e no *software* do servidor (Vim), todas remediadas e confirmadas com 0 vulnerabilidades nos *scans* de verificação.

No conjunto, os resultados indicam que a plataforma MedBlock apresenta uma postura de desempenho e segurança compatível com as exigências de um sistema de gestão de consentimento para dados de saúde, reconhecendo as limitações inerentes ao ambiente de teste (*single-peer*, *orderer* Solo) e à cobertura de testes de segurança (ferramentas automatizadas sem *pentesting* manual completo).

7

CONCLUSÕES

A plataforma resultante, designada MedBlock, assenta numa rede Hyperledger Fabric 2.5 com quatro organizações, na qual os metadados de consentimento são geridos *on-chain* por meio de um *chaincode* dedicado com três operações, concessão, revogação e consulta de autorizações, enquanto os dados clínicos permanecem *off-chain* em servidores HAPI FHIR protegidos por *reverse proxies* Nginx com mTLS e validação por chave de API. A autenticação dos pacientes é realizada por meio da integração com o sistema nacional Autenticação.Gov, via CMD, utilizando o número do SNS como identificador. As sessões são protegidas por *tokens* JWE, a camada aplicacional incorpora controlos de segurança via Helmet.js e *cookies* seguros, e o perímetro da aplicação *web* é protegido por uma WAF da Cloudflare com regras personalizadas de filtragem de agentes maliciosos e de limitação de taxa nos *endpoints* da API.

O primeiro objetivo específico consistia em analisar o panorama da partilha de dados de saúde em Portugal e em elaborar uma arquitetura de requisitos que captasse as necessidades funcionais e não funcionais, bem como as restrições regulamentares impostas pelo RGPD. Este objetivo foi cumprido ao longo dos Capítulos 1 e 3, nos quais foram identificados três desafios: fragmentação sistémica, ambiente de ameaças à cibersegurança e quadro regulatório rigoroso. A revisão da literatura no Capítulo 3 examinou seis sistemas existentes e sistematizou as suas lacunas na Tabela 3.1, confirmando que nenhum abordava simultaneamente a conformidade com o RGPD, a interoperabilidade com o FHIR e a integração com um sistema nacional de identidade digital. Os requisitos foram categorizados em funcionais (gestão de consentimento, recuperação de dados entre instituições, auditabilidade) e não funcionais (minimização de dados conforme o artigo 5.º, direito ao esquecimento conforme o artigo 17.º, privacidade desde a conceção conforme o artigo 25.º e segurança do tratamento conforme o artigo 32.º).

O segundo objetivo específico exigia a conceção de uma arquitetura de *blockchain* permissionada que separasse os metadados de consentimento *on-chain* dos dados clínicos *off-chain*, incorporando *smart contracts* para automatizar a gestão do consentimento. O Capítulo 4 documentou as decisões de *design* que sustentam

esta arquitetura: a seleção do Hyperledger Fabric 2.5 pela sua funcionalidade de canais, gestão de identidade via MSP e ausência de criptomoeda (Polge et al., 2021); a adoção do HL7 FHIR R4 como norma de dados clínicos; e a implementação de cada medida relevante do RGPD numa decisão de *design* concreta, desde a minimização de dados no *ledger* até à arquitetura híbrida que preserva o direito ao esquecimento. O Capítulo 5 concretizou esta conceção na implementação do *chaincode* `org-authorizations`, na hierarquia de PKI, com CAs por organização, e na configuração de canais com políticas de *endorsement* específicas.

O terceiro objetivo específico visava à implementação de um protótipo funcional que integrasse o Hyperledger Fabric, servidores FHIR, a CMD e uma aplicação *web*. O Capítulo 5 detalha a totalidade desta implementação: a infraestrutura de contentorização Docker com 21 *containers*, a rede Fabric multi-organização, os servidores HAPI FHIR com bases de dados PostgreSQL protegidos por *reverse proxies* Nginx, o *backend* em TypeScript/Express com integração do Fabric SDK, o fluxo de autenticação CMD, a gestão de sessões JWE, a REST API com 14 *endpoints* organizados por domínio funcional, a aplicação *web frontend* e os controlos de segurança abrangendo a camada aplicacional e o perímetro via WAF da Cloudflare.

O quarto objetivo específico exigia a avaliação do desempenho e da segurança da plataforma. O Capítulo 6 apresenta os resultados de ambas as dimensões. Na avaliação de desempenho, o *benchmarking* com o Hyperledger Caliper, ao longo de cinco níveis de carga progressivos, demonstrou que o *throughput* de escrita estabiliza entre 50 e 64 TPS, independentemente da taxa de envio-alvo, enquanto o *throughput* de leitura atinge um pico de 263 TPS. O ponto de inflexão de latência foi identificado no nível de 100 TPS de escrita-alvo, a partir do qual a latência média de escrita ultrapassa 2,5 segundos. A latência de leitura mantém-se abaixo de 0,18 segundos mesmo sob saturação. O resultado mais expressivo é a ausência total de falhas em 50 000 transações submetidas, o que corresponde a uma taxa de sucesso de 100%. Na dimensão de segurança, a análise estática com o SonarQube confirmou a ausência de vulnerabilidades e de *security hotspots* tanto na aplicação *web* quanto no *chaincode*. A análise dinâmica com o OWASP ZAP não identificou vulnerabilidades de risco elevado, e a verificação com a ferramenta shcheck confirmou a presença de 10 cabeçalhos de segurança, sendo 0 em falta. A análise de infraestrutura com o Nessus Professional identificou vulnerabilidades nos protocolos TLS obsoletos e no *software* do servidor, todas remediadas e confirmadas com zero vulnerabilidades nos *scans* de verificação subsequentes.

A primeira contribuição desta tese consiste numa arquitetura de referência validada para a partilha de dados de saúde baseada em *blockchain*, enquadrada nas

restrições regulatórias e de infraestrutura do contexto português. A revisão da literatura conduzida no Capítulo 3 demonstrou que a maioria dos protótipos existentes foi desenvolvida fora da Europa e sem considerar os mecanismos de conformidade específicos do RGPD, e que nenhuma investigação publicada abordava simultaneamente a conformidade com o RGPD, a integração com a infraestrutura digital e os sistemas de identidade existentes do SNS, a gestão de consentimento baseada no Hyperledger Fabric, com interoperabilidade com o FHIR, e uma avaliação de segurança estruturada. O MedBlock preenche esta lacuna ao propor e validar uma arquitetura concebida de raiz para o contexto português, constituindo um modelo que outros sistemas nacionais de saúde com infraestruturas fragmentadas e legislação rigorosa em matéria de proteção de dados podem adaptar.

A segunda contribuição reside na evidência funcional de viabilidade técnica. Num campo dominado por propostas conceptuais sem implementação ou com protótipos de âmbito limitado (Xi et al., 2022), o protótipo MedBlock demonstra que a arquitetura proposta é tecnicamente concretizável e não apenas teoricamente coerente. A implementação abrange uma pilha tecnológica vasta, desde a rede *blockchain* e o *chaincode* até à aplicação *web* com autenticação nacional e os resultados da avaliação de desempenho e de segurança constituem dados empíricos que permitem a comparação com resultados publicados na literatura académica e informam futuras decisões de *design* para plataformas semelhantes.

7.1 LIMITAÇÕES

O ambiente de teste utilizado para a avaliação de desempenho apresenta limitações que condicionam a extrapolação dos resultados. O *benchmarking* foi executado numa configuração com um único *peer* por organização e um *orderer* Solo, num único servidor. Estas condições diferem substancialmente de um sistema de produção, no qual múltiplos *peers* por organização, um serviço de *ordering* baseado em Raft e a distribuição geográfica dos nós introduziriam variáveis adicionais de latência de rede, de contenção de recursos e de tolerância a falhas. Os valores de *throughput* e latência reportados no Capítulo 6 devem, por conseguinte, ser interpretados como indicadores do comportamento do *chaincode* e da configuração base da rede, e não como projeções diretas do desempenho em ambiente de produção.

A avaliação de segurança baseou-se exclusivamente em ferramentas automatizadas: análise estática de código com o SonarQube, análise dinâmica com o OWASP ZAP e análise de vulnerabilidades de infraestrutura com o Tenable Nessus Professional.

Embora esta abordagem tenha proporcionado uma cobertura adequada para um protótipo de investigação, identificando e permitindo a remediação de todas as vulnerabilidades detetadas, não substitui a realização de testes de penetração manuais conduzidos por especialistas em segurança. A ausência de *pentesting* manual implica que vulnerabilidades de lógica de negócio, falhas de controlo de acesso contextual e vetores de ataque que requerem um encadeamento criativo de técnicas podem não ter sido identificados.

A escalabilidade sob cargas de dimensão nacional permanece uma limitação reconhecida da arquitetura do Hyperledger Fabric. O ponto de inflexão de latência identificado a 100 TPS de envio-alvo, a partir do qual a latência média ultrapassa 2,5 segundos e a latência máxima atinge valores entre 8 e 13 segundos, indica que um *deployment* à escala do SNS exigiria otimizações arquitetónicas. Entre estas incluem-se a adição de múltiplos *peers* por organização para distribuição de carga, a transição para um serviço de *ordering* Raft com tolerância a falhas bizantinas e, potencialmente, o particionamento de canais por região. Não obstante, os valores de *throughput* de leitura, superiores a 228 TPS, mesmo sob saturação, excedem as necessidades esperadas de um sistema de consulta de consentimentos no contexto do SNS, em que as operações de concessão e revogação de acesso são inerentemente menos frequentes do que as consultas de autorização.

7.2 PERSPETIVAS FUTURAS

A primeira linha de trabalho futuro consistiria na expansão da rede para uma configuração com múltiplos *peers* e um serviço de *ordering* Raft. A transição do *orderer* Solo para um *cluster* Raft com múltiplos nós de *ordering* introduziria tolerância a falhas e permitiria avaliar o comportamento da rede em condições mais próximas de um ambiente de produção. A adição de múltiplos *peers* por organização possibilitaria a distribuição de carga de *endorsement* e de consultas, potencialmente elevando o teto de *throughput* identificado no Capítulo 6. A realização de um *benchmarking* comparativo com o Hyperledger Caliper, utilizando a mesma metodologia de cinco níveis de carga, permitiria quantificar o impacto destas alterações arquitetónicas no *throughput* e na latência.

A segunda linha de trabalho futuro abordaria a gestão de consentimento do lado das organizações de saúde. Na versão atual do MedBlock, a gestão do consentimento é iniciada exclusivamente pelo paciente, que autoriza ou revoga o acesso de uma organização aos seus dados. A implementação de funcionalidades equivalentes do

lado das organizações estava prevista no plano inicial do projeto, mas não chegou a ser concretizada devido às restrições temporais. O desenvolvimento de um *chaincode* complementar que conferisse às organizações a capacidade de consultar a lista dos pacientes que lhes concederam autorização e de revogar, do seu lado, o acesso a determinados pacientes reforçaria a governação descentralizada da rede. Esta autonomia operacional permitiria que cada organização gerisse as suas relações de consentimento de forma independente, sem depender exclusivamente da ação do paciente.

A terceira linha de trabalho futuro direcionar-se-ia à integração com a infraestrutura real do SNS. O protótipo atual opera com dados clínicos sintéticos em servidores FHIR isolados, validando decisões de *design* sem processar dados reais de pacientes. Um piloto com uma unidade de saúde ou hospital, estabelecendo a ligação aos sistemas de informação existentes por meio do HL7 FHIR e utilizando dados reais anonimizados, permitiria avaliar a viabilidade da plataforma em condições operacionais concretas. Esta integração exigiria a resolução de desafios adicionais, nomeadamente a interoperabilidade com sistemas legados, a articulação com as políticas de dados do SPMS e a avaliação da usabilidade junto de profissionais de saúde e de pacientes, cuja aceitação constitui um fator determinante para a adoção efetiva da plataforma.

A quarta linha de trabalho futuro incidiria na realização de testes de penetração manuais e na obtenção de certificação de segurança. A avaliação de segurança conduzida no Capítulo 6, embora tenha demonstrado a ausência de vulnerabilidades detetáveis por ferramentas automatizadas, não dispensaria a execução de *pentesting* manual, abrangendo categorias de teste que requerem raciocínio contextual e encadeamento criativo de vetores de ataque. A complementação com uma avaliação NIST, mapeando cada subcategoria das cinco funções da *framework* aos controlos implementados na plataforma, constituiria o passo necessário para alinhar a postura de segurança do MedBlock com os requisitos de certificação exigidos para sistemas de informação de saúde no contexto do SNS.

BIBLIOGRAFIA

Agência para a Modernização Administrativa (AMA) (2026). *Autenticação.Gov — Documentação Técnica*. República Portuguesa. https://www.autenticacao.gov.pt/.

Androulaki, Elli et al. (2018). «Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains». Em: *Proceedings of the Thirteenth EuroSys Conference*. DOI: 10.1145/3190508.3190538.

Atzei, Nicola, Massimo Bartoletti e Tiziana Cimoli (2017). «A Survey of Attacks on Ethereum Smart Contracts (SoK)». Em: *Principles of Security and Trust (POST 2017)*. Springer. DOI: 10.1007/978-3-662-54455-6_8.

Azaria, A. et al. (2016). «MedRec: Using Blockchain for Medical Data Access and Permission Management». Em: *Proc. 2nd Int. Conf. Open and Big Data (OBD)*. Vienna, Austria, pp. 25–30. DOI: 10.1109/OBD.2016.11.

Bessani, Alysson, João Sousa e Eduardo E. P. Alchieri (2014). «State Machine Replication for the Masses with BFT-SMaRt». Em: *2014 44th Annual IEEE/IFIP International Conference on Dependable Systems and Networks*, pp. 355–362. DOI: 10.1109/DSN.2014.43.

Checkmarx (2024). *ZAP – Zed Attack Proxy*. URL: https://www.zaproxy.org/ (acedido em 12/03/2026).

CompTIA Blockchain Advisory Council (2023). *Blockchain Terminology: A Glossary for Beginners*. Website. https://connect.comptia.org/content/articles/blockchain-terminology.

ConsenSys (2023). *GoQuorum: Enterprise Ethereum Client Documentation*. URL: https://goquorum.readthedocs.io (acedido em 15/03/2026).

Cooper, David et al. (2008). *Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*. RFC 5280. IETF.

Cyran, M. A. (2018). «Blockchain as a Foundation for Sharing Healthcare Data». Em: *Blockchain in Healthcare Today* 1, pp. 1–6. DOI: 10.30953/bhty.v1.13.

European Data Protection Board (EDPB) (2026). *Annual Reports and Enforcement Decisions*. Website. https://edpb.europa.eu/.

European Parliament and Council of the European Union (2014). *Regulation (EU) No 910/2014 on electronic identification and trust services for electronic tran-*

sactions in the internal market and repealing Directive 1999/93/EC (eIDAS). Official Journal of the European Union, L 257/73.

European Parliament and Council of the European Union (mai. de 2016). «Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the Protection of Natural Persons with Regard to the Processing of Personal Data and on the Free Movement of Such Data (General Data Protection Regulation)». Em: *Official Journal of the European Union* L 119, pp. 1–88.

Gangula, R. et al. (2021). «Leveraging the Hyperledger Fabric for Enhancing the Efficacy of Clinical Decision Support Systems». Em: *Blockchain in Healthcare Today* 4. DOI: 10.30953/bhty.v4.154.

Haber, S. e W. S. Stornetta (1991). «How to Time-Stamp a Digital Document». Em: *Journal of Cryptology* 3.2, pp. 99–111. DOI: 10.1007/BF00196791.

Harman, L. B., C. A. Flite e K. Bond (2012). «Electronic Health Records: Privacy, Confidentiality, and Security». Em: *Virtual Mentor* 14.9, pp. 712–719. DOI: 10.1001/virtualmentor.2012.14.9.stas1-1209.

Hevner, A. R. et al. (2004). «Design Science in Information Systems Research». Em: *MIS Quarterly* 28.1, pp. 75–105. DOI: 10.2307/25148625.

HL7 International (2019). *HL7 FHIR Release 4 (R4)*. https://hl7.org/fhir/R4/.

Hyperledger Foundation (2026). *Hyperledger Caliper — Blockchain Benchmark Framework*. Documentation. https://hyperledger.github.io/caliper/.

IBM Security and Ponemon Institute (jul. de 2025). *Cost of a Data Breach Report 2025*. IBM Corporation, Armonk, NY, USA. https://www.bakerdonelson.com/webfiles/Publications/20250822_Cost-of-a-Data-Breach-Report-2025.pdf.

Koens, Tommy e Erik Poll (2018). «What Blockchain Alternative Do You Need?» Em: *Data Privacy Management, Cryptocurrencies and Blockchain Technology*. Springer. DOI: 10.1007/978-3-030-00305-0_9.

Lammerts, Melvin (2023). *shcheck – Security Headers Check*. URL: https://github.com/melvinsh/shcheck (acedido em 12/03/2026).

Martins, D. (2020). *How Portugal is Advancing the Use of eHealth in Europe*. Interview published by Healthcare IT News, HIMSS Media. https://www.healthcareitnews.com/news/emea/how-portugal-advancing-use-ehealth-europe.

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

National Institute of Standards and Technology (abr. de 2018). *Framework for Improving Critical Infrastructure Cybersecurity, Version 1.1*. https://nvlpubs.nist.gov/nistpubs/cswp/nist.cswp.04162018.pdf.

Oki, O. A., A. O. Agbeyangi e A. Mgidi (2024). *Blockchain in Healthcare: Implementing Hyperledger Fabric for Electronic Health Records at Frere Provincial Hospital*. arXiv preprint arXiv:2407.15876. https://arxiv.org/abs/2407.15876.

Ongaro, Diego e John Ousterhout (2014). «In Search of an Understandable Consensus Algorithm». Em: *2014 USENIX Annual Technical Conference (USENIX ATC 14)*, pp. 305–319.

OWASP Foundation (2020). *Web Security Testing Guide v4.2*. https://owasp.org/www-project-web-security-testing-guide/.

Peffers, K. et al. (2007). «A Design Science Research Methodology for Information Systems Research». Em: *Journal of Management Information Systems* 24.3, pp. 45–77. DOI: 10.2753/MIS0742-1222240302.

Polge, J., J. Robert e Y. Le Traon (2021). «Permissioned Blockchain Frameworks in the Industry: A Comparison». Em: *ICT Express* 7.2, pp. 229–233. DOI: 10.1016/j.icte.2020.09.002.

R3 (2024). *Corda: Enterprise Blockchain Platform Documentation*. URL: https://docs.r3.com (acedido em 15/03/2026).

Ramachandran, M. (2023). «S³EF-HBCAs: Secure and Sustainable Software Engineering Framework for Healthcare Blockchain Applications». Em: *Blockchain in Healthcare Today* 6.2. DOI: 10.30953/bhty.v6.286.

Shen, B., J. Guo e Y. Yang (2019). «MedChain: Efficient Healthcare Data Sharing via Blockchain». Em: *Applied Sciences* 9.6, p. 1207. DOI: 10.3390/app9061207.

SonarSource (2024). *SonarQube: Code Security, Quality & Static Analysis Tool*. URL: https://www.sonarsource.com/products/sonarqube/ (acedido em 12/03/2026).

SPMS — Serviços Partilhados do Ministério da Saúde (2023). *HealthData@PT — Setting up a Health Data Access Body in Portugal*. EU4Health Programme 2021–2027 action documentation. https://www.spms.min-saude.pt/healthdatapt-eng/.

— (2026a). *Official Institutional Documentation*. Website. https://www.spms.min-saude.pt/.

— (2026b). *OurHealth@PT — MyHealth@EU Cross-Border Digital Infrastructure*. Website. https://www.spms.min-saude.pt/ourhealthpt-eng/.

Szabo, Nick (1997). «Formalizing and Securing Relationships on Public Networks». Em: *First Monday* 2.9. DOI: 10.5210/fm.v2i9.548.

TEHDAS — Towards the European Health Data Space (mar. de 2023). *Portugal Country Visit Factsheet*. Joint Action co-funded by the Health Programme of the European Union. https://tehdas.eu/app/uploads/2023/03/portugal-country-visit-factsheet-03-2023.pdf.

Tenable, Inc. (2024). *Nessus Vulnerability Scanner*. URL: https://www.tenable.com/products/nessus (acedido em 12/03/2026).

Theodouli, A. et al. (2018). «On the Design of a Blockchain-Based System to Facilitate Healthcare Data Sharing». Em: *Proc. 17th IEEE Int. Conf. Trust, Security and Privacy in Computing and Communications*. New York, NY, USA.

U.S. Department of Health and Human Services (ago. de 1996). *Health Insurance Portability and Accountability Act of 1996 (HIPAA)*. Public Law 104-191. https://www.hhs.gov/hipaa/.

Wüst, Karl e Arthur Gervais (2018). «Do you need a Blockchain?» Em: *2018 Crypto Valley Conference on Blockchain Technology (CVCBT)*. IEEE, pp. 45–54. DOI: 10.1109/CVCBT.2018.00011.

Xi, P., X. Zhang e L. Wang (2022). «A Review of Blockchain-Based Secure Sharing of Healthcare Data». Em: *Applied Sciences* 12.15, p. 7912. DOI: 10.3390/app12157912.

Yaga, Dylan et al. (2018). *Blockchain Technology Overview*. NIST Interagency Report 8202. National Institute of Standards e Technology. DOI: 10.6028/NIST.IR.8202.

Zhang, A. e X. Lin (2018). «Towards Secure and Privacy-Preserving Data Sharing in e-Health Systems via Consortium Blockchain». Em: *Journal of Medical Systems* 42, p. 140. DOI: 10.1007/s10916-018-0995-5.

APÊNDICES

APÊNDICE A

Este apêndice reúne artefactos técnicos cuja extensão impede a sua inclusão integral no corpo principal do documento, mas cuja consulta é relevante para a auditoria, reprodução e revisão do protótipo MedBlock.

A.1 CÓDIGO-FONTE DO CHAINCODE `ORG-AUTHORIZATIONS`

A Listagem 4 reproduz, na íntegra, o código-fonte do *smart contract* `OrgAuthorizationsContract`, implementado em JavaScript sobre a API `fabric-contract-api` do Hyperledger Fabric e responsável pela gestão de autorizações de acesso entre organizações na rede MedBlock.

Listagem 4: Totalidade do código-fonte do *smart contract* `OrgAuthorizationsContract`.

```
'use strict';

const { Contract } = require('fabric-contract-api');

class OrgAuthorizationsContract extends Contract {
  _key(enrollmentId) {
    return `orgAuth:${enrollmentId}`;
  }

  async GetAuthorizedOrgs(ctx, enrollmentId) {
    if (!enrollmentId) {
      throw new Error('enrollmentId is required');
    }

    const key = this._key(enrollmentId);
    const existing = await ctx.stub.getState(key);

    if (!existing || existing.length === 0) {
      return Buffer.from(JSON.stringify({ enrollmentId, orgs: [] }));
    }

    return existing;
  }

  async AddAuthorizedOrg(ctx, enrollmentId, org) {
    if (!enrollmentId) {
```

```
      throw new Error('enrollmentId is required');
    }
    if (!org) {
      throw new Error('org is required');
    }

    const key = this._key(enrollmentId);
    const existing = await ctx.stub.getState(key);

    let record = { enrollmentId, orgs: [] };
    if (existing && existing.length > 0) {
      record = JSON.parse(existing.toString());
    }

    if (!Array.isArray(record.orgs)) {
      record.orgs = [];
    }

    if (!record.orgs.includes(org)) {
      record.orgs.push(org);
    }

    const payload = Buffer.from(JSON.stringify(record));
    await ctx.stub.putState(key, payload);
    return payload;
  }

  async RemoveAuthorizedOrg(ctx, enrollmentId, org) {
    if (!enrollmentId) {
      throw new Error('enrollmentId is required');
    }
    if (!org) {
      throw new Error('org is required');
    }

    const key = this._key(enrollmentId);
    const existing = await ctx.stub.getState(key);

    if (!existing || existing.length === 0) {
      return Buffer.from(JSON.stringify({ enrollmentId, orgs: [] }));
    }

    const record = JSON.parse(existing.toString());
    if (!Array.isArray(record.orgs)) {
      record.orgs = [];
    }

    record.orgs = record.orgs.filter((item) => item !== org);

    const payload = Buffer.from(JSON.stringify(record));
    await ctx.stub.putState(key, payload);
    return payload;
  }
}

```

```
82 module.exports = OrgAuthorizationsContract;
```

D E C L A R A Ç Ã O

Declaro, sob compromisso de honra, que o trabalho apresentado neste Projeto, com o título *"Medblock: Design e Desenvolvimento de uma Plataforma Baseada em Blockchain para Partilha Segura de Dados de Saúde"*, é original e foi realizado por Estudante José Diogo Palma Irio (2230053) sob orientação de Professor Carlos Jorge Machado Antunes(carlos.machado@ipleiria.pt).

Leiria, junho de 2026

Estudante José Diogo Palma Irio