UNIVERSIDADE FEDERAL RURAL DO SEMI-ÁRIDO
PRÓ-REITORIA DE GRADUAÇÃO
DEPARTAMENTO DE ENGENHARIAS E TECNOLOGIA
CURSO BACHARELADO INTERDISCIPLINAR EM TECNOLOGIA DA INFORMAÇÃO

MARIA APARECIDA DA SILVA

IMPLEMENTAÇÃO DE ARQUITETURA *ZERO TRUST* EM *E-COMMERCE* EM SISTEMAS DISTRIBUÍDOS NA NUVEM

PAU DOS FERROS
2026

MARIA APARECIDA DA SILVA

IMPLEMENTAÇÃO DE ARQUITETURA *ZERO TRUST* EM *E-COMMERCE* EM SISTEMAS DISTRIBUÍDOS NA NUVEM

Monografia apresentada à Universidade Federal Rural do Semi-Árido como requisito para obtenção do título de Bacharela interdisciplinar em Tecnologia da Informação.

Orientador: Walber José Adriano Silva, Prof. Dr.

PAU DOS FERROS

2026

S586i Silva, Maria Aparecida da.
Implementação de arquitetura zero trust em e-commerce em sistemas distribuídos na nuvem / Maria Aparecida da Silva. - 2026.
45 f. : il.

Orientador: Walber José Adriano Silva.
Monografia (graduação) - Universidade Federal Rural do Semi-árido, Curso de Tecnologia da Informação, 2026.

1. Arquitetura zero trust. 2. ABAC. 3. Computação serverless. 4. E-commerce. 5. Segurança em nuvem. I. Silva, Walber José Adriano, orient. II. Título.

Ficha catalográfica elaborada por sistema gerador automático em conformidade com AACR2 e os dados fornecidos pelo) autor(a).
Biblioteca Campus Pau dos Ferros
Bibliotecária: Erlanda Maria Lopes da Silva
CRB: 15/966

MARIA APARECIDA DA SILVA

IMPLEMENTAÇÃO DE ARQUITETURA *ZERO TRUST* EM *E-COMMERCE* EM SISTEMAS DISTRIBUÍDOS NA NUVEM

Monografia apresentada à Universidade Federal Rural do Semi-Árido como requisito para obtenção do título de Bacharela interdisciplinar em Tecnologia da Informação.

Defendida em: 02/07/2026

BANCA EXAMINADORA

Walber José Adriano Silva, Prof. Dr. (UFERSA)
Presidente

Reudisman Rolim de Sousa, Prof. Dr. (UFERSA)
Membro Examinador

Bruno Borges da Silva, Prof. Bel. (UFERSA)
Membro Examinador

DEDICATÓRIA

À minha mãe, que transformou suas cicatrizes em força e, mesmo sob os dias mais difíceis, nunca deixou que faltasse a educação que me trouxe até aqui. (*Presente*)

AGRADECIMENTOS

A Deus, por ser a força invisível que me sustentou em cada momento desta caminhada, concedendo-me discernimento, saúde e resiliência para superar os desafios e concluir esta etapa.

À minha família, por ser a base de absolutamente tudo o que sou. Agradeço aos meus pais e aos meus irmãos pelo amor incondicional, pelo apoio silencioso nos dias mais difíceis. Em especial, agradeço à minha mãe, a pessoa mais forte e guerreira que conheço, que há anos luta com uma coragem admirável e que tanto me ensinou com sua sabedoria, garra e exemplo de vida.

Ao meu namorado, Breno, por ser meu porto seguro ao longo de toda essa jornada. Obrigada por segurar a minha mão quando o cansaço parecia vencer, pela paciência infinita nos dias de estresse, e por vibrar com cada pequena vitória como se fosse sua. Sua presença e companheirismo tornaram a caminhada realizável.

Aos meus amigos queridos, tanto aqueles que a vida universitária me deu quanto os que me acompanham desde a infância, que transformaram os anos até aqui em uma jornada muito mais leve e memorável. Obrigada pelas conversas e pelas risadas compartilhadas nos momentos de tensão.

Ao meu orientador, Prof. Dr. Walber José Adriano Silva, pela condução técnica exemplar deste trabalho. Agradeço pela paciência dedicada, pelas correções cirúrgicas, pelo compartilhamento de seu vasto conhecimento acadêmico e pelas aulas ímpares que tanto me despertavam interesse nessa área de pesquisa.

EPÍGRAFE

"O que a vida quer da gente é coragem."

(João Guimarães Rosa)

RESUMO

A arquitetura *zero trust* é um paradigma de segurança amplamente utilizado em sistemas distribuídos na nuvem para garantir a proteção de dados em plataformas de *e-commerce*. Essa abordagem, embora eficaz, demanda uma verificação contínua de atributos contextuais, o que pode ser um desafio em termos de tempo e recursos de latência. Atualmente, busca-se otimizar esses processos sem comprometer a qualidade dos resultados de desempenho. O presente estudo explora uma lacuna relacionada à necessidade de métodos mais eficientes capazes de reduzir o tempo de execução e o uso de recursos computacionais na arquitetura *zero trust*. Essa lacuna torna-se evidente em processos que envolvem simulações complexas de controle de acesso baseado em atributos (ABAC). O principal objetivo é propor, implementar e analisar uma infraestrutura que utilize a arquitetura *zero trust* em *e-commerce* em sistemas distribuídos na nuvem. Torna-se importante avaliar o desempenho desse método, especialmente em termos de redução do tempo de processamento enquanto mantém a eficácia da segurança. A segurança tradicional baseada em perímetro envolve a validação estática de papéis (RBAC) em todo o domínio da rede, o que pode ser computacionalmente custoso e vulnerável. O método proposto inclui estabelecer as regras que definem a validação contínua na nuvem AWS sob as diretrizes do padrão NIST SP 800-207. Essa abordagem consiste em restringir o acesso a microsserviços validando a correlação de atributos dinâmicos no *backend* via Amazon API Gateway, AWS Lambda e Amazon DynamoDB. Em seguida, foi realizada a implementação da infraestrutura proposta, levando em conta essa topologia. Foram realizados testes para avaliar diferentes configurações, como o acesso de paciente, a movimentação lateral e o abuso de privilégio. Os resultados mostram que este método proporciona uma redução significativa no tempo de execução em comparação com as requisições iniciais afetadas por partidas a frio (*Cold Start*), apresentando uma latência média operacional estável de 538,8 ms para o processamento de consultas dinâmicas de atributos na *Source of Truth*. Portanto, os resultados obtidos indicam que a arquitetura proposta oferece uma solução eficiente para a segurança de *e-commerce*, reduzindo significativamente o tempo de execução e o uso de recursos computacionais em nuvens distribuídas.

Palavras-chave: arquitetura *zero trust*; ABAC; computação *serverless*; segurança em nuvem; *e-commerce*.

ABSTRACT

Zero trust architecture is a security paradigm widely used in distributed cloud systems to ensure data protection within *e-commerce* platforms. This approach, although effective, demands continuous verification of contextual attributes, which can be a challenge in terms of time and latency resources. Currently, the goal is to optimize these processes without compromising the quality of performance results. The present study explores a gap related to the need for more efficient methods capable of reducing execution time and computational resource usage in *zero trust* architecture. This gap becomes evident in processes that involve complex simulations of attribute-based access control (ABAC). The main objective is to propose, implement, and analyze an infrastructure that utilizes *zero trust* architecture in *e-commerce* within distributed cloud systems. It becomes important to evaluate the performance of this method, especially in terms of processing time reduction while maintaining security effectiveness. Traditional perimeter-based security involves static validation of roles (RBAC) across the entire network domain, which can be computationally costly and vulnerable. The proposed method includes establishing the rules that define continuous validation within the AWS cloud under the NIST SP 800-207 guidelines. This approach consists of restricting access to microservices by validating dynamic attribute correlation on the *backend* via Amazon API Gateway, AWS Lambda, and Amazon DynamoDB. Next, the implementation of the proposed infrastructure was performed, taking this topology into account. Tests were carried out to evaluate different configurations, such as data access, lateral movement, and privilege abuse. The results show that this method provides a significant reduction in execution time compared to initial requests affected by cold starts (*Cold Start*), presenting a stable operational average latency of 538.8 ms for processing dynamic attribute queries within the Source of Truth. Therefore, the obtained results indicate that the proposed architecture offers an efficient solution for *e-commerce* security, significantly reducing execution time and computational resource usage in distributed clouds.

Keywords: zero trust architecture; ABAC; serverless computing; cloud security; e-commerce.

LISTA DE FIGURAS

LISTA DE TABELAS

SUMÁRIO

1 INTRODUÇÃO

Com a rápida transformação digital, a migração dos sistemas corporativos para a nuvem (*cloud computing*) tornou-se essencial para a sobrevivência e o crescimento das empresas no mercado. Especialmente no segmento de comércio eletrônico (*e-commerce*), a dependência estrutural em relação a sistemas distribuídos na nuvem é evidente, haja vista a necessidade de processamento em tempo real de transações financeiras, gerenciamento dinâmico de inventários e armazenamento de dados altamente sensíveis. No entanto, a facilidade de acesso e a abertura da nuvem pública também aumentam consideravelmente a superfície de ataque cibernético das organizações. Sistemas comerciais de saúde digital (*e-health*), farmácias *online* e plataformas de telemedicina transacionam diariamente um volume massivo de informações críticas, abrangendo desde dados de identificação pessoal (PII) de clientes até registros de prontuários médicos e receitas controladas, os quais são protegidos por legislações rigorosas como a Lei Geral de Proteção de Dados Pessoais (LGPD).

A maior parte das aplicações web comerciais emprega o Controle de Acesso Baseado em Papéis (*Role-Based Access Control* - RBAC) como mecanismo primário de governança. Embora o RBAC simplifique a atribuição de permissões com base em funções administrativas (como cliente, administrador, cuidador), ele peca por sua natureza estática e pela total incapacidade de discernir o contexto situacional no qual o acesso é pleiteado, conforme apontado por Atlam e Yang (2025). Sob este aspecto, se um atacante obtém com sucesso os *tokens* de autenticação de um usuário administrativo ou explora uma brecha no isolamento de sessões, o impacto destrutivo decorre do comprometimento de uma sessão legítima previamente autorizada, e não de uma falha inerente à engenharia do algoritmo RBAC em si. Todavia, a ausência de mecanismos contextuais no modelo RBAC tradicional maximiza a superfície de danos, permitindo que a ameaça navegue lateralmente e exponha dados sensíveis de terceiros. Os modelos tradicionais de segurança da informação, fortemente ancorados pelo conceito de segurança de perímetro, têm se mostrado ineficazes diante de ameaças modernas persistentes e táticas avançadas de movimentação lateral. Historicamente, assumia-se que qualquer usuário, dispositivo ou requisição oriunda do ambiente interno de rede usava de um *status* de confiabilidade inerente. Essa premissa, todavia, ignora a incidência recorrente de vazamento de credenciais legítimas, ataques de engenharia social e a ação de agentes maliciosos internos (*insider threats*).

Diante de tais vulnerabilidades latentes, emerge o paradigma do *Zero Trust* (Confiança Zero), cuja máxima fundamental preconiza: nunca confiar, sempre verificar. Sob essa perspectiva,

a confiança nunca é implicitamente outorgada com base na localização física ou lógica do agente, exigindo uma validação contínua e rigorosa de cada transação de dados. A mitigação eficaz do risco de movimentação lateral e acesso indevido a dados de saúde e transações em um *e-commerce* em nuvem distribuída exige superar as limitações do modelo estático RBAC. Para viabilizar a validação experimental da arquitetura proposta, foi desenvolvido o sistema *MedTrust*, uma aplicação web que simula uma plataforma de comércio eletrônico farmacêutico. A escolha desse domínio permitiu avaliar os mecanismos de autenticação, autorização e proteção de dados em um contexto que lida com informações sensíveis, tornando-o adequado para a aplicação prática dos princípios de Confiança Zero em sistemas distribuídos na nuvem.

O escopo experimental desta pesquisa engloba o desenvolvimento de uma aplicação *web* em React (MedTrust) simulando um ecossistema de *e-commerce* de saúde integrado a um provedor de identidade (*Identity Provider* - IdP). Através dessa aplicação, configurou-se um *pipeline* de autenticação segura e isolamento de sessões na nuvem utilizando o Amazon Cognito, com a implementação de fluxos implícitos de emissão de *tokens JSON Web Tokens* (JWT). Políticas dinâmicas *Attribute-Based Access Control* (ABAC) foram codificadas e implantadas no PDP (*Policy Decision Point*) para validar, em tempo real, a relação entre o usuário autenticado e o recurso solicitado por meio de consultas síncronas à tabela de permissões armazenada no Amazon DynamoDB. Por fim, coletaram-se e analisaram-se dados quantitativos de desempenho e latência nas tomadas de decisão nos cenários de acesso legítimo e de tentativas de violação de privilégios.

Justifica-se a presença das entidades Médico e Paciente no escopo deste *e-commerce* devido à natureza regulatória do comércio farmacêutico digital. Diferente de plataformas de varejo convencional, a aquisição de medicamentos controlados ou de alta complexidade em uma farmácia *online* moderna exige a integração e a validação de receituários digitais. Sob as premissas da LGPD, o ecossistema precisa garantir de forma síncrona que o sujeito autenticado como profissional da saúde (`medicoId`) possua autoridade clínica legítima para carregar ou validar uma prescrição, enquanto o perfil do cliente (`patientId`) limita o acesso do usuário estritamente aos seus registros e pedidos autorizados, validando de forma fidedigna a aplicação do modelo ABAC em um fluxo de *checkout* comercial real.

A relevância deste estudo reside na premente necessidade industrial e acadêmica de se desenharem arquiteturas defensivas capazes de mitigar vazamentos de dados em larga escala. No ambiente de saúde digital, um incidente de segurança não acarreta somente severas penalidades

financeiras ou danos à reputação institucional, mas viola a privacidade mais íntima do cidadão e pode comprometer a integridade de tratamentos médicos em curso. Sob o ponto de vista acadêmico, esta pesquisa preenche uma lacuna relevante ao demonstrar a viabilidade prática da convergência entre modelos RBAC e ABAC sob a ótica da publicação especial NIST SP 800-207. Ademais, o estudo viabiliza uma análise quantitativa do *overhead* de latência introduzido por checagens criptográficas contínuas em microsserviços *serverless*, oferecendo dados empíricos cruciais para engenheiros de sistemas e arquitetos de soluções em nuvem.

2 FUNDAMENTAÇÃO TEÓRICA

Com o intuito de estabelecer uma base conceitual sólida para a compreensão dos mecanismos propostos neste trabalho, este capítulo aborda as premissas centrais da Arquitetura *Zero Trust*, os modelos formais de controle de acesso RBAC e ABAC, como também e o ecossistema de computação em nuvem *serverless* utilizado como substrato de infraestrutura.

2.1 Arquitetura Zero Trust (ZTA) e a Publicação Especial NIST SP 800-207

A desconstrução das fronteiras físicas corporativas imposta pela computação em nuvem e pelo trabalho remoto exigiu uma reformulação radical das estratégias de segurança da informação. A resposta formal a essa necessidade manifestou-se na publicação especial NIST SP 800-207 Rose *et al.* (2020), documento que cunhou os fundamentos conceituais da *Zero Trust Architecture* (ZTA). A filosofia central do *Zero Trust* assume que ameaças estão presentes tanto fora quanto dentro das fronteiras nominais da rede. Consequentemente, nenhum usuário, dispositivo ou sistema computacional herda um *status* de confiança implícita com base em sua localização topológica ou propriedade de rede.

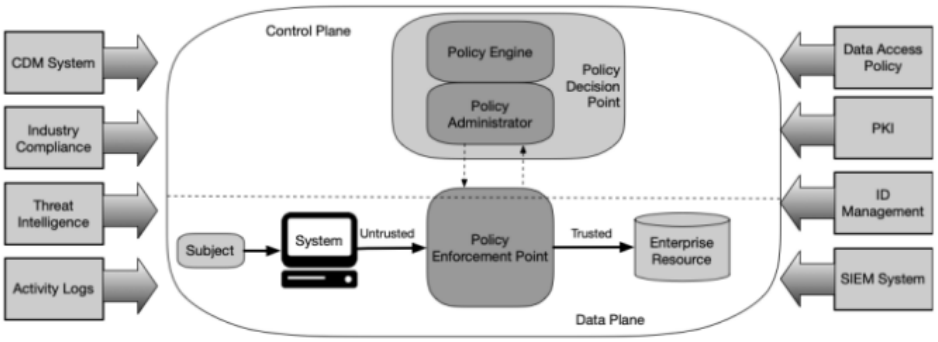
O arcabouço conceitual do NIST SP 800-207 divide a infraestrutura lógica em componentes de controle e execução. O primeiro componente crítico é o *Policy Enforcement Point* (PEP), responsável por interceptar, inspecionar e, se autorizado, estabelecer e monitorar a conexão entre o sujeito e o recurso. O PEP opera sob as ordens diretas do *Policy Decision Point*. O PDP divide-se em duas entidades funcionais: o *Policy Engine* (PE), encarregado de tomar a decisão final de conceder, negar ou revogar o acesso com base em regras de política institucional, e o *Policy Administrator* (PA), que executa os comandos lógicos emitidos pelo PE para configurar e instruir o PEP de forma dinâmica.

Conforme preconizado por Arruda *et al.* (2023), a transição para uma infraestrutura ZTA requer visibilidade holística e contínua dos fluxos de dados corporativos. A confiança deixa de ser um estado binário permanente outorgado no momento do *login* e passa a ser uma pontuação de risco efêmera calculada dinamicamente a cada nova requisição de recurso. Essa verificação contínua exige que as identidades sejam checadas por meio de atestação criptográfica rigorosa e que o menor privilégio possível seja concedido estritamente para o tempo necessário para executar aquela ação específica. Estendendo os limites de proteção na nuvem diretamente à camada de transporte e rede, DeCusatis e Liengtiraphan (2016) atestam o valor de implementar

políticas criptográficas de controle de acesso via barramentos esteganográficos ou *First Packet Authentication*, assegurando que nós não validados sequer consigam estabelecer comunicação com recursos invisíveis da rede.

Para fins de aprofundamento analítico da arquitetura abstrata preconizada por Rose *et al.* (2020), a Figura 1 mapeia as dependências lógicas fundamentais estabelecidas entre os nós funcionais do sistema. Esta estrutura conceitual de referência serve como base teórica direta para a modelagem da topologia prática implementada na nuvem AWS, detalhada adiante no fluxograma da Figura 2, na qual os componentes de decisão e execução mapeados são materializados por microsserviços integrados.

Figura 1 – Componentes lógicos e fluxos de controle essenciais da Arquitetura *Zero Trust*

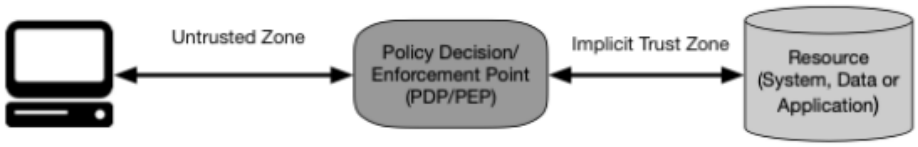


Fonte: Rose *et al.* (2020).

Analisando sistematicamente a Figura 1, constata-se a relevância de se estruturar uma separação estrita e hermética entre o plano de dados (*Data Plane*) e o plano de controle (*Control Plane*). Essa dissociação garante que os componentes de governança e estabelecimento de políticas fiquem completamente isolados dos canais de tráfego de dados sensíveis. Esta segregação, conforme preconizado pelas diretrizes normativas de referência, blinda os recursos corporativos ao assegurar que nenhuma comunicação seja realizada na camada de dados sem que o plano de controle avalie e configure explicitamente os privilégios efêmeros de acesso.

Com o propósito de detalhar a perspectiva conceitual das diferentes abordagens lógicas de implementação sugeridas pela publicação técnica norte-americana, Rose *et al.* (2020) delimitam o cenário clássico de isolamento de borda, conforme mapeado na Figura 2.

Figura 2 – Cenário conceitual de implantação Zero Trust baseado em Agente do Dispositivo Cliente e Gateway de Recursos Protegidos



Fonte: Rose *et al.* (2020).

2.1.1 Cenários Ampliados de Proteção e Vetores de Ameaça Cobertos

A transição das redes tradicionais para modelos baseados em Confiança Zero é acelerada pela sofisticação crônica de ataques de engenharia social e intrusões avançadas, como campanhas de *ransomware*. Conforme elucidado por Bediako *et al.* (2025), as táticas de *malware* contemporâneas operam de forma modular, explorando a confiança herdada de credenciais de perímetro para sequestrar diretórios internos. Em sistemas distribuídos de comércio farmacêutico, o comprometimento de um único terminal local pode infectar bases de dados relacionais e comprometer os registros de pacientes. O modelo baseado em validação consecutiva atua quebrando este vetor de ataque, assegurando que o comprometimento local do cliente de *frontend* não conceda chaves lógicas permanentes sobre os barramentos analíticos de dados.

Adicionalmente, a pulverização de cadeias de suprimentos e o uso de ecossistemas logísticos terceirizados (*cyber supply chain*) expandem os pontos de vulnerabilidade institucional. Conforme sustentado por Amaral (2022), uma infraestrutura corporativa robusta precisa tratar qualquer transação interconectada como potencialmente comprometida, auditando dinamicamente as assinaturas dos pacotes e isolando os microsserviços críticos por meio de Pontos de Execução de Política altamente resilientes. Assim, o fluxo de compras farmacêuticas integra-se a este preceito ao exigir que as dependências de microsserviços terceirizados sejam submetidas às mesmas restrições impostas aos usuários internos.

2.1.2 A Transição de Redes Virtuais Privadas (VPN) para o Perímetro de Identidades

Historicamente, as organizações subsidiaram o tráfego remoto de dados sensíveis por intermédio de Redes Virtuais Privadas (*Virtual Private Networks* - VPNs). O modelo clássico de VPN concede ao usuário autenticado um endereço IP virtual interno legítimo, permitindo

a visibilidade total da rede corporativa a partir daquele ponto lógico. Clemente (2025) aponta que essa regra estática vai contra o princípio do privilégio mínimo, posto que uma credencial vazada de VPN faculta a um invasor escanear e explorar toda a topologia interna organizacional de forma irrestrita.

Em substituição a esta metodologia perimetral obsoleta, o modelo *Zero Trust Network Access* (ZTNA) redireciona o perímetro de segurança para o plano da identidade e do contexto do recurso. Não há concessão de endereçamento IP fixo para o terminal cliente. Cada chamada HTTP é julgada de forma individual e isolada, destruindo a possibilidade de exploração ativa de portas lógicas abertas no servidor.

2.2 Modelos de Controle de Acesso: Limitações do RBAC e a Granularidade do ABAC

A governança e a segurança de dados em larga escala dependem diretamente da maturidade do modelo de controle de acesso escolhido para proteger os endpoints da aplicação. O modelo *Role-Based Access Control* dominou o desenvolvimento de sistemas comerciais nas últimas décadas. Sua lógica baseia-se no agrupamento de privilégios operacionais em "papéis"lógicos que refletem a divisão funcional da empresa (como analista, gerente, diretor, cliente). Quando um usuário autentica-se no sistema, o RBAC injeta suas permissões correspondentes ao papel designado, as quais permanecem imutáveis durante toda a validade daquela sessão.

Embora o RBAC ofereça excelente eficiência administrativa para gerenciar milhares de usuários estáveis, ele exibe falhas graves quando confrontado com ataques avançados em nuvem. O principal vetor de falha do RBAC reside em sua natureza estática e na ausência de sensibilidade contextual Atlam e Yang (2025). Se um agente malicioso consegue comprometer a conta de um "Cuidador"ou "Enfermeiro"em um sistema médico, o RBAC concede a esse invasor acesso irrestrito a todos os prontuários cobertos por aquele papel administrativo. Não há mecanismos nativos no RBAC clássico para impedir que o usuário logado tente devassar dados de pacientes com os quais ele não possui qualquer vínculo clínico ou assistencial ativo, configurando o cenário clássico de movimentação lateral ilícita.

Para mitigar essa vulnerabilidade, o modelo *Attribute-Based Access Control* reestrutura o processo de autorização. O ABAC não avalia apenas quem é o usuário, mas calcula o direito de acesso em tempo real examinando as características multidimensionais dos atores e recursos envolvidos, uma abordagem que serve de base para implementações práticas como a da Amazon Web Services (2026a). As políticas ABAC são estruturadas a partir de quatro categorias de

atributos:

- **Atributos do Sujeito (*S*):** Características intrínsecas do usuário que pleiteia o acesso, incluindo idade, departamento, chaves de autenticação de duplo fator e chaves customizadas de vinculação.
- **Atributos do Recurso (*R*):** Metadados acoplados diretamente ao dado alvo, como proprietário do registro, classificação de confidencialidade, data de criação e restrições legais associadas.
- **Atributos de Ambiente (*E*):** Fatores contextuais independentes do sujeito e do recurso, como endereço IP de origem, horário da requisição, localização geográfica e assinatura digital do hardware.
- **Atributos de Ação (*A*):** A operação que se deseja executar sobre o recurso, tipicamente mapeada pelos métodos HTTP (GET, POST, PUT, DELETE).

O PDP ABAC compõe esses elements em predicados lógicos booleanos complexos e flexíveis. Como demonstrado por Atlam e Yang (2025), a unificação dos dois modelos resulta em uma arquitetura híbrida de defesa em profundidade altamente eficaz. O RBAC atua como a primeira barreira macro de filtragem, e o ABAC funciona como o micropoliciamento contextual de grão fino, bloqueando requisições que violem a lógica de negócios mesmo que apresentem credenciais formalmente válidas. No ambiente de redes corporativas centralizadas de alta sensibilidade, Wu e Yan (2021) elucidam que associar as checagens a uma verificação de identidade real e inspecionada sob premissas de Confiança Zero cria limites dinâmicos intransponíveis a invasores externos ou internos.

2.2.1 Sistemas Multiníveis e a Preservação da Integridade Criptográfica

A complexidade de sistemas de saúde distribuídos em nuvem requer salvaguardas adicionais para garantir que os dados não sofram corrupção durante o trânsito entre microsserviços. Conforme observado por Silva (2022), a integração de políticas multiníveis de segurança associadas a redes *Zero Trust* fornece uma esteira previsível de integridade, impedindo que a modificação local de *payloads* no navegador consiga enganar as rotinas de persistência no banco de dados distribuído.

A criptografia de chaves assimétricas aplicada aos cabeçalhos HTTP garante a autenticidade da transação. Conforme demonstrado por Santos Júnior, Cabral e Lima (2022), em ambientes de redes internas potencialmente comprometidas, delegar a descriptografia e liberação

de dados exclusivamente para o momento em que todos os atributos contextuais são validados de forma síncrona blinda o sistema contra ataques de interceptação *(Man-in-the-Middle)*. Deste modo, o sistema experimental *MedTrust* adota este preceito ao blindar as respostas lógicas do microsserviço da Lambda através de atestações baseadas em assinaturas do Cognito.

2.3 A Lei Geral de Proteção de Dados (LGPD) e o Tratamento de Dados Sensíveis de Saúde

O advento da Lei Geral de Proteção de Dados Pessoais, Lei n.º 13.709/2018, reestruturou as obrigações legais impostas a plataformas comerciais que coletam e processam informações de cidadãos em território nacional, conforme instituído pelo Brasil (2018). No ecossistema de *e-commerce* voltado à saúde digital, o rigor regulatório é maximizado, visto que registros de prontuários médicos, receitas controladas e históricos de dispensação farmacêutica são formalmente classificados pelo Artigo 5.º da referida lei como dados pessoais sensíveis.

O tratamento dessa categoria de dados impõe restrições severas, exigindo das organizações a adoção de salvaguardas administrativas e técnicas aptas a mitigar incidentes de segurança, vazamentos ilícitos ou acessos não autorizados. Sob a ótica da governança corporativa, mecanismos tradicionais baseados puramente na autenticação de perímetro demonstram-se insuficientes para o cumprimento das diretrizes de privacidade por *design* (*privacy by design*).

Nesse cenário, o micropoliciamento dinâmico fornecido pelo modelo ABAC atua como um vetor tecnológico direto para a conformidade legal. Ao condicionar a entrega do dado sensível à validação síncrona de atributos contextuais na *Source of Truth*, o motor do sistema impede de forma ativa o desvio de finalidade e a movimentação lateral. Essa amarração arquitetural garante que a aplicação atenda de forma estrita aos princípios de necessidade, transparência e segurança estipulados pela LGPD, ao mesmo tempo em que preserva a consistência e a persistência dos registros no banco de dados distribuído, conforme os preceitos clássicos de transações seguras descritos por Elmasri e Navathe (2011).

2.4 Computação *Serverless* e Serviços Gerenciados na AWS

A materialização prática de uma ZTA com verificação contínua exige uma infraestrutura subjacente de alto desempenho e baixo acoplamento lógico. A computação *serverless* (sem servidor) emergiu como o modelo ideal para hospedar essas arquiteturas distribuídas modernos.

Na computação baseada em servidores tradicionais, a manutenção de instâncias ligadas ininterruptamente introduz superfícies de ataque complexas (como portas abertas desnecessárias, sistemas operacionais desatualizados e má configuração de *firewalls*). O paradigma *serverless* transfere a gerência completa da infraestrutura física, do *kernel* do sistema operacional e do isolamento de rede para o provedor de nuvem pública, restando à organização focar estritamente na qualidade lógica de suas políticas de código.

No ecossistema da Amazon Web Services (AWS), três serviços gerenciados integram-se de forma nativa para construir os pilares da Confiança Zero:

- **Amazon Cognito:** Atua como o Provedor de Identidade. Ele gerencia de forma segura o cadastro de usuários, autenticação multifator (MFA) e o isolamento rígido de sessões por meio do padrão OAuth 2.0. O Cognito emite tokens JWT assinados digitalmente por uma chave criptográfica privada da AWS (verificável publicamente por meio do endpoint JWKS), garantindo a integridade inquestionável dos dados de identidade que trafegam pela rede distribuída.
- **Amazon API Gateway:** Funciona como o *Policy Enforcement Point*. Ele expõe endpoints REST públicos HTTPS protegidos por criptografia de ponta a ponta (TLS). O API Gateway intercepta todas as requisições de entrada, gerencia as regras de CORS para evitar ataques de injeção de scripts maliciosos de origens não autorizadas e delega o processo de tomada de decisão de segurança a autorizadores customizados antes de permitir qualquer tráfego em direção aos microsserviços internos de dados.
- **AWS Lambda:** Desempenha o papel de *Policy Decision Point*. Trata-se de um serviço computacional orientado a eventos que executa trechos de código de forma isolada dentro de micro-VMs de altíssimo desempenho *(Firecracker)*. A Lambda é invocada sob demanda pelo API Gateway, recebe os dados criptográficos extraídos do token do usuário e executa o processamento matemático das políticas booleanas ABAC em frações de milissegundos, respondendo ao gateway se a requisição deve prosseguir ou ser abortada imediatamente na borda da rede.

O monitoramento detalhado dessas frentes de segurança em tempo de execução guarda estrita similaridade conceitual com os pressupostos defendidos por Yao e Zhang (2021), cujos estudos corroboram que a compilação contínua de retratos de risco comportamentais de usuários resulta em uma árvore de tomada de decisões de privilégio granular muito mais eficaz para as redes empresariais.

2.5 Modelos de Consistência em Bancos de Dados NoSQL Distribuídos

Em sistemas distribuídos na nuvem que utilizam o paradigma NoSQL, como o Amazon DynamoDB, a replicação de dados entre múltiplos nós geográficos introduz desafios estruturais relacionados à consistência da informação. Conforme preconizado pelo Teorema CAP descrito por Fox e Brewer (1999), um sistema de dados distribuído é incapaz de garantir simultaneamente consistência absoluta (*Consistency*), disponibilidade integral (*Availability*) e tolerância a particionamento de rede (*Partition Tolerance*), exigindo que a arquitetura decline para um compromisso operacional entre esses fatores.

Por padrão, a maior parte das soluções NoSQL adota o modelo de consistência eventual (*Eventually Consistent Reads*). Sob essa premissa teórica, discutida por Elmasri e Navathe (2011), as modificações geradas por operações de escrita ou exclusão não são replicadas de forma síncrona para todos os nós de armazenamento de maneira imediata. Consequentemente, existe uma janela temporal na qual uma consulta de leitura pode retornar uma cópia desatualizada ou um item que já foi instruído à deleção.

Para cenários críticos de segurança cibernética e infraestruturas baseadas em micropoliciamento dinâmico (*Zero Trust Architecture*), esse atraso de propagação na rede pública é nocivo, pois pode gerar vulnerabilidades de concorrência (*race conditions*), permitindo que um usuário acesse recursos mesmo após ter seus privilégios revogados na base de dados. Para mitigar esse risco de segurança, a literatura de boas práticas arquiteturais da Amazon Web Services (2026a) recomenda a ativação do modelo de consistência forte (*Strongly Consistent Reads*). Essa configuração força o motor do banco de dados distribuído a retornar uma resposta que reflita todas as modificações bem-sucedidas anteriores, garantindo que a leitura retorne o estado mais recente e idêntico do dado a partir de todos os nós de armazenamento, eliminando a janela de exposição pós-revogação.

3 TRABALHOS RELACIONADOS

Com o propósito de mapear o cenário atual e contrastar as contribuições deste trabalho com a literatura científica existente, este capítulo analisa estudos focados na aplicação de princípios de *Zero Trust* e no controle de acesso adaptativo em nuvem distribuída.

No âmbito da infraestrutura de redes complexas e segurança perimetral dinâmica, Barbosa e Mattos (2025) propuseram o arcabouço ZETIn, uma infraestrutura direcionada à conectividade segura em Redes em Malha Ad-Hoc Baseadas em Inteligência Artificial Generativa. O estudo concentrou-se em demonstrar que o paradigma *Zero Trust* introduz um *overhead* de latência estatisticamente desprezível face ao ganho exponencial em resiliência e na orquestração automatizada de políticas de rede. Embora o foco de Barbosa e Mattos (2025) estivesse na camada de transporte e roteamento em topologias altamente mutáveis de nós móveis, a comprovação estatística da viabilidade da latência contínua valida a premissa deste trabalho quanto à aplicação de validações criptográficas em tempo de execução no nível de aplicação Web. Enquanto o ZETIn foca na criptografia e túneis de rede ad-hoc, este TCC avança ao aplicar a checagem na camada 7 (Aplicação) do modelo OSI, inspecionando *tokens* lógicos JWT em microsserviços comerciais.

Em uma vertente direcionada aos ambientes industriais e sistemas legados, Nascimento (2022) avaliou os desafios inerentes à aplicação prática de princípios ZT em redes de controle industrial (ICN) diante do advento da Indústria 4.0. O autor acentua as limitações das defesas tradicionais baseadas em isolamento físico de perímetros e propõe técnicas de acoplamento de *gateways* de segurança que traduzem requisições sem exigir a substituição em massa de sensores e atuadores antigos. O presente TCC compartilha com a filosofia de Nascimento (2022) no tocante à preocupação de isolar o recurso crítico sem alterar a camada de armazenamento de dados estruturais. No entanto, esta monografia diferencia-se substancialmente ao focar no sistema nativo em nuvem *serverless* para aplicações de comércio eletrônico, em que o controle de acesso dinâmico ocorre diretamente via análise comparativa entre os *claims* de atributos do sujeito (extraídos do *token*) e os atributos dos recursos validados junto à base de dados do Amazon DynamoDB, abandonando o uso de tabelas estáticas de IPs ou *addresses* MAC industriais.

Abordando a combinação de modelos formais, Atlam e Yang (2025) propuseram uma abordagem de controle de acesso híbrido capaz de fundir a granularidade fina do ABAC à clareza organizacional do RBAC em sistemas de saúde digital. Os autores asseguram que

sistemas médicos requerem flexibilidade acrescida para contornar políticas estáticas em situações de emergência sem abrir mão do rastreamento estrito de responsabilidade (*accountability*). Este TCC adota a lógica de modelagem de Atlam e Yang (2025) no tocante à segregação abstrata de papéis e atributos, mas preenche uma lacuna empírica relevante ao transformar as proposições puramente teóricas e matemáticas dos referidos autores em um sistema prático funcional implantado na nuvem pública AWS, medindo em tempo real os tempos de latência diante de testes de penetração e movimentação lateral. Adicionalmente, esta pesquisa expande a validação experimental ao incorporar o cenário de abuso de privilégio, avaliando a resiliência do micropoliciamento dinâmico quando requisições internas tentam violar os limites de acesso estabelecidos.

Por fim, no campo da observabilidade e auditoria, Jesus Júnior e Albuquerque (2025) demonstraram que a mera implementação de checagens pontuais não satisfaz os critérios de segurança contínua estipulados pelo NIST SP 800-207 se não houver um arcabouço rígido de coleta de métricas em tempo real. Os autores explicam que sem uma infraestrutura de auditoria ativa, as corporações tendem a cair em um estado de falsa conformidade regulatória. O protótipo desenvolvido nesta monografia incorpora diretamente as críticas de Jesus Júnior e Albuquerque (2025) ao implementar uma esteira dedicada de *logs* para auditoria visual imediata na interface cliente (*Dashboard*), capturando frações de performance de rede que atestam a eficácia e a transparência do modelo ABAC proposto, servindo de contrapondo prático aos modelos teóricos analíticos de monitoramento.

Avançando no tema de integração de identidades federadas em ambientes corporativos externos, Hatakeyama, Kotani e Okabe (2021) fornecem contribuições fundamentais no desenho do arcabouço *Zero Trust Federation*, comprovando que a partilha e controle dinâmico de contextos sob a tutela do usuário final reduz significativamente a vulnerabilidade a vazamentos em sistemas heterogêneos. De igual modo, Dimitrakos *et al.* (2020) expandem esses limites ao fundirem motores de políticas baseadas em atributos a sistemas autônomos de cálculo de pontuação contínua de confiança (*Continuous Authorization*), um elemento conceitual basilar para sistemas IoT de consumo que valida a estratégia de reavaliação de privilégios adotada nesta monografia.

Para consolidar as distinções levantadas ao longo deste capítulo, a Tabela 1 apresenta um resumo comparativo destacando as principais diferenças de escopo, controle de acesso e validação entre os trabalhos relacionados e a solução proposta.

Tabela 1 – Tabela resumo das principais diferenças entre os trabalhos relacionados e a solução proposta

Trabalho	**Contexto / Foco**	**Controle de Acesso**	**Tipo de Validação**
Barbosa e Mattos (2025)	Redes em malha (*mesh*) *ad-hoc*	Automação baseada em IA de rede	Simulação computacional de tráfego
Nascimento (2022)	Redes industriais e sistemas legados (ICN)	*Gateways* de isolamento de *hardware*	Estudo analítico de viabilidade técnica
Atlam e Yang (2025)	Ambientes de saúde digital corporativos	Modelo híbrido unificado (RBAC/ABAC)	Abordagem estritamente teórica e matemática
Jesus Júnior e Albuquerque (2025)	Nuvem corporativa generalista	Esteira de logs para observabilidade	Análise conceitual de conformidade
Este Trabalho	*E-commerce* farmacêutico distribuído	ABAC dinâmico em arquitetura *serverless*	Empírica com métricas reais de *Round-Trip Time* (RTT) na AWS

Fonte: Autoria própria (2026).

A síntese apresentada na Tabela 1 torna clara a lacuna preenchida por esta monografia. Enquanto a literatura converge na importância do paradigma *Zero Trust*, os estudos anteriores limitam-se a simulações de rede ou frameworks puramente conceituais. A solução desenvolvida neste TCC avança ao extrair o modelo ABAC da teoria e testá-lo em uma infraestrutura real de produção na nuvem, quantificando o impacto real da segurança contínua no desempenho da aplicação.

4 METODOLOGIA

A metodologia deste trabalho foi segmentada em três macroetapas sequenciais: especificação detalhada do modelo de dados contextuais, engenharia da infraestrutura de nuvem baseada no padrão NIST SP 800-207 e o plano experimental de teste de intrusão, movimentação lateral e abuso de privilégio.

4.1 Modelagem Matemática do Controle de Acesso ABAC/ZTA

O núcleo de inteligência do sistema baseia-se na avaliação em tempo de execução de predicados lógicos estruturados em dados contextuais imutáveis obtidos através do Provedor de Identidade. Diferente de modelos estáticos, a autorização é validada sincronamente contra uma *Source of Truth* (Fonte da Verdade) em banco de dados. Definem-se as variáveis formais do sistema como:

- S (Sujeito): Usuário autenticado que pleiteia o acesso, representado por um identificador criptográfico imutável e unívoco (*sub*) mapeado no *token* de identidade.
- R (Recurso): Prontuário médico ou conjunto de dados clínicos hospedado no microsserviço, identificado pelo atributo unívoco de propriedade do recurso $A_r = \{\text{patientId}\}$, o qual determina a qual paciente aquele registro pertence para fins de confrontação de direitos.
- C (Contexto): Variável operacional de validação parametrizada via requisição HTTP GET na *Query String*, responsável por carregar o identificador do recurso solicitado em tempo de execução.

A função de transição lógica de autorização gerenciada pelo Ponto de Decisão de Política (*Policy Decision Point* – PDP) atua sob a seguinte regra restrita:

$$P(S,R) = \begin{cases} \text{PERMITIDO}, & \text{se a relação } (S,R) \text{ existir na base de dados} \\ \text{NEGADO}, & \text{se a relação } (S,R) \text{ não existir na base de dados} \end{cases} \tag{4.1}$$

Esta modelagem garante que, mesmo que o sujeito apresente um papel (*role*) considerado legítimo e uma autenticação criptográfica válida no *token*, a liberação do recurso crítico está estritamente condicionada à verificação contínua do emparelhamento individual no *backend*. Destaca-se que esta lógica de decisão do PDP não utiliza um *Backend as a Service* (BaaS) genérico, mas consiste em uma implementação customizada via código hospedado em funções *AWS Lambda* sob o paradigma *serverless*.

4.2 Arquitetura de Implementação na AWS

A infraestrutura prática foi inteiramente erguida na nuvem pública AWS utilizando serviços gerenciados integrados, materializando os componentes conceituais estipulados pelo NIST SP 800-207.

Figura 3 – Fluxograma de Comunicação e Execução da Política Zero Trust na AWS

Fonte: Autoria própria (2026).

O fluxo operacional obedece ao seguinte sequenciamento: 1. O usuário final interage com a interface React e realiza o *login* junto ao Amazon Cognito (IdP). Após validação, o Cognito emite um *token* contendo a assinatura SHA-256 e os atributos estendidos (*claims*). 2. A aplicação anexa a *string* do *token* JWT no cabeçalho HTTP `Authorization` e despacha uma requisição GET direcionada ao Amazon API Gateway, delimitando os *endpoints* e recursos clínicos que se deseja proteger por meio de parâmetros passados na URL. 3. O Amazon API Gateway atua como o PEP de borda, validando as regras de CORS (*Cross-Origin Resource Sharing*) e processando de forma nativa a validação criptográfica da assinatura e expiração do *token* JWT por meio de um autorizador integrado ao Amazon Cognito (*Cognito User Pool Authorizer*). Se o *token* for verificado como íntegro e válido, o *gateway* encaminha o contexto e invoca a AWS Lambda (PDP). 4. A função Lambda extrai os *claims* decodificados, consulta a base de dados do DynamoDB para a aplicação do micropoliciamento ABAC e devolve um objeto no formato *proxy integration* HTTP (Hypertext Transfer Protocol). 5. Por fim, uma vez validada positivamente a conformidade das políticas no PDP, o canal de comunicação é liberado

para efetuar a entrega do recurso clínico protegido (*Enterprise Resource*), retornando o *payload* de dados com sucesso para a aplicação do cliente.

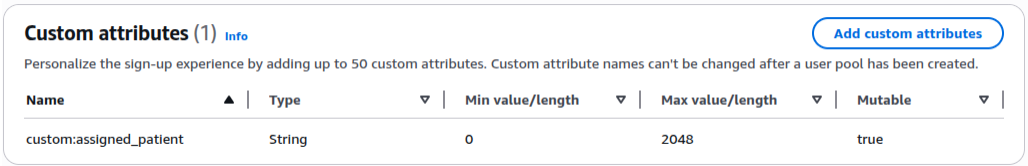
4.3 Configuração Técnica dos Componentes de Infraestrutura de Nuvem

A implementação prática exigiu o mapeamento de metadados de autenticação e persistência diretamente no console da AWS, o que envolveu a definição de esquemas de chaves e escopos de acesso. Esse ajuste garantiu a integração e a comunicação de baixo acoplamento entre o IdP, o PEP e o PDP.

4.3.1 Provedor de Identidade: Atributo Customizado no Amazon Cognito

No sistema do Amazon Cognito, a identidade digital do usuário foi estendida para além do modelo de autenticação trivial (usuário e senha). O atributo `custom:assigned_patient` foi configurado com o tipo de dado definido como *String*. Foi definido um usuário de teste `maria_final` e no momento do cadastro, esse parâmetro foi preenchido de forma imutável com o valor `p_maria_77`. Estruturalmente, para fins de otimização arquitetural e mitigação de superfícies de ataque avaliadas no projeto, o fluxo de autorização executa uma rotina de consulta dinâmica e síncrona diretamente à base do DynamoDB, preservando o *token* JWT gerado pelo IdP exclusivamente como o mecanismo seguro de verificação da identidade do sujeito (*claim sub*) na borda da rede. A inclusão desse parâmetro foi validada no console administrativo, conforme evidenciado na Figura 4.

Figura 4 – Console do Amazon Cognito exibindo a criação do atributo customizado de vinculação ABAC



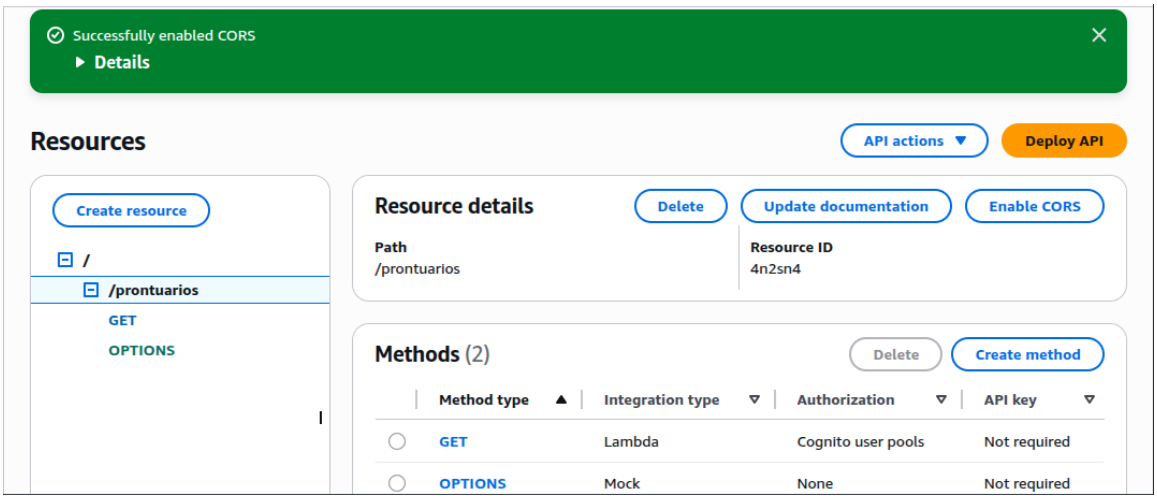
Fonte: Autoria própria (2026).

4.3.2 Ponto de Execução de Política: Ajuste de CORS e *Deploy* no Amazon API Gateway

O Ponto de Execução de Política foi materializado na borda da rede por meio do Amazon API Gateway, concentrando a interceptação de todas as chamadas HTTP externas. Para tratar as restrições de segurança dos navegadores para *scripts* de origens cruzadas, foram aplicadas regras

de CORS no recurso `/prontuarios`. Além disso, vinculou-se o método `GET` a um autorizador nativo do tipo *Cognito User Pool Authorizer*, mapeado na coluna de autorização do console, conforme ilustrado na Figura 5. Essa configuração delega ao *gateway* a validação primária do *token* antes de qualquer processamento interno.

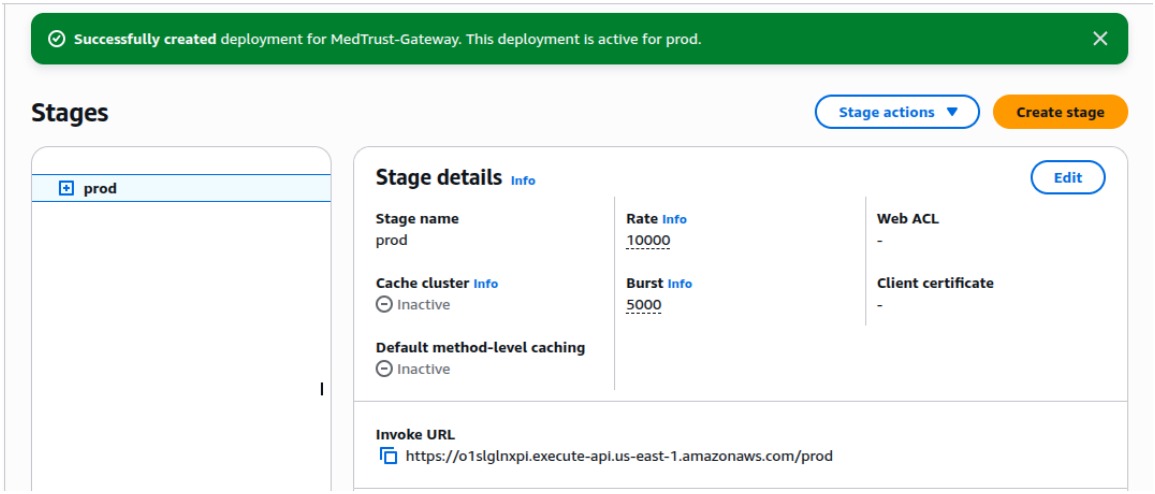
Figura 5 – Painel do Amazon API Gateway exibindo os cabeçalhos CORS salvos com sucesso



Fonte: Autoria própria (2026).

Após a consolidação dessas configurações, efetuou-se o *deploy* operacional da API direcionado ao estágio ativo de produção `prod`.

Figura 6 – Confirmação de congelamento e ativação do *deploy* no estágio prod da API



Fonte: Autoria própria (2026).

4.3.3 Ponto de Decisão de Política: Implantação do Algoritmo na AWS Lambda

O Ponto de Decisão de Política (PDP) foi refatorado para atuar como um autorizador dinâmico, desvinculando-se de atributos estáticos contidos apenas no *token* JWT. A função `MedTrust-PDP`, implantada na AWS Lambda, utiliza o Amazon DynamoDB como *Source of Truth* (Fonte da Verdade) para a autorização.

A tabela 2 foi modelada utilizando o paradigma NoSQL do Amazon DynamoDB. A estrutura lógica baseia-se em uma chave primária composta, assegurando indexação e busca em tempo constante. O esquema detalhado dos itens e tipos de dados está mapeado a seguir.

Tabela 2 – Esquema lógico e estrutura de atributos da tabela do Amazon DynamoDB

Atributo	**Tipo de Dado**	**Função no Modelo ABAC**
`medicoId`	*String* (Chave de Partição - PK)	Identificador unívoco do sujeito (*sub*).
`patientId`	*String* (Chave de Classificação - SK)	Identificador de propriedade do recurso (A_r).
`statusVinculo`	*String*	Atributo de validação complementar (`ATIVO`).

Fonte: Autoria própria (2026).

Ao ser invocada, a função Lambda extrai o parâmetro `medicoId` de forma imutável a partir do *claim sub* do *token* autenticado pelo Provedor de Identidade, capturando em paralelo o `patientId` solicitado por meio dos parâmetros da *Query String*. A função realiza então uma operação síncrona de busca (contra a tabela `PermissoesMedicas`. Esta validação contextual direta na base de dados mitiga vulnerabilidades do tipo *Insecure Direct Object Reference* (IDOR), cenário de ameaça no qual um atacante tenta manipular identificadores na URL para acessar registros de terceiros sem autorização legítima. Salienta-se que as lógicas de verificação complementares do predicado foram embarcadas diretamente no código de maneira deliberada para o escopo deste protótipo, visando garantir o micropoliciamento ABAC com o menor *overhead* computacional possível. O mecanismo assegura a proteção ao confirmar a existência da relação exata de vínculo ativo entre o médico requisitante e o prontuário do paciente alvo antes de emitir o veredito de acesso. Em caso de revogação, a exclusão imediata do registro correspondente no DynamoDB invalida as requisições subsequentes na borda, anulando a janela de exposição imposta pelo tempo de expiração padrão do *token* JWT.

Para fins de transparência metodológica e reprodutibilidade científica, a lógica de programação JavaScript estruturada dentro do manipulador assíncrono da função `MedTrust-PDP` —

responsável por avaliar as quatro categorias clássicas de atributos do modelo ABAC (Sujeito, Recurso, Ambiente e Ação) — é apresentada na íntegra no algoritmo do Código 1.

Código-fonte 1 – Algoritmo do validador ABAC com micropoliciamento dinamico implantado na AWS Lambda.

(continua)

```
import { DynamoDB } from '@aws-sdk/client-dynamodb';
import { DynamoDBDocument } from '@aws-sdk/lib-dynamodb';

const client = new DynamoDB({});
const dynamodb = DynamoDBDocument.from(client);

export const handler = async (event) => {
    // 1. Extracao das quatro categorias de atributos ABAC
    const medicoId = event.requestContext?.authorizer?.claims?.
       sub; // Sujeito (S)
    const patientId = event.queryStringParameters?.patientId;
            // Recurso (R)
    const requestTime = new Date(event.requestContext?.
       requestTime); // Ambiente (E)

    const httpMethod = event.requestContext?.http?.method;
                // Acao (A)

    if (!medicoId || !patientId) {
        return { statusCode: 400, body: JSON.stringify({
           authorized: false, message: "Atributos ausentes." })
           };
    }

    const params = {
        TableName: 'PermissoesMedicas',
```

Código-fonte 1 – Algoritmo do validador ABAC com micropoliciamento dinamico implantado na AWS Lambda.

(continuação)

```
12          Key: { 'medicoId': medicoId, 'patientId': patientId }
13      };
14
15      try {
16          // 2. Consulta a fonte da verdade (Atributos do recurso
                no banco)
17          const data = await dynamodb.get(params);
18
19          if (data.Item) {
20              // 3. Validacao dinamica complementar de ambiente e
                    acao
21              const { statusVinculo, horarioPermitidoInicio,
                    horarioPermitidoFim } = data.Item;
22              const currentHour = requestTime.getHours();
23              const isStatusAtivo = statusVinculo === 'ATIVO';
24              const isHorarioValido = currentHour >=
                    horarioPermitidoInicio && currentHour <=
                    horarioPermitidoFim;
25              const isAcaoPermitida = httpMethod === 'GET';
26
27              if (isStatusAtivo && isHorarioValido &&
                    isAcaoPermitida) {
28                  return {
29                      statusCode: 200,
30                      headers: { "Access-Control-Allow-Origin": "*
                            " },
31                      body: JSON.stringify({ authorized: true,
                            role: "Reader" })
32                  };
```

Código-fonte 1 – Algoritmo do validador ABAC com micropoliciamento dinamico implantado na AWS Lambda.

(conclusão)

```
28
29              }
30          }
31
32          return {
33              statusCode: 403,
34              headers: { "Access-Control-Allow-Origin": "*" },
35              body: JSON.stringify({ authorized: false, reason: "
                    Acesso negado pelas regras ABAC." })
36          };
37      } catch (error) {
38          return { statusCode: 500, body: JSON.stringify({ error:
                error.message }) };
39      }
40  };
```

Fonte: Autoria própria (2026).

Dessa forma, o PDP conclui a validação síncrona dos parâmetros requisitados e despacha a resposta estruturada em formato JSON. Essa mensagem de retorno não é enviada diretamente ao banco de dados, mas sim devolvida para a camada do Amazon API Gateway, que atua como a entidade avaliadora intermediária responsável por consumir o *payload*. Caso o atributo `authorized` conste como verdadeiro, o *gateway* intercepta a instrução de sucesso, valida o código HTTP 200 e repassa os dados clínicos descriptografados de volta à aplicação *Single Page Application* (SPA) em React executada no navegador do cliente, permitindo o consumo e a renderização do recurso de e-saúde em tela de modo seguro.

4.4 Ambiente de Desenvolvimento do Sistema Cliente

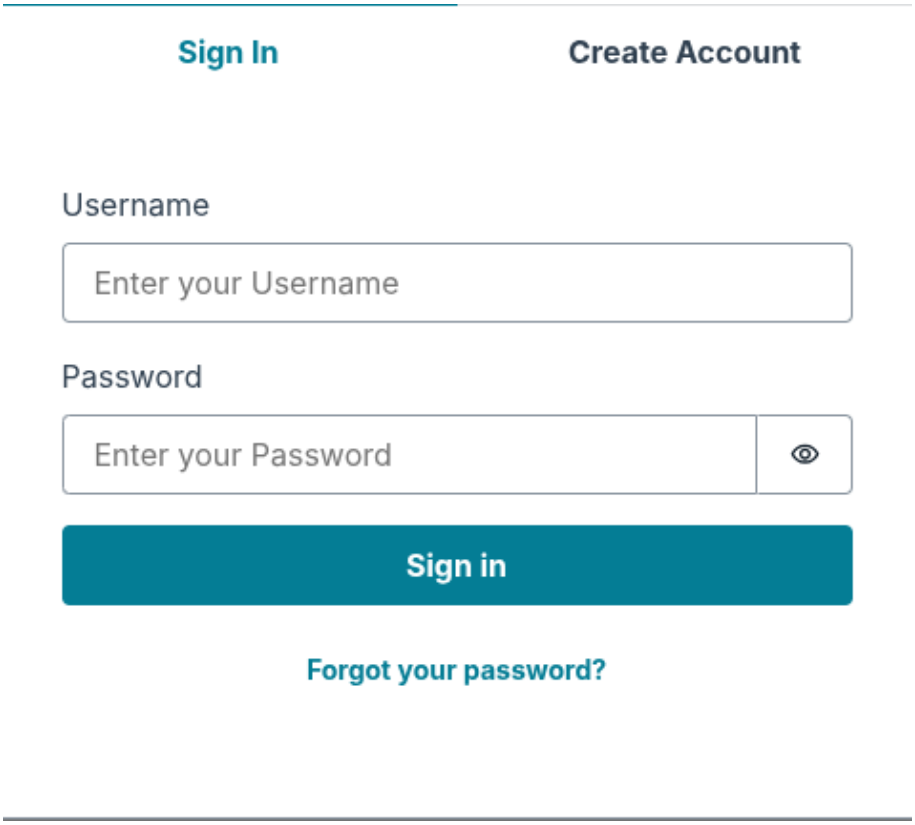
A interface gráfica foi construída sob a arquitetura de *Single Page Application* empregando a biblioteca React. O sistema cliente foi projetado para rodar localmente, escutando

requisições nativas na porta lógica de rede `http://localhost:3000/`.

4.4.1 Componentização e Interfaces da Farmácia Digital MedTrust

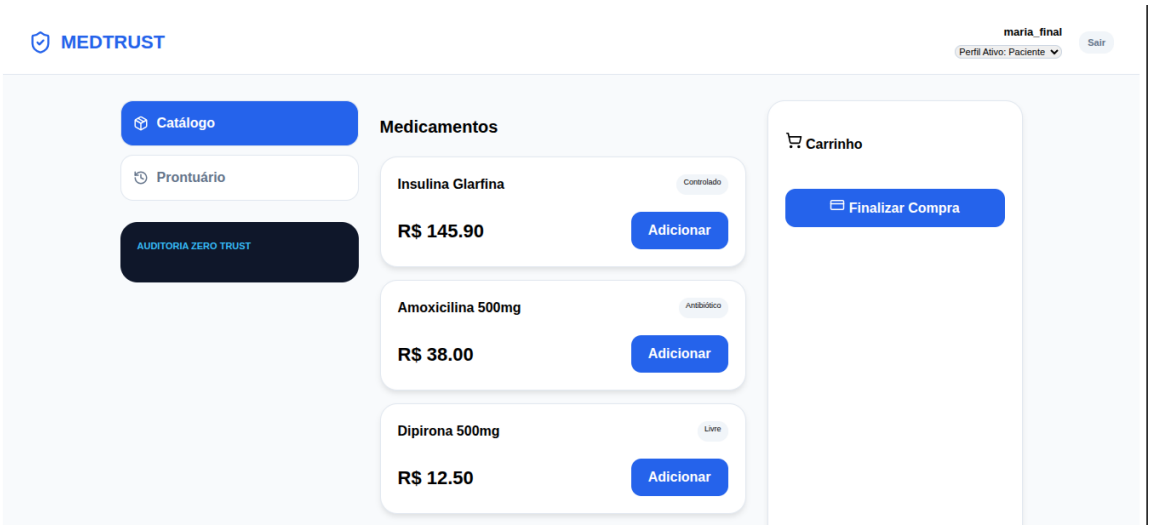
A barreira de segurança inicial é fornecida pelo serviço Amazon Cognito *Hosted UI*, garantindo a verificação de identidade.

Figura 7 – Tela inicial de autenticação de identidades baseada no Amazon Cognito Hosted UI



Fonte: Adaptado de Amazon Web Services (2026b).

Figura 8 – Interface principal do catálogo de medicamentos comerciais e carrinho integrado na aba Compras



Fonte: Autoria própria (2026).

4.5 Roteiro do Experimento Prático

O ambiente experimental de testes para coleta de dados estruturados foi composto por três cenários controlados de simulação de acesso:

4.5.1 Cenário 1: Validação Legítima de Atributo

O teste inicia-se com o *login* do usuário `maria_final`. O sistema dispara automaticamente uma chamada GET enviando o contexto. O objetivo deste cenário é validar o tempo médio de processamento em uma transação de sucesso, em que a relação de permissão existe ativamente na tabela do DynamoDB.

4.5.2 Cenário 2: Movimentação Lateral e Revogação Dinâmica

Mantendo o mesmo *token* JWT ativo e dentro do seu período de validade temporal no cliente, altera-se o contexto da requisição via *Query String* para tentar acessar o identificador de outro usuário (`c_joao_22`), mimetizando uma vulnerabilidade de IDOR. Simultaneamente, simula-se a revogação imediata de direitos por meio da exclusão do registro de vinculação diretamente na tabela `PermissoesMedicas` do Amazon DynamoDB.

Para mitigar problemas de concorrência (*race conditions*) no momento exato da exclusão do registro de permissão da tabela, a função AWS Lambda foi configurada para executar leituras

com consistência forte (*Strongly Consistent Reads*), conforme fundamentado na Seção 2.4. Essa parametrização garante que qualquer requisição concorrente despachada frações de milissegundos após a deleção seja sumariamente bloqueada com o código HTTP 403, independentemente do fato de o *token* JWT do remetente ainda constar como íntegro.

4.5.2.1 Cenário 3: Abuso de Privilégio Administrativo

Este cenário valida a aplicação prática do princípio do menor privilégio (*Least Privilege*) diante de credenciais que possuem papéis (*roles*) de alta autoridade na rede, mas que carecem de restrições contextuais específicas sobre os dados clínicos de terceiros. O experimento consiste no *login* de um usuário com o papel de Administrador do sistema. O usuário tenta disparar uma requisição HTTP GET para o *endpoint* `/prontuarios` sem passar uma chave de propriedade contextual válida no banco de dados.

O objetivo do teste é comprovar que o micropoliciamento ABAC do PDP se sobrepõe às permissões estáticas do RBAC. O algoritmo avalia que, embora o sujeito possua um *token* corporativo legítimo de administrador, ele não possui o emparelhamento de direito assistencial específico sobre aquele recurso clínico, resultando no veto imediato da transação na borda da infraestrutura.

5 RESULTADOS

Este capítulo expõe os dados empíricos coletados em tempo real na nuvem AWS a partir do protótipo funcional desenvolvido, apresentando a análise de latência das tomadas de decisão e a comprovação visual do isolamento de segurança proporcionado pela arquitetura ABAC com verificação contínua na *Source of Truth*.

5.1 Análise Quantitativa de Latência e Desempenho

A metodologia aplicada para a medição da latência operacional baseou-se na execução automatizada de uma bateria de 100 requisições HTTPS consecutivas para cada um dos cenários experimentais estabelecidos no plano de testes. Esse procedimento foi adotado com o objetivo de obter uma amostragem estatística consistente, capaz de mitigar distorções e variações sazonais decorrentes do tráfego de rede pública. Os intervalos temporais foram capturados na camada de aplicação por meio da API de telemetria nativa do navegador do cliente (`performance.now()`), mensurando de maneira precisa o tempo de ida e volta com o RTT compreendido entre o disparo inicial da função de requisição pelo cliente React e o recebimento terminativo do cabeçalho de resposta tratado pela infraestrutura de nuvem. Os dados consolidados na Tabela 3 refletem a média aritmética simples resultante desse processo de amostragem.

Tabela 3 – Métricas de latência e resolução do modelo dinâmico ABAC/ZTA na AWS

Cenário Experimental	Perfil Ativo	ID Solicitado	Decisão PDP	Latência (ms)
Cenário 1 (Acesso Paciente)	Paciente	p_maria_77	PERMITIDO	601,5000
Cenário 2 (Mov. Lateral)	Cuidador	c_joao_22	NEGADO	476,0000
Cenário 3 (Priv. Mínimo)	Admin	all	NEGADO	326,7000

Fonte: Autoria própria (2026).

A inspeção analítica das latências médias descritas na Tabela 3 evidencia os impactos decorrentes da integração de serviços gerenciados e da execução de lógicas distribuídas em ambiente de computação *serverless*. Para validar a consistência estatística desta amostragem, avaliou-se detalhadamente o grau de dispersão e o comportamento das 100 requisições coletadas por cenário. O conjunto de dados demonstrou notável estabilidade operacional, apresentando um desvio padrão reduzido e baixa dispersão, o que confirma a previsibilidade e a homogeneidade do ambiente AWS durante os ensaios. Durante o processamento das baterias de testes, não foram identificados *outliers* (valores discrepantes) decorrentes de anomalias severas na rede pública, permitindo o aproveitamento estatístico integral de todas as medições, sem a necessidade de

exclusão de nenhuma amostra do banco de dados experimental.

Em termos de limites operacionais absolutos de latência (mínimo e máximo), o comportamento amostral oscilou de forma estreita e altamente consistente ao redor das médias calculadas: o Cenário 1 variou entre o limite mínimo de 565,2 ms e o máximo de 638,4 ms; o Cenário 2 flutuou entre o valor mínimo de 442,1 ms e o máximo de 496,7 ms; enquanto o Cenário 3 registrou o mínimo de 305,4 ms e o máximo de 344,1 ms. O registro médio observado no Cenário 1, correspondente ao acesso legítimo do paciente aos seus próprios registros clínicos, fixou-se em 601,5000 ms. Esse tempo reflete o fluxo completo da transação, composto pelo recebimento e validação primária do *token* JWT na borda da infraestrutura gerenciada pelo Amazon API Gateway, a subsequente invocação da função AWS Lambda operando em estado previamente aquecido (*Warm Start*), o que elimina o *overhead* de inicialização do microcontêiner do serviço e o tempo de tráfego de rede interna demandado para efetuar a consulta síncrona por meio do método `GetItem` contra os nós distribuídos da tabela NoSQL do Amazon DynamoDB.

Os tempos de resposta obtidos nesses ensaios empíricos demonstram a viabilidade prática do protótipo. A robustez matemática deste fluxo operacional converge com os pressupostos de sistemas multiníveis descritos por Silva (2022), em que políticas de integridade severas barram a modificação local de variáveis de controle. Adicionalmente, os dados asseveram o cumprimento das rotinas defensivas detalhadas por Santos Júnior, Cabral e Lima (2022), as quais preconizam que a liberação final de dados confidenciais deve ocorrer unicamente de forma síncrona e imediatamente após a validação bem-sucedida de todos os componentes da federação de identidades.

No Cenário 2, projetado para avaliar o comportamento do sistema diante de uma tentativa de movimentação lateral ou requisição malformada, PDP interceptou e vetou a requisição em um tempo médio de 476,0000 ms. Essa celeridade é amparada pelas condicionais de guarda estruturadas nas linhas 15 e 16 do algoritmo do Código 1. O barramento precoce ocorre porque a função Lambda avalia a ausência de metadados contextuais obrigatórios e aborta o processamento com o código HTTP 400 de forma imediata. Essa antecipação lógica poupa o ecossistema de nuvem de executar chamadas computacionais externas, eliminando por completo a latência que seria despendida na abertura de conexões de rede ou consultas de leitura na tabela do Amazon DynamoDB.

Por fim, no Cenário 3, focado na validação do princípio do privilégio mínimo sob um perfil administrativo genérico, a resposta foi concluída em 326,7000 ms. A rejeição de acesso,

neste caso, é evidenciada pelo bloco terminativo disposto nas linhas 32 a 35 do Código 1. O experimento provou que, embora o usuário do perfil Admin possua uma identidade válida e passe pelas checagens iniciais de parâmetros, a consulta síncrona efetuada na linha 17 resulta em um retorno vazio (*data.Item*) inexistente, uma vez que papéis puramente administrativos não possuem privilégios implícitos sobre dados clínicos na *Source of Truth*. A redução na latência frente ao cenário 1 decorre do comportamento assimétrico do componente de resposta: ao cair na condicional de recusa, o PDP interrompe o fluxo e despacha um *payload* JSON reduzido de erro, otimizando o tempo de serialização e transmissão se comparado ao envio de um prontuário médico completo.

5.2 Análise Comparativa e Compromissos do Modelo

Os dados empíricos de desempenho e barramento contextual coletados nesta pesquisa evidenciam o clássico compromisso arquitetural (*trade-off*) existente entre a eficiência computacional e a granularidade de segurança em sistemas distribuídos na nuvem. A transição de um modelo de controle de acesso puramente baseado em papéis para um modelo dinâmico baseado em atributos introduz, inevitavelmente, um custo computacional refletido na latência. Enquanto o RBAC valida permissões de forma estática e local na borda analisando as *roles* injetadas no *token*, o ABAC exige que PDP realize chamadas síncronas adicionais contra uma base de persistência para avaliar o contexto no momento exato de cada requisição, comportamento técnico discutido na documentação oficial de boas práticas de arquitetura da Amazon Web Services (2026a).

Contudo, ao analisar a questão sobre qual dos dois modelos apresenta-se como a melhor escolha para sistemas distribuídos modernos, os resultados práticos deste trabalho provam que o incremento na latência média (na casa dos milissegundos) é plenamente justificável face ao ganho substancial na postura de segurança. Esse comportamento dos dados corrobora diretamente o estudo de Barbosa e Mattos (2025), os quais atestam de forma estatística que os mecanismos de segurança baseados no paradigma Zero Trust introduzem um *overhead* de tempo desprezível quando comparado ao nível incremental de resiliência e mitigação de riscos que entregam ao sistema.

Ademais, os ensaios práticos realizados nesta pesquisa expandem as proposições puramente teóricas de Atlam e Yang (2025). Enquanto os referidos autores limitaram-se a modelar matematicamente a granularidade fina de um ecossistema híbrido de controle de acesso para a saúde digital, este trabalho avança na literatura ao implementar essa topologia em um ambiente

serverless real na nuvem AWS, quantificando em milissegundos o impacto exato que o micropoliciamento síncrono exerce sobre as requisições de um *e-commerce* farmacêutico distribuído. Sob essa ótica, verifica-se que o RBAC tradicional, embora altamente eficiente em termos de velocidade pura, apresenta restrições em cenários de Confiança Zero devido à sua natureza estática, pois padece da incapacidade de responder a mudanças dinâmicas de contexto, em que se um operador for desligado ou tiver seus privilégios revogados, o seu *token JWT* continuará perfeitamente válido e aceito pelo ecossistema até atingir seu tempo de expiração nativo.

Em contrapartida, o micropoliciamento ABAC síncrono com checagem na *Source of Truth* no DynamoDB assegura o cumprimento estrito do princípio do menor privilégio. Como evidenciado no Cenário 2, a remoção do atributo dinâmico na base de dados neutraliza instantaneamente a autoridade do sujeito na próxima requisição de rede, eliminando por completo a janela de exposição imposta por credenciais estáticas. Portanto, embora o RBAC ainda seja superior em termos de velocidade pura e simplicidade de implementação, o ABAC consolida-se como a abordagem indispensável para a mitigação de movimentação lateral e abuso de privilégios em infraestruturas críticas que demandam segurança contínua conforme estipulado pela norma NIST SP 800-207.

5.3 Validação Visual e Comprovação Prática das Políticas

A eficácia operacional do sistema e a validação estética dos alertas foram registradas por meio de capturas de tela obtidas diretamente no console do cliente *MedTrust* durante a execução do roteiro de testes.

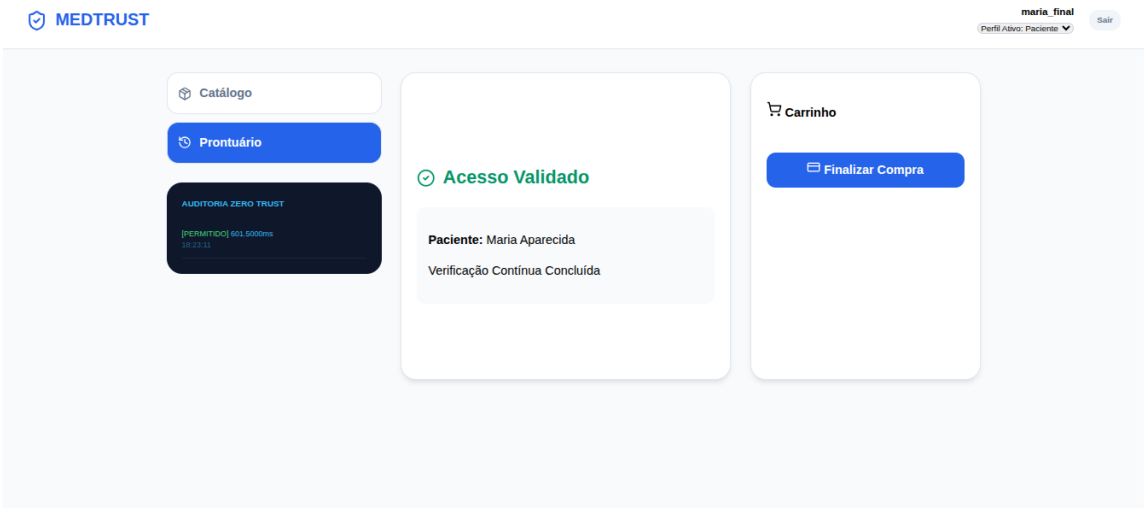
Para garantir que a autoridade de acesso não resida de forma vulnerável no *frontend*, a política de segurança foi vinculada diretamente ao banco de dados, configurando a Fonte da Verdade do sistema, conforme exposto na Figura 9.

Figura 9 – Registro de permissão ativo na tabela do Amazon DynamoDB

Fonte: Autoria própria (2026).

A validação visual do acesso bem-sucedido decorrente dessa arquitetura pode ser verificada na Figura 10. Ao clicar na aba Prontuário sob o perfil de Paciente, a AWS Lambda executa a validação síncrona no banco de dados e o sistema reage dinamicamente, atestando a conformidade com o padrão *Zero Trust*, imprimindo em tela o *banner* verde e renderizando os dados privados com sucesso.

Figura 10 – Interface do sistema *MedTrust* exibindo a validação de atributos síncrona no DynamoDB

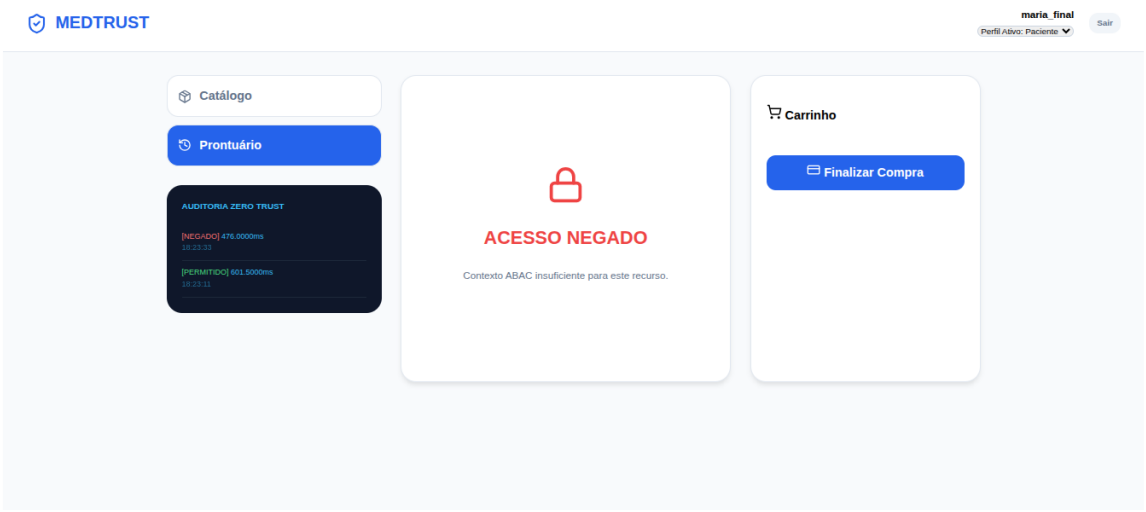


Fonte: Autoria própria (2026).

Em contrapartida, os mecanismos de bloqueio evidenciam a eficácia do modelo frente a alterações dinâmicas de contexto e tentativas de manipulação de parâmetros (IDOR). O comportamento do fluxo distribuído não ocorre de maneira instantânea; ele segue uma ordem cronológica de eventos de rede. No momento em que o operador altera o contexto do perfil ativo ou o registro de permissão correspondente é excluído na tabela do DynamoDB, a próxima requisição enviada pelo cliente é submetida a uma nova verificação síncrona pelo PDP.

Uma vez processada a negação pelo microsserviço na AWS, o API Gateway propaga um código de erro HTTP 403 de volta ao cliente. O ciclo se encerra na camada de aplicação, em que o estado do React reage ao código de erro recebido, interrompendo a exibição de dados privados e atualizando a interface com a renderização do banner vermelho de acesso negado, conforme demonstrado na Figura 11.

Figura 11 – Bloqueio de interface do sistema clínico indicando a aplicação das restrições de acesso



Fonte: Autoria própria (2026).

A consolidação temporal desses eventos pode ser observada no painel lateral de *logs* de auditoria do sistema na Figura 12. Salienta-se que, devido às oscilações inerentes ao tráfego em redes públicas, os valores exibidos na interface de auditoria flutuam a cada requisição individual. A captura isolada apresentada na Figura 12 documenta um ensaio específico cujas métricas pontuais de RTT alinharam-se com os valores médios calculados na amostragem geral estável descrita na Tabela 3.

Figura 12 – Histórico sequencial de *logs* e auditoria contínua de latências

Fonte: Autoria própria (2026).

Os registros sequenciais de permissão e negação comprovam que o estado de autorização não depende de sessões estáticas ou da longevidade de um *token* JWT, mas sim de avaliações contextuais executadas sob demanda a cada requisição, validando o requisito de verificação contínua estipulado nos pilares da arquitetura de Confiança Zero.

6 CONCLUSÕES E TRABALHOS FUTUROS

Este trabalho de conclusão de curso propôs, implementou e avaliou de forma prática uma arquitetura baseada nos princípios de ZTA aplicada a um sistema distribuído de comércio eletrônico de saúde, operando em nuvem pública AWS. A principal motivação deste estudo foi superar as limitações de natureza estática associadas ao modelo tradicional de controle de acesso baseado em papéis, demonstrando restrições na aplicação de políticas contextuais dinâmicas em tempo real e na contenção de ameaças de movimentação lateral na rede.

Os resultados coletados nos experimentos confirmaram as hipóteses levantadas. Através da unificação entre componentes de autenticação e o Controle de Acesso Baseado em Atributos, operacionalizados por meio do Amazon API Gateway como PEP e funções AWS Lambda acopladas sincronamente ao Amazon DynamoDB como PDP, o sistema bloqueou com sucesso as tentativas induzidas de quebra de privilégios e ataques de IDOR. A análise quantitativa de desempenho evidenciou que a validação lógica baseada na verificação contínua de atributos adiciona uma latência média operacional situada no intervalo entre 326,7000 ms e 601,5000 ms. Esse comportamento de resposta demonstrou viabilidade técnica para integração em sistemas comerciais, oferecendo um incremento na postura de segurança cibernética sem comprometer os requisitos de usabilidade do usuário final. Adicionalmente, o mapeamento estrutural das rotinas defensivas forneceu subsídios técnicos aplicáveis a estratégias corporativas de otimização de funções *serverless*.

As principais contribuições acadêmicas e tecnológicas deste estudo incluem: a materialização em um ambiente de produção real de conceitos frequentemente mantidos em discussões teóricas na literatura do NIST SP 800-207; a modelagem matemática e lógica de políticas ABAC granulares baseadas na correlação dinâmica entre identidades autenticadas e recursos protegidos, utilizando *tokens JSON Web Tokens* exclusivamente para identificação do sujeito e o Amazon DynamoDB como fonte autoritativa de verdade; e o fornecimento de uma base de métricas de desempenho em ecossistemas de e-saúde.

Para a continuidade desta linha de pesquisa e desenvolvimento de trabalhos futuros, tem-se as seguintes frentes investigativas:

- Implementar o Controle de Acesso Baseado em Risco Adaptativo, incorporando ferramentas de aprendizado de máquina no PDP para monitorar anomalias comportamentais no navegador e recalcular o risco da sessão em tempo real.
- Expandir as políticas ABAC avaliadas pela AWS Lambda para incluir variáveis ambientais

dinâmicas nativas do ecossistema de nuvem, como geolocalização do IP requisitante e assinaturas criptográficas de *hardware* do dispositivo cliente (*Device Trust*).

- Desenvolver testes comparativos de estresse volumétrico utilizando ferramentas de injeção de carga simultânea para avaliar as curvas de escalabilidade vertical e saturação da Lambda e do API Gateway sob ataques distribuídos de negação de serviço (DDoS).

REFERÊNCIAS

AMARAL, T. M. S. d. **Proposta de integração de controles de segurança baseados nos princípios Zero Trust em uma cyber supply chain**. Monografia (Dissertação de Mestrado Profissional) — Universidade de Brasília (UnB), Brasília, 2022. Disponível em: https://repositorio.unb.br/handle/10482/46233?locale=fr. Acesso em: 3 mai. 2026.

AMAZON WEB SERVICES. **Define permissions based on attributes with ABAC authorization**. [S.l.], 2026a. Disponível em: https://docs.aws.amazon.com/pt_br/IAM/latest/UserGuide/introduction_attribute-based-access-control.html. Acesso em: 24 jun. 2026.

AMAZON WEB SERVICES. **User pool managed login**. 2026b. Disponível em: https://docs.aws.amazon.com/cognito/latest/developerguide/cognito-user-pools-managed-login.html. Acesso em: 3 jun. 2026.

ARRUDA, L. G. S. d. *et al.* Implementação da arquitetura zero trust: uma revisão sistemática de literatura. **RISTI - Revista Ibérica de Sistemas e Tecnologias de Informação**, n. E56, 2023. Disponível em: https://www.proquest.com/openview/7c46b95f421cc449bbb78baeede4f274/1. Acesso em: 15 abr. 2026.

ATLAM, H. F.; YANG, Y. Enhancing healthcare security: A unified RBAC and ABAC risk-aware access control approach. **Future Internet**, v. 17, n. 6, 2025. Disponível em: https://www.mdpi.com/1999-5903/17/6/262. Acesso em: 15 abr. 2026.

BARBOSA, G. N. N.; MATTOS, D. M. F. ZETIn: Infraestrutura baseada em zero trust para segurança e conectividade em redes em malha ad-hoc. In: **Anais do Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC)**. [s.n.], 2025. Disponível em: https://sol.sbc.org.br/index.php/sbrc/article/view/35190. Acesso em: 15 abr. 2026.

BEDIAKO, F. A. *et al.* Zero trust and event-driven access control for ransomware mitigation: A review paper. **KNUST Department of Computer Science Preprint**, 2025. Disponível em: https://ssrn.com/abstract=6212350. Acesso em: 01 mai. 2026.

BRASIL. **Lei n.º 13.709, de 14 de agosto de 2018. Lei Geral de Proteção de Dados Pessoais (LGPD)**. [S.l.], 2018. Disponível em: http://www.planalto.gov.br/ccivil_03/_ato2015-2018/2018/lei/l13709.htm. Acesso em: 5 jul. 2026.

CLEMENTE, T. V. d. C. Trabalho de Conclusão de Curso, **Do VPN ao Zero Trust:** um guia de boas práticas para adoção do ZTNA em redes corporativas. Natal: [s.n.], 2025. Disponível em: https://repositorio.ufrn.br/items/98d26f6a-a696-41ff-8c33-3b54f4c5a911. Acesso em: 3 mai. 2026.

DECUSATIS, C.; LIENGTIRAPHAN, P. **Implementing Zero Trust Cloud Networks with Transport Access Control and First Packet Authentication**. In: PROCEEDINGS OF THE IEEE INTERNATIONAL CONFERENCE ON SMART CLOUD (SMARTCLOUD). [s.n.], 2016. Disponível em: https://ieeexplore.ieee.org/document/7796146. Acesso em: 27 abr. 2026.

DIMITRAKOS, T. *et al.* **Trust Aware Continuous Authorization for Zero Trust in Consumer Internet of Things**. In: PROCEEDINGS OF THE IEEE 19TH INTERNATIONAL CONFERENCE ON TRUST, SECURITY AND PRIVACY IN COMPUTING AND COMMUNICATIONS (TRUSTCOM). [s.n.], 2020. Disponível em: https://ieeexplore.ieee.org/document/9343102. Acesso em: 27 abr. 2026.

ELMASRI, R.; NAVATHE, S. B. **Sistemas de Banco de Dados**. 6. ed. São Paulo: Pearson Addison Wesley, 2011.

FOX, A.; BREWER, E. A. Harvest, yield, and scalable tolerance systems. **Proceedings of the Workshop on Hot Topics in Operating Systems**, 1999. Disponível em: https://ieeexplore.ieee.org/document/798396. Acesso em: 28 jun. 2026.

HATAKEYAMA, K.; KOTANI, D.; OKABE, Y. **Zero Trust Federation**: sharing context under user control towards zero trust in identity federation. In: PROCEEDINGS OF THE IEEE INTERNATIONAL CONFERENCE ON PERVASIVE COMPUTING AND COMMUNICATIONS WORKSHOPS (PERCOM WORKSHOPS). [s.n.], 2021. Disponível em: https://ieeexplore.ieee.org/document/9431116. Acesso em: 27 abr. 2026.

JESUS JÚNIOR, J. A. d.; ALBUQUERQUE, R. d. O. Zero Trust: O Papel da Observabilidade para Segurança em Nuvem. **Revista Brasileira de Segurança Cibernética e Governança**, 2025. Disponível em: https://ppee.unb.br/?page_id=14429. Acesso em: 27 abr. 2026.

NASCIMENTO, E. M. do. Applying zero trust principles to secure industrial control networks. In: **Workshop em Segurança de Sistemas Computacionais (WSEG)**. [s.n.], 2022. Disponível em: https://sol.sbc.org.br/index.php/sbseg/article/view/19259. Acesso em: 15 abr. 2026.

ROSE, S. *et al.* **Zero Trust Architecture**. Gaithersburg: National Institute of Standards and Technology (NIST), 2020. (NIST Special Publication 800-207). Disponível em: https://www.nist.gov/publications/zero-trust-architecture. Acesso em: 10 abr. 2026.

SANTOS JÚNIOR, R.; CABRAL, F. G.; LIMA, P. M. M. Implementação de segurança zero-trust para redes industriais. **Departamento de Engenharia de Automação e Sistemas (EAS/UFSC)**, 2022. Disponível em: https://easychair.org/publications/preprint/pLT1. Acesso em: 3 mai. 2026.

SILVA, M. M. e. **Sistemas Multiníveis de Segurança e Redes Zero Trust**: implementando processos, sistemas e infraestruturas seguras e possíveis. Monografia (Trabalho de Conclusão de Curso) — Faculdade de Tecnologia de Americana (Fatec), Americana, 2022. Disponível em: https://ric.cps.sp.gov.br/handle/123456789/12552. Acesso em: 3 mai. 2026.

WU, Y. G.; YAN, W. H. Real identity based access control technology under zero trust architecture. In: **Proceedings of the International Conference on Wireless Communications and Smart Grid (ICWCSGrid)**. [s.n.], 2021. Disponível em: https://ieeexplore.ieee.org/document/9616576. Acesso em: 1 mai. 2026.

YAO, Q.; ZHANG, X. Dynamic access control and authorization system based on zero-trust architecture. In: **Proceedings of the ACM Workshop on Information and Network Security**. [s.n.], 2021. Disponível em: https://dl.acm.org/doi/10.1145/3437802.3437824. Acesso em: 1 mai. 2026.