

Visão e Escopo

Versão 1.3

22 de setembro de 2025

Preparado para

Hakai Security – Score Trust

NEW GROUP LABS

Índice

ÍNDICE.....	1
1. SOBRE ESTE DOCUMENTO	2
1.1. IDENTIFICAÇÃO DO PROJETO.....	2
1.2. INTEGRANTES	2
1.3. INTRODUÇÃO.....	3
2. VISÃO GERAL DO PROJETO.....	4
2.1. HAKAI SECURITY	4
2.2. MOTIVAÇÃO DO PROJETO	4
2.3. PROPOSTA.....	4
2.4. VISÃO.....	5
3. ROADMAP	6
3.1. VERSÃO ATUAL	6
4. PROVA DE CONCEITO (POC)	6
4.1. CENÁRIO NEXSHOP	6
4.2. SOLUÇÃO PROPOSTA INICIAL.....	7
4.3. STACK DE DESENVOLVIMENTO DA FASE 1	7
4.4. DEMO DA SOLUÇÃO – FASE 1	8
5. IMPLEMENTAÇÃO DA SOLUÇÃO	13
5.1. STACK DE DESENVOLVIMENTO DA FASE 2	13
5.2. DEMO DA SOLUÇÃO – FASE 2.....	14
6. IMPLEMENTAÇÃO AVANÇADA DA SOLUÇÃO.....	20
6.1. STACK DE DESENVOLVIMENTO DA FASE 3	20
6.2. DEMO DA SOLUÇÃO – FASE 3.....	21
6.2.1 DASHBOARD.....	25
6.2.2 TESTES DE REQUISIÇÃO NO INSOMNIA.....	29
6.2.3 ATUALIZAÇÕES DE SEGURANÇA	32
6.2.4 DOCUMENTAÇÃO	33
7. IMPLEMENTAÇÃO FINAL DA SOLUÇÃO.....	35
7.1. STACK DE DESENVOLVIMENTO DA FASE 4	35
7.2. DEMO DA SOLUÇÃO – FASE 4.....	37
8. CONCLUSÃO.....	43

NEW GROUP LABS

1. Sobre este documento

1.1. IDENTIFICAÇÃO DO PROJETO

Nome do projeto: Score Trust

Cliente: Hakai Security

Faculdade: FIAP

Curso: Defesa Cibernética

Turma: 2TDCOA - 2025/2

Documentação: <https://github.com/luizpessol/score-trust>

Vídeo da Fase 1: <https://www.youtube.com/watch?v=UlpHxzTAgaW>

Vídeo da Fase 2: <https://www.youtube.com/watch?v=R4Elh9VUHic>

Vídeo da Fase 3: https://www.youtube.com/watch?v=yDG2NvRn_Vs

Vídeo da Fase 4: <https://www.youtube.com/watch?v=2RpAKV2BBzw>

1.2. INTEGRANTES

Tabela 1 – Integrantes

RM	Nome Completo
RM558027	Adrian da Silva Wicke
RM555049	Ana Carolina Araujo Paro
RM557331	Camille Alencastro Medina
RM556606	Demétrio Rodrigo Paszko
RM554505	Luiz Alberto dos Santos Pessol

1.3. INTRODUÇÃO

Nos últimos anos, a segurança dos espaços cibernéticos tornou-se fator decisivo ao determinar um fornecedor, um cliente ou até mesmo um funcionário, visto que zelar o patrimônio digital de uma empresa garante que as receitas poderão manter-se saudáveis — principalmente por sanções no descumprimento de leis de segurança da informação ou extorsões causadas por atores de ameaças.

A partir dessa visão, a empresa Hakai Security, renomada empresa no setor de cibersegurança, apresentou uma desafiadora proposta para as equipes de estudantes do segundo ano do curso de Defesa Cibernética, buscando incentivar a interpretação e a compreensão de sistemas antifraude nos e-commerces que tanto são utilizados no dia a dia para proteger seus consumidores e informações confidenciais.

Neste primeiro relatório da equipe New Group Labs, a equipe apresenta os primeiros passos do seu projeto *Score Trust* e a implementação inicial de um SDK (Software Development Kit) para avaliar a autenticidade do acesso de um usuário em uma loja virtual com base em um sistema de scores.

NEW GROUP LABS

2. Visão geral do projeto

2.1. HAKAI SECURITY

A Hakai Security é uma empresa focada em cibersegurança ofensiva e consultoria, oferecendo testes de penetração e segurança, de exploração de vulnerabilidades, simulações de ataque e de engenharia social.

2.2. MOTIVAÇÃO DO PROJETO

No ano de 2024, a empresa ClearSale, especializada em soluções antifraude e score de crédito para lojas virtuais, mercados financeiros e outros setores, reportou que o Brasil teve um registro de mais de 2,8 milhões de tentativas de fraude em e-commerces. O número foi menor que em 2023, mas atores de ameaça optaram por fraudar produtos mais caros, o que prejudicou a reputação de inúmeras empresas e debilitou a confiança dos clientes nas marcas.

Em um cenário como o que foi apresentado, fraudes podem ocorrer através de diferentes formas, como vulnerabilidades no código, clonagem de cartão, esquemas robustos, vazamento de dados e acesso indevido de terceiros. A partir de um acesso não autorizado em um sistema debilitado, um fraudador pode alterar informações do perfil, capturar endereços postais, realizar compras indevidas e inclusive utilizar cartões de crédito roubados, “terceirizando” a culpa.

2.3. PROPOSTA

Em virtude dos dados evidenciados, a equipe New Group Labs buscou aplicar os fatos conhecidos e compreender a proposta anunciada pela Hakai Security, onde o principal objetivo era desenvolver um Software Development Kit (SDK) para avaliar o usuário no contexto antifraude, auxiliando as aplicações de e-commerce a validarem a identidade legítima dos clientes no momento do login, checkout ou em ações sensíveis.

Para o pleno funcionamento da solução, o SDK deveria ser capaz de coletar dados proveitosos para gerar um score de confiança e permitir que o sistema agisse de acordo com

NEW GROUP LABS

SDK para avaliar o acesso do usuário. No contexto técnico, a Hakai apresentou o caso da empresa NexShop que, com o crescimento do seu negócio e clientes, passou a enfrentar tentativas de fraude digital em diversas etapas da jornada de compra, principalmente no login e no checkout.

A empresa estava optando por uma solução leve, buscando um SDK que fosse modular e reutilizável, possível de integrar em qualquer aplicação web moderna; plugável tanto no frontend quanto no backend; com documentação de uso simples; fácil de instalar e com capacidade de capturar dados úteis do cliente.

2.4. VISÃO

Esta solução contribui de forma significativa na visão da equipe de cibersegurança da Hakai Security e do New Group Labs, visando:

- O desenvolvimento de um sistema de pontuação de fraude, buscando aplicar conceitos de cibersegurança e programação para garantir a eficiência da solução;
- A elaboração de um projeto que poderá ser utilizado para portfólio pessoal de cada membro;
- A aplicação dos pilares da segurança da informação — confidencialidade, integridade e disponibilidade.

NEW GROUP LABS

3. Roadmap

3.1. VERSÃO ATUAL

Fase 1: PoC com Payload em JSON e Ambiente AWS API Gateway, Lambda e Demo da solução;

Fase 2: Front-end para teste, SDK v1, integração com AbuseIP, DynamoDB e demo da solução;

Fase 3: Front-end (React), SDK v1.1 (JavaScript), Back-end (Python 3.13), AWS (API Gateway, Lambda, DynamoDB, WAF, IAM e CloudWatch, AWS 53, AWS ACM), Abuse IP (<https://www.abuseipdb.com/>) e Documentação (<https://github.com/luizpessol/score-trust>)

Fase 4: Front-end (React, HTML, Vite, TailwindCSS), SDK v1.2 (JavaScript), Back-end (Python 3.13), API Gateway (Face-verify), Lambda (faceVerify), Amazon Rekognition – CompareFaces, Amazon S3 – Armazenamento de fotos e Documentação atualizada.

4. Prova de Conceito (PoC)

4.1. CENÁRIO NEXSHOP

A NexShop, um e-commerce de gadgets e eletrônicos em ascensão, viu seu faturamento crescer 300% no último ano. Contudo, a partir do seu sucesso, a empresa identificou problemas na segurança digital, sendo que nos últimos três meses:

- Estornos por fraudes em contas recém-criadas;
- Criação de perfis falsos com dados de terceiros;
- Compras feitas por meio de dispositivos estranhos, sem que o usuário original tivesse conhecimento;
- Bots que simulavam comportamentos humanos para burlar etapas de verificação.

Contando com uma pequena equipe de segurança, a NexShop busca uma solução simples, leve e inteligente para verificar a integridade das contas dos seus usuários e garantir

NEW GROUP LABS

que os acessos em seu site sejam legítimos, promovendo a segurança dos clientes e mantendo sua reputação intacta.

4.2. SOLUÇÃO PROPOSTA INICIAL

Segundo as informações detalhadas da NexShop e recomendações enviadas pela Hakai Security, a New Group Labs optou pelo desenvolvimento de um Software Development Kit (SDK) para ser integrado ao checkout e login, analisando a validade do acesso de um usuário — mesmo sem autenticação multifator tradicional —, utilizando dados comportamentais, do dispositivo, localização e biometria, garantido o mínimo de fricção para o cliente legítimo.

Para verificar essas informações, o sistema de *Score Points* foi idealizado, atribuindo uma pontuação para cada suspeito que resultando em uma soma, onde de 0 – 30 permite que aquele que está fazendo o login possa acessar a página (ou seja, uma entrada segura), 31 – 75 representa um risco inicial, solicitando uma revisão, e de 76 – 100 que indica um risco completo, bloqueando o acesso.

Figura 1 – Tabela de scores e ação recomendada

SCORE	AÇÃO
0 - 30	Seguro
31 - 75	Revisar
76 - 100	Bloquear

Fonte: Os autores (2025)

4.3. STACK DE DESENVOLVIMENTO DA FASE 1

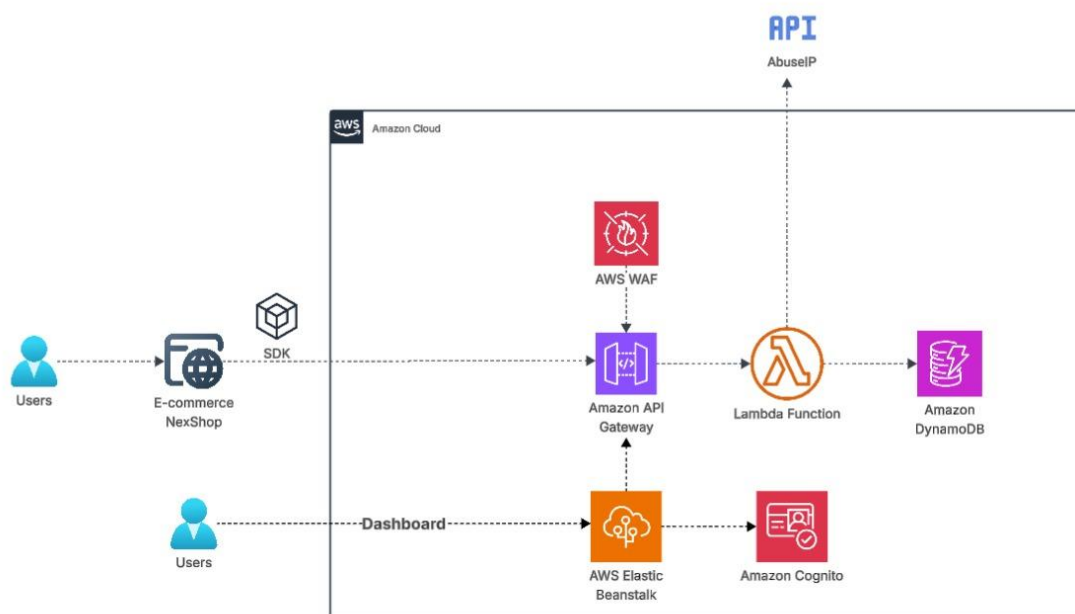
Neste tópico, apresentamos a Stack de Desenvolvimento utilizado na Fase 1, ou seja, o conjunto de ferramentas e linguagens de programação aplicadas para a elaboração da solução.

- Payload em JSON: os dados que serão enviados entre o servidor e o cliente;

NEW GROUP LABS

- Insomnia: framework Open Source para desenvolvimento/teste de API, ou seja, no contexto atual, será responsável pelo disparo dos payloads para a API;
- Python 3.13: para o desenvolvimento da API;
- AWS API Gateway: serviço que permite que desenvolvedores criem, publiquem, mantenham, monitorem e protejam APIs em qualquer escala com facilidade;
- AWS Lambda: serviço de computação sem servidor para executar código sem a necessidade de provisionar ou gerenciar servidores.

Figura 2 – Arquitetura da solução

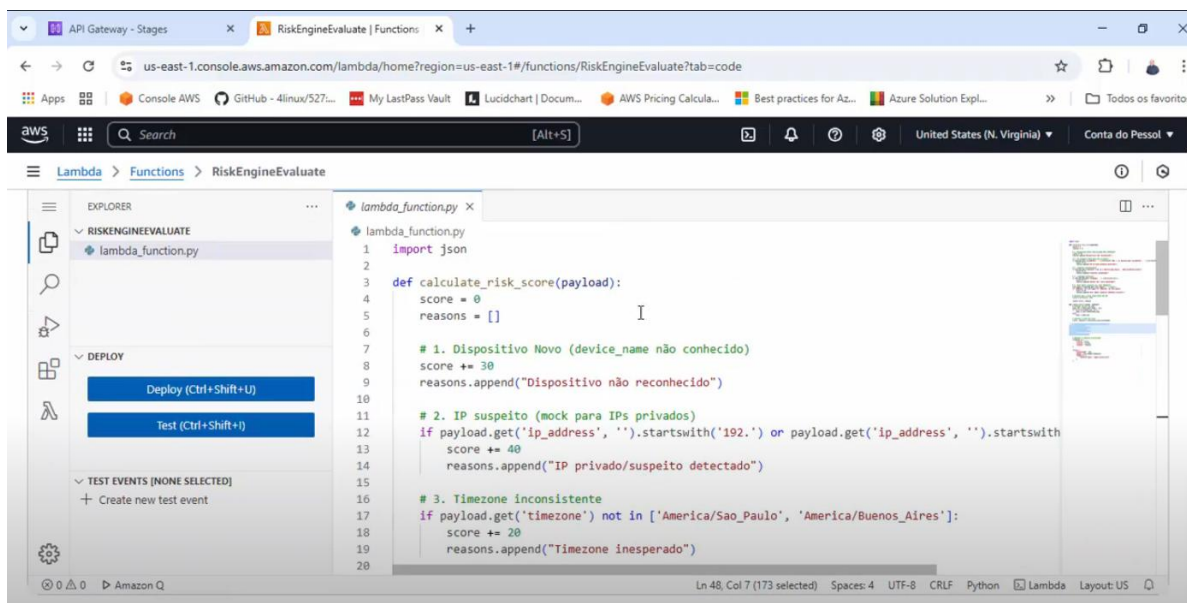


Fonte: Os autores (2025)

4.4. DEMO DA SOLUÇÃO – FASE 1

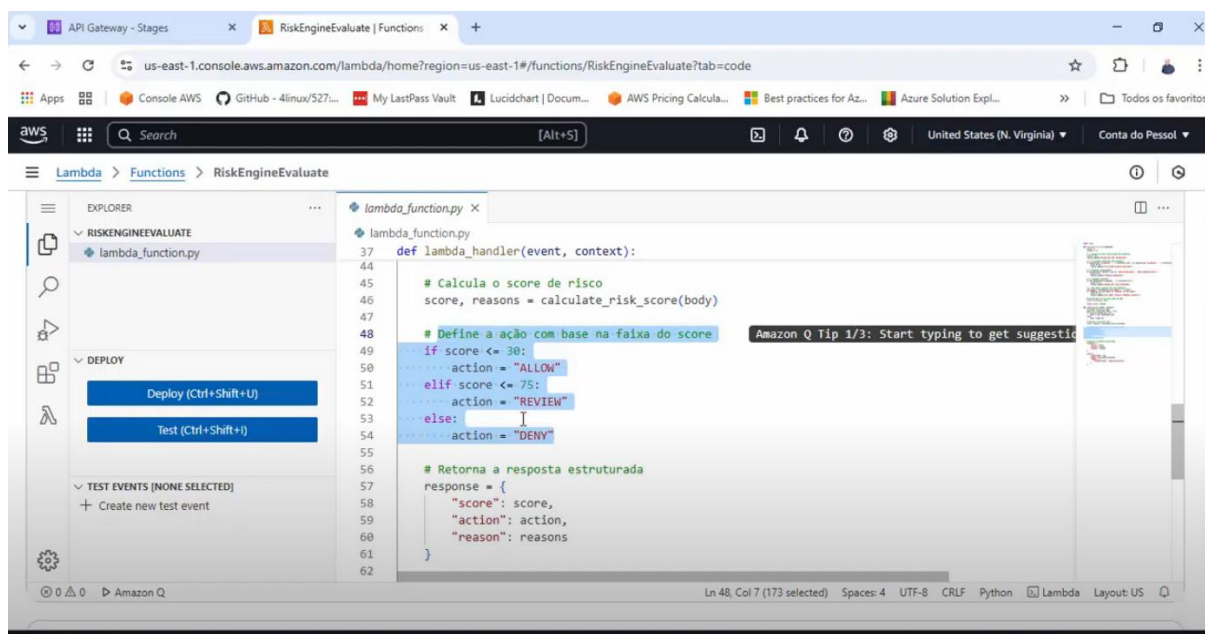
Em um ambiente AWS, foi possível iniciar o desenvolvimento da solução proposta, aplicando a API construída em Python na Lambda que contém a lógica para verificar o primeiro acesso, IP suspeito, fuso horário e outras informações necessárias.

Figura 3 – AWS Lambda, back-end desenvolvido em Python 3.13



Fonte: Os autores (2025)

Figura 4 – Lógica para os Score Points, definindo cada ação com base na faixa de score (<= 30 – ALLOW, <= 75 – REVIEW e >= 76 – DENY)

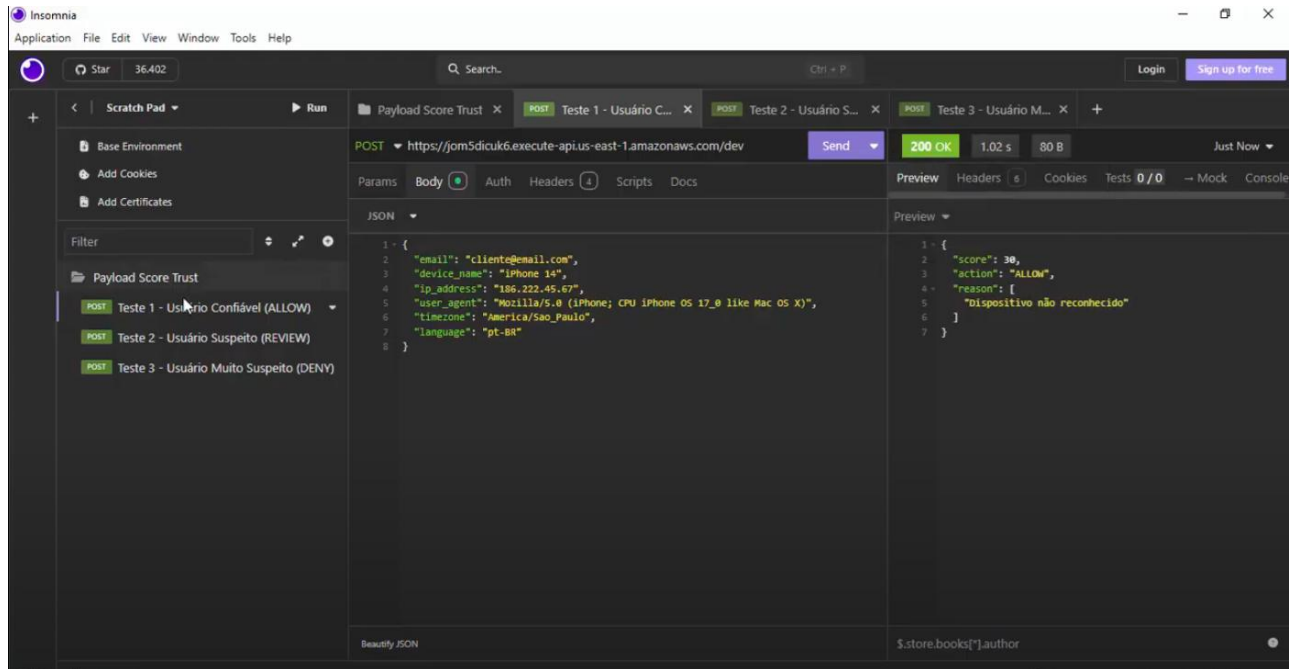


Fonte: Os autores (2025)

Através do Insomnia, uma chamada de API é realizada para demonstrar a funcionalidade do código Python apresentado para cada um dos usuários, ou seja, usuário confiável (ALLOW), usuário suspeito (REVIEW) e usuário muito suspeito (DENY).

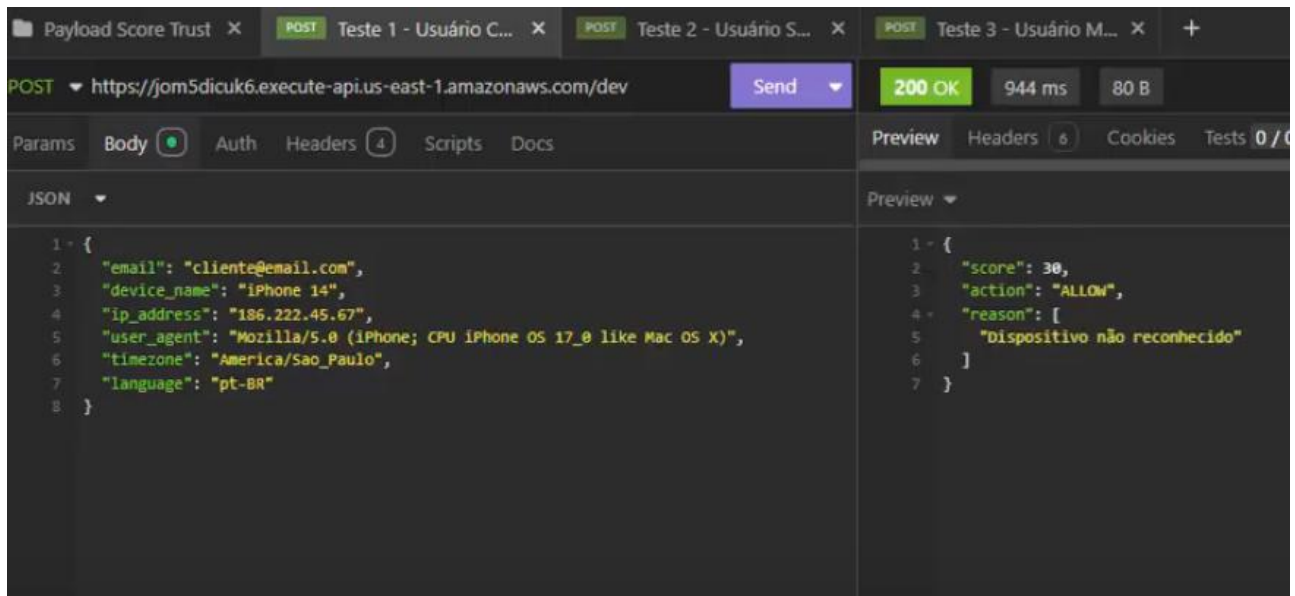
NEW GROUP LABS

Figura 5 – Tela do Insomnia



Fonte: Os autores (2025)

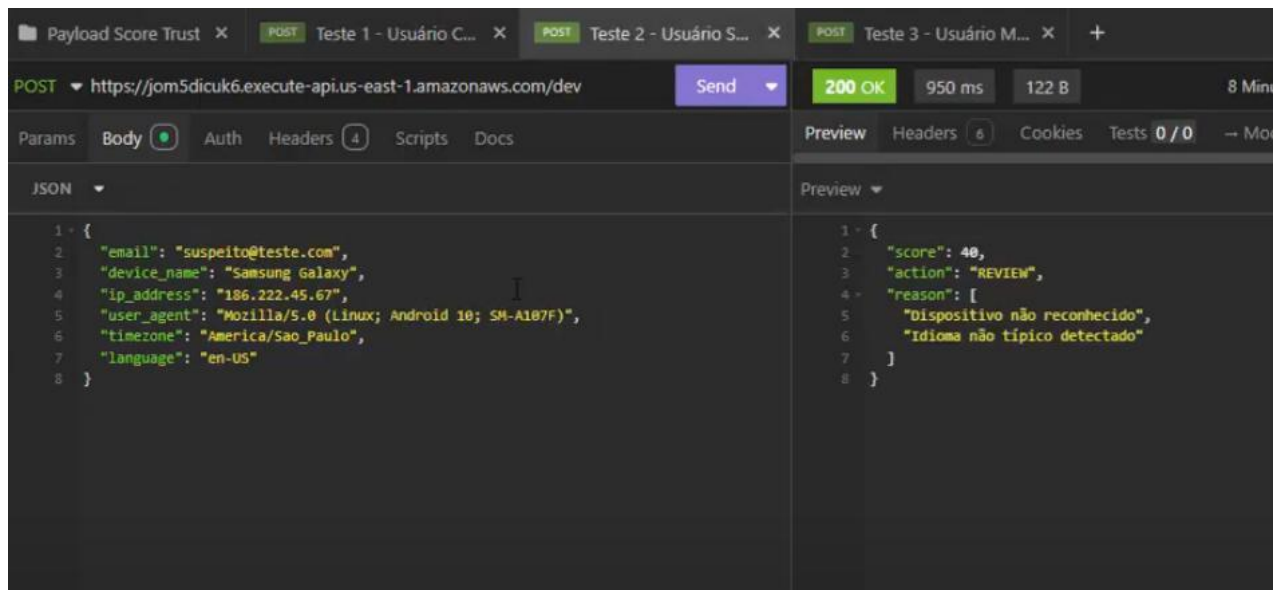
Figura 6 – Chamada de API para usuário legítimo



Fonte: Os autores (2025)

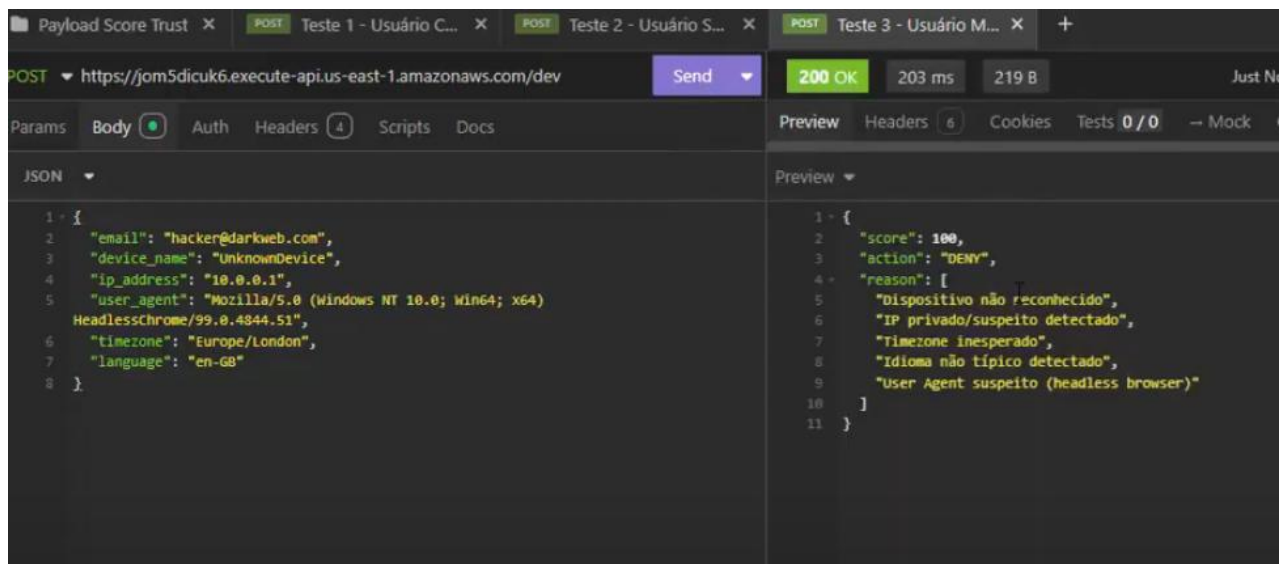
NEW GROUP LABS

Figura 7 – Chamada de API para usuário suspeito, visto que seu idioma não foi o tipicamente detectado para aquela conta



Fonte: Os autores (2025)

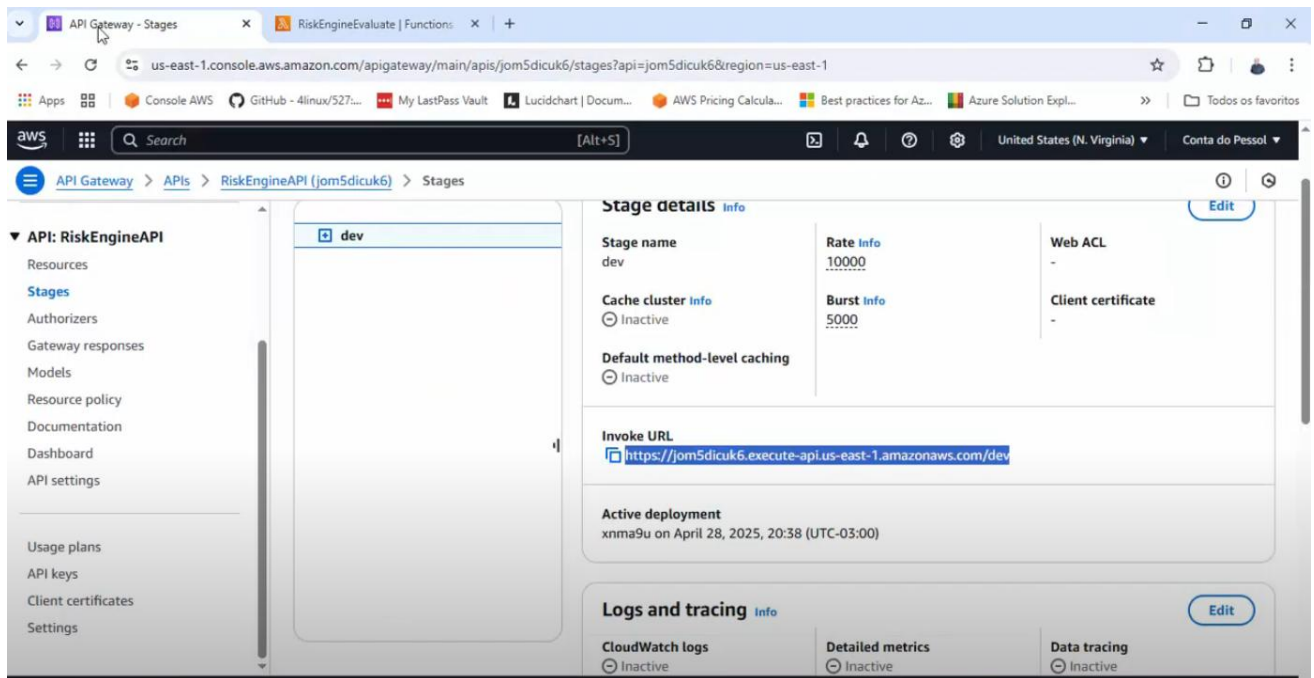
Figura 8 – Chamada de API para usuário muito suspeito, já que suas informações são totalmente desconhecidas, alcançando a pontuação de 100 no score (acesso negado)



Fonte: Os autores (2025)

NEW GROUP LABS

Figura 9 – API Gateway na AWS que realiza as chamadas de API



Fonte: Os autores (2025)

5.1. STACK DE DESENVOLVIMENTO DA FASE 2

- Front-end (HTML);
- SDK (simples em JavaScript);
- Backend (API em Python 3.13);
- AWS API Gateway (Method POST): envia dados ao servidor;
- AWS Lambda (backend em Python);
- AWS DynamoDB: serviço de banco de dados não relacional (NoSQL) e que até a presente fase possui a tabela chamada RiskEvents.

Figura 10 – Arquitetura da solução “Score Trust”



NEW GROUP LABS

5.2. DEMO DA SOLUÇÃO – FASE 2

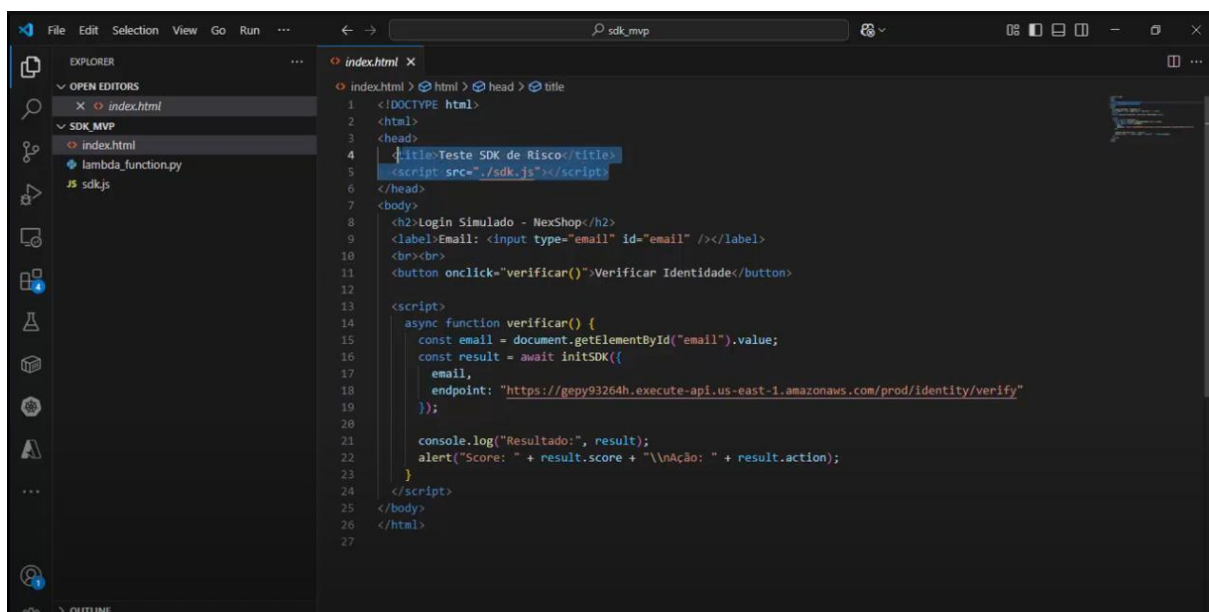
Com base no estudo levantado e apresentado na fase 1, a equipe iniciou os testes no ambiente simulado AWS para averiguar o pleno funcionamento da solução. De forma inicial, um front-end simples em HTML foi desenvolvido, carregando o SDK e solicitando um único input — o e-mail. Os dados são carregados em um formato JSON e enviados para o *endpoint* da API (sendo o caminho esperado */identity/verify*).

Figura 11 – Path */identity/verify* da API no SDK

```
try {  
  const res = await fetch(config.endpoint || "/identity/verify", {  
    method: "POST",  
    headers: { "Content-Type": "application/json" },  
    body: JSON.stringify(payload)  
  });  
  return await res.json();  
}
```

Fonte: Os autores (2025)

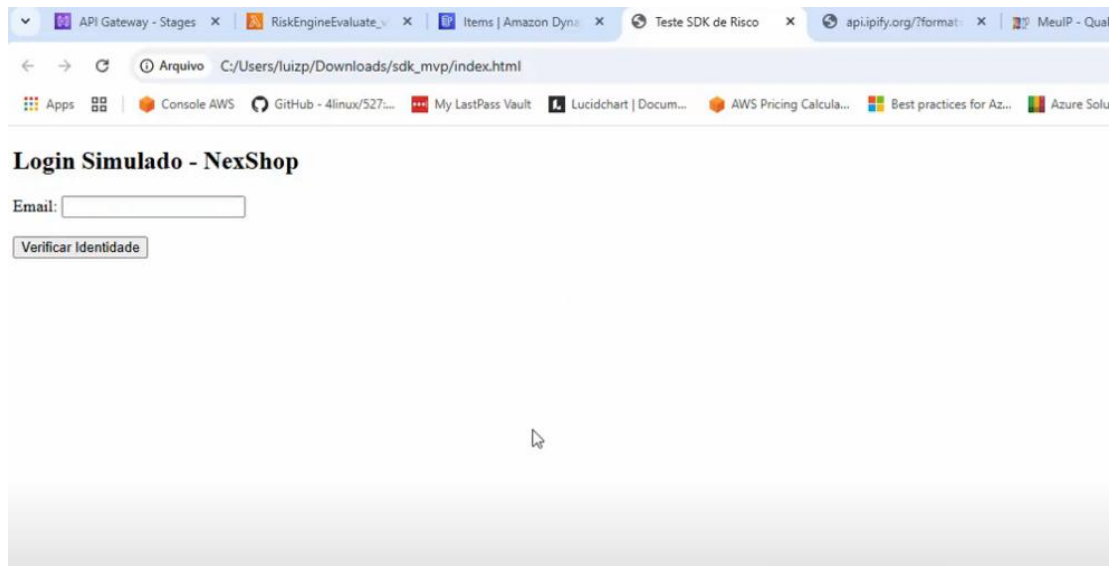
Figura 12 – Front-end em HTML



Fonte: Os autores (2025)

NEW GROUP LABS

Figura 13 – Aparência do front-end



Fonte: Os autores (2025)

O SDK, ao ser carregado, consulta e valida o IP público do usuário por meio da API ipify, assim como realiza a formatação de informações como user-agent, linguagem, fuso horário e resolução da tela, formulando um payload para ser disparado com JSON.

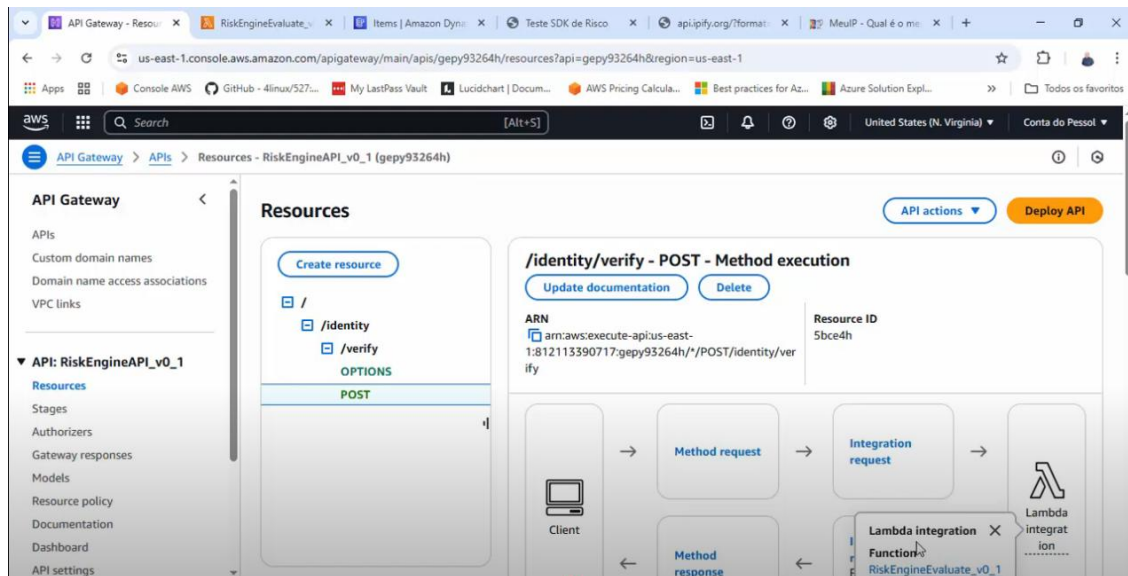
Figura 14 – SDK em JSON apresentando a API ipify e a coleta de fingerprint

```
JS sdkjs x
JS sdkjs > initSDK > getIp > res
1 // sdk.js
2
3 async function initSDK(config = {}) {
4   const startTime = Date.now();
5
6   const getIp = async () => {
7     try {
8       const res = await fetch("https://api.ipify.org?format=json");
9       const data = await res.json();
10      return data.ip;
11    } catch {
12      return "0.0.0.0";
13    }
14  };
15
16  const getFingerprint = async () => {
17    return {
18      user_agent: navigator.userAgent,
19      language: navigator.language,
20      timezone: Intl.DateTimeFormat().resolvedOptions().timeZone,
21      screen_resolution: `${window.screen.width}x${window.screen.height}`,
22      time_on_page: Math.round((Date.now() - startTime) / 1000)
23    };
24  };
25
26  const ip_address = await getIp();
27  const fingerprint = await getFingerprint();
28
29  const payload = {
30    email: config.email || "",
31    device_name: config.device_name || navigator.platform
```

Fonte: Os autores (2025)

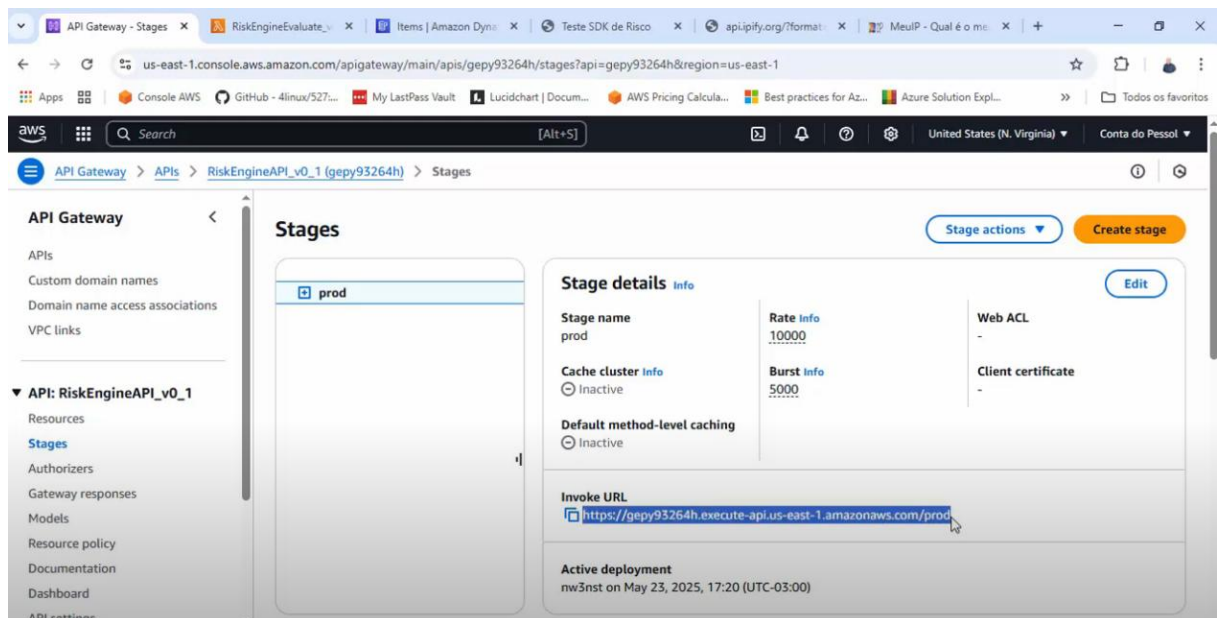
NEW GROUP LABS

Figura 15 – AWS API Gateway integrado com a Lambda e caminho /identity/verify do endpoint



Fonte: Os autores (2025)

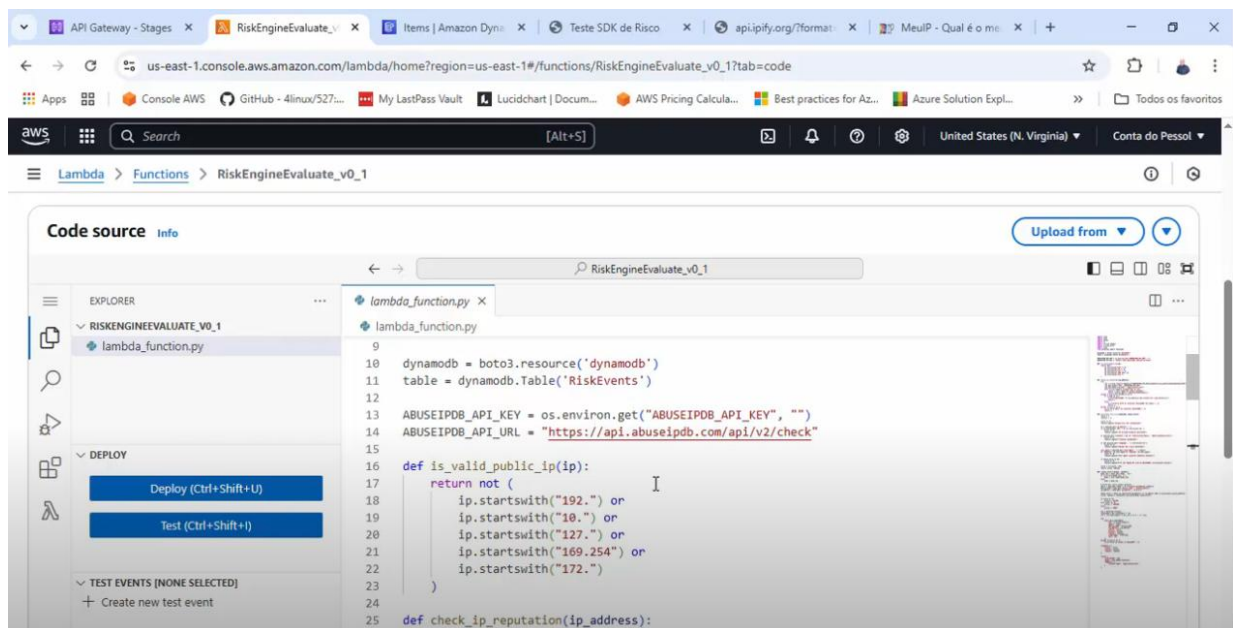
Figura 16 – Endpoint destacado



Fonte: Os autores (2025)

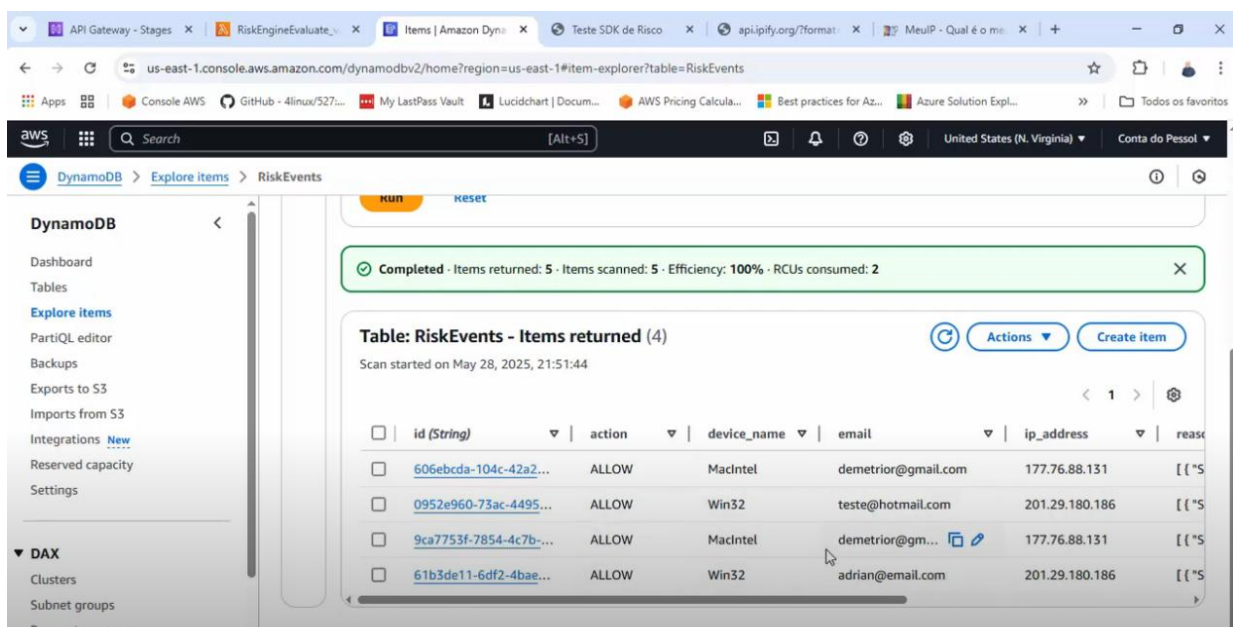
NEW GROUP LABS

Figura 17 – Back-end atualizado, apresentando a integração com o AbuseIP



Fonte: Os autores (2025)

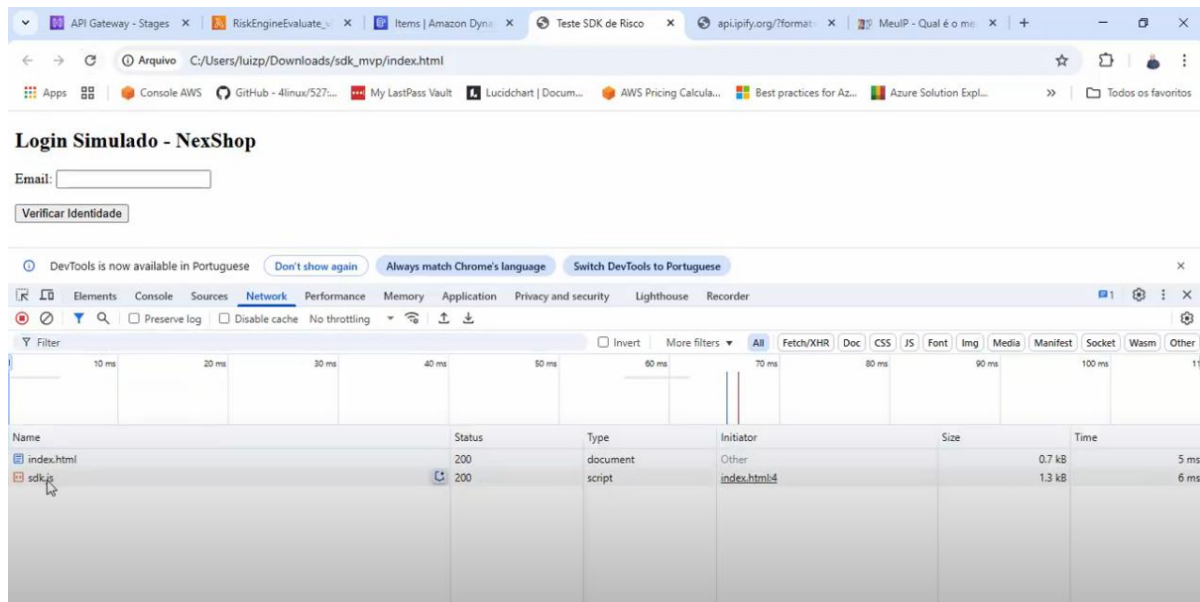
Figura 18 – Banco de dados no AWS DynamoDB, apresentando a tabela “RiskEvents” (dados recebidos)



Fonte: Os autores (2025)

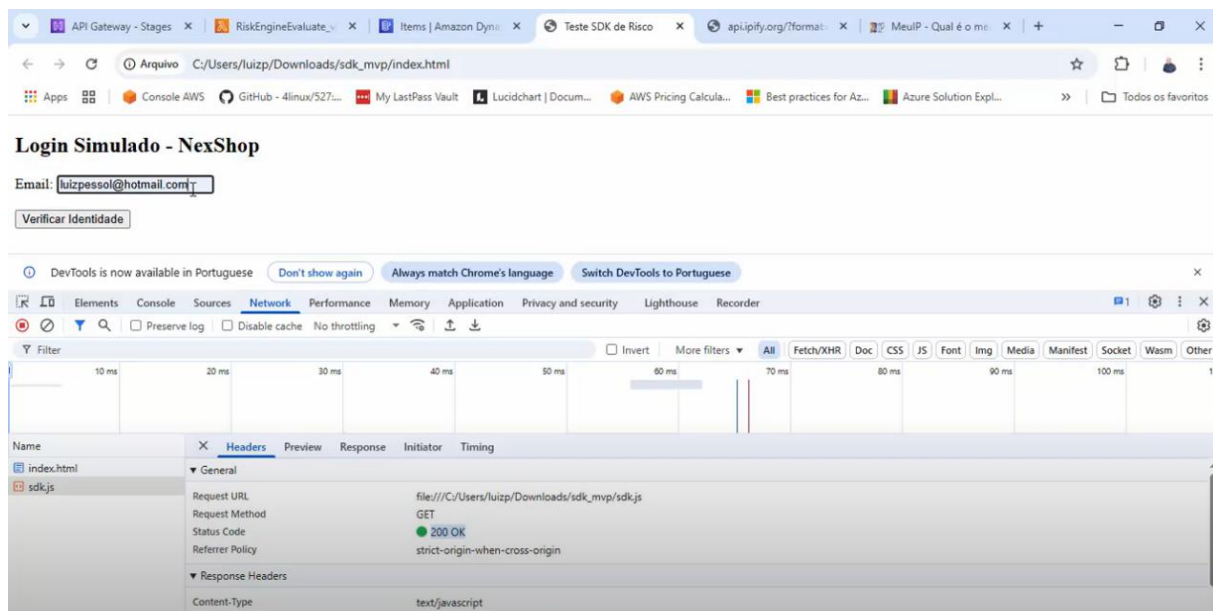
NEW GROUP LABS

Figura 19 – Simulação de um login na página de front-end, onde é possível observar o SDK carregado



Fonte: Os autores (2025)

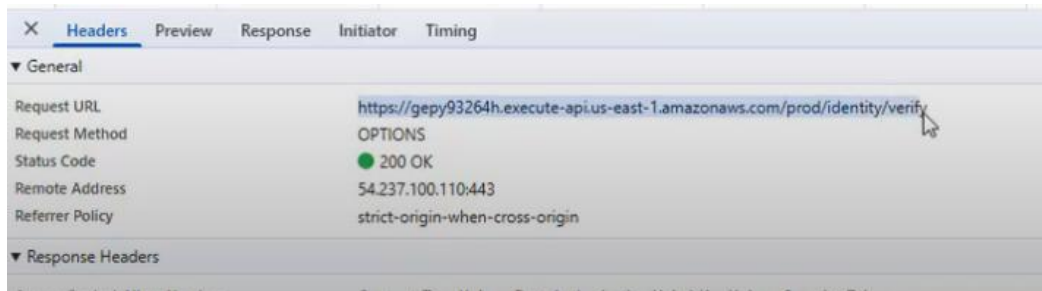
Figura 20 – Teste a partir de um e-mail pessoal



Fonte: Os autores (2025)

NEW GROUP LABS

Figura 21 – Requisição feita para a API, status code 200



Fonte: Os autores (2025)

Figura 22 – Banco de dados atualizado com o e-mail utilizado para os testes, indicando que dados estão sendo armazenados

Completed · Items returned: 5 · Items scanned: 5 · Efficiency: 100% · RCUs consumed: 2

Table: RiskEvents - Items returned (5)

Scan started on May 28, 2025, 21:59:40

< 1 >

⚙

<input type="checkbox"/>	id (String)	action	device_name	email	ip_address	reason
<input type="checkbox"/>	606ebcda-104c-42a2...	ALLOW	MacIntel	demetrior@gmail.com	177.76.88.131	[{"S
<input type="checkbox"/>	0952e960-73ac-4495...	ALLOW	Win32	teste@hotmail.com	201.29.180.186	[{"S
<input type="checkbox"/>	9ca7753f-7854-4c7b-...	ALLOW	MacIntel	demetrior@gmail.com	177.76.88.131	[{"S
<input type="checkbox"/>	61b3de11-6df2-4bae...	ALLOW	Win32	adrian@email.com	201.29.180.186	[{"S
<input type="checkbox"/>	3482d60a-0719-485...	ALLOW	Win32	luizpessol@hotmail.com	179.130.165.52	[{"S

Fonte: Os autores (2025)

6. Implementação Avançada da Solução

6.1. STACK DE DESENVOLVIMENTO DA FASE 3

Neste tópico, apresentamos a Stack de Desenvolvimento utilizado na Fase 3, ou seja, o conjunto de ferramentas e linguagens de programação aplicadas para a elaboração da solução.

- Front-end (HTML, Java);
- SDK v1.1 (JavaScript);
- Backend (API em Python 3.13);
- AWS API Gateway (Method POST): envia dados ao servidor;
- AWS Lambda (backend em Python);
- AWS DynamoDB: serviço de banco de dados não relacional (NoSQL) que possui as seguintes tabelas: RiskEvents, Known Devices, RuleWeights, Scoring Rules;
- Dashboard (Backend: Python, Front-end: JavaScript);
- Route 53 (Registro e Gerenciamento do domínio);
- AWS WAF (Proteção de ataques na borda);
- AbuseIPDB API (external API).

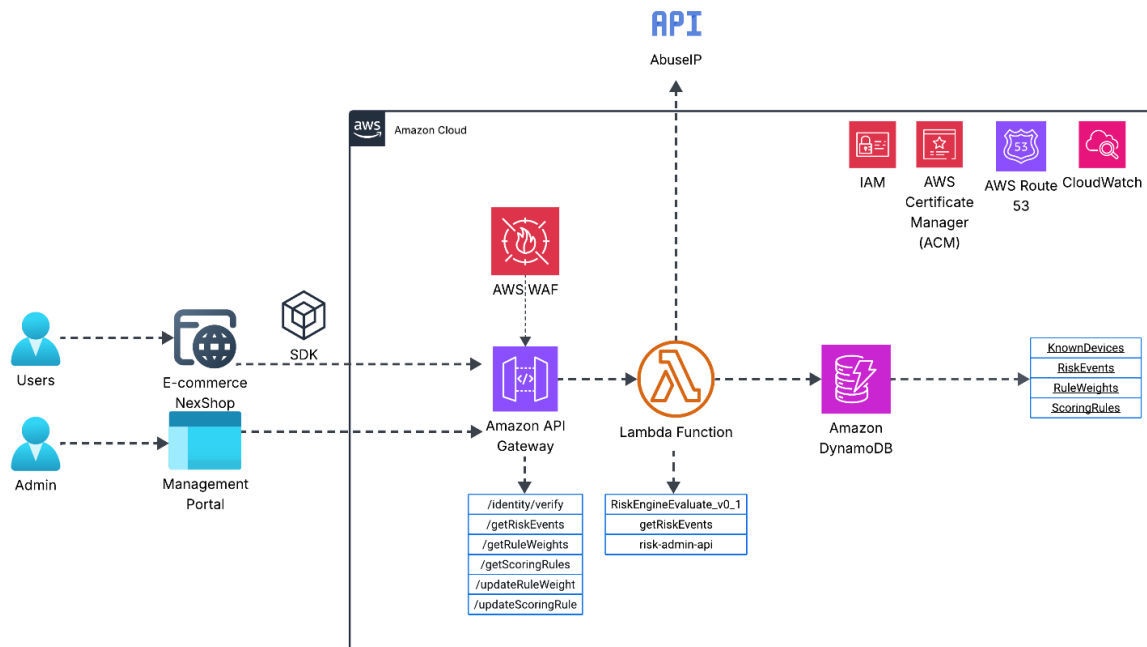
Na figura 23, é possível observar a arquitetura atualizada da solução, incluindo o que foi apresentado na fase 1 e fase 2 — Amazon API Gateway, Lambda Function, SDK v1, front-end simples, Amazon DynamoDB —, assim como o que foi proposto para a fase 3 — SDK v1.1, front-end atualizado, expansão da arquitetura com serviços gerenciados da AWS, incluindo:

- **Segurança e controle de acesso:** AWS WAF, IAM, ACM;
- **Gerenciamento de APIs e execução de lógica de negócio:** Amazon API Gateway e AWS Lambda;
- **Persistência de dados:** Amazon DynamoDB;
- **Monitoramento e DNS:** CloudWatch e Route 53;

NEW GROUP LABS

- **Integração externa:** API AbuseIP para validação de IPs suspeitos.

Figura 23 – Arquitetura da solução “Score Trust” atualizada



Fonte: Os autores (2025)

6.2. DEMO DA SOLUÇÃO – FASE 3

Com base no estudo levantado e apresentado na fase 1, a equipe evoluiu a arquitetura da solução no ambiente simulado AWS, aprimorando o SDK para a versão 1.1.

Nesta etapa, o SDK passou a coletar dados detalhados do dispositivo, como navegador, idioma, resolução de tela e fuso horário, gerando um identificador único via hash SHA-256. Os dados são enviados em formato JSON para o endpoint da API, que agora conta com lógica de score dinâmico baseada em tabelas específicas no DynamoDB, como **RuleWeights** e **ScoringRules**. A solução também incorporou novos serviços da AWS, como WAF, Route 53 e integração com AbuseIPDB, reforçando a segurança e a escalabilidade da aplicação.

Figura 24 – Versão V1.1 do SDK

```
1 // sdk.js
2
3 async function generateDeviceHash() {
4   const userAgent = navigator.userAgent || '';
5   const language = navigator.language || '';
6   const screenSize = `${screen.width}x${screen.height}`;
7   const timezone = Intl.DateTimeFormat().resolvedOptions().timeZone || '';
8
9   const raw = `${userAgent}|${language}|${screenSize}|${timezone}`;
10  const encoder = new TextEncoder();
11  const data = encoder.encode(raw);
12  const hashBuffer = await crypto.subtle.digest('SHA-256', data);
13  const hashArray = Array.from(new Uint8Array(hashBuffer));
14  const hashHex = hashArray.map(b => b.toString(16).padStart(2, '0')).join('');
15  return hashHex;
16 }
17
18 async function sendRiskPayload(email) {
19   const userAgent = navigator.userAgent || '';
20   const language = navigator.language || '';
21   const timezone = Intl.DateTimeFormat().resolvedOptions().timeZone || '';
22   const deviceHash = await generateDeviceHash();
23
24   const payload = {
25     email: email,
26     device_name: userAgent,
27     user_agent: userAgent,
28     language: language,
29     timezone: timezone,
30     device_hash: deviceHash
31   };
32
33   try {
34     const response = await fetch("https://api.score-trust.com/identity/verify", {
35       method: "POST",
36       headers: {
37         "Content-Type": "application/json",
38         "x-api-key": "DTFu5bcCwjwtilrFuscG6CDXAZ16wP45jnZpfFn1"
```

Fonte: Os autores (2025)

Na imagem anterior (figura 24), uma parte do código do SDK é destacada, onde alguns itens para a identificação do dispositivo e coleta de dados são demonstrados, como:

- Navegador (“userAgent”);
- Idioma (“language”);
- Resolução da tela (“screenSize”);
- Fuso horário (“timezone”).

A partir dessas informações, um hash SHA-256 é gerado, criando um identificador único para cada dispositivo. Na versão atual do SDK, uma API Key é utilizada para atuar como uma chave atribuída a um plano de serviço com um limite diário de quinze requisições para essa chave de teste.

Figura 25 – Versão V1.1 do SDK

```
30 device_hash: devicehash
31 };
32
33 try {
34   const response = await fetch("https://api.score-trust.com/identity/verify", {
35     method: "POST",
36     headers: {
37       "Content-Type": "application/json",
38       "x-api-key": "DTFu5bcCwjwtilrFuSCG6CDXAZ16wP45jnZpfFn1"
39     },
40     body: JSON.stringify(payload)
41   });
42
43   const result = await response.json();
44
45   if (!response.ok) {
46     alert(`Erro ao verificar identidade:\n${JSON.stringify(result, null, 2)}`);
47   } else {
48     alert(`Risco: ${result.action} (score ${result.score})`);
49   }
50
51   console.log(result);
52   return result;
53
54 } catch (error) {
55   alert(`Erro de rede ou CORS: ${error.message}`);
56   console.error("Erro ao enviar payload:", error);
57   return { error: "Erro de comunicação com o Risk Engine" };
58 }
59 }
```

Fonte: Os autores (2025)

Figura 26 – Banco de dados no AWS DynamoDB, tabela “KnownDevices” (dispositivos conhecidos pelo banco de dados)

Tabela: KnownDevices - Itens retornados (40)

Ações

Criar item

Verificação iniciada em agosto 30, 2025, 20:38:18

< 1 >

<input type="checkbox"/>	email (String)	device_hash (String)	created_at
<input type="checkbox"/>	cliente@email.com	unknown	2025-08-13T14:14:25.429618+00:00
<input type="checkbox"/>	camille.costa@hyund...	1c4e78f1f66cfc6810d02...	2025-06-04T05:52:28.757842+00:00
<input type="checkbox"/>	camille.costa@hyund...	a6e9af019bb5398ab108...	2025-08-26T22:45:39.383270+00:00
<input type="checkbox"/>	demetrio@drpp.com.br	26fc72c0205595aacec16...	2025-08-26T18:18:11.197260+00:00
<input type="checkbox"/>	demetrio@drpp.com.br	b8890ef01992f580bc14...	2025-06-09T20:40:38.991448+00:00
<input type="checkbox"/>	demetrio@drpp.com.br	cfd5daf17548000b756b...	2025-08-18T22:53:31.259965+00:00
<input type="checkbox"/>	joao@gmail.com	7ea625a13f41b8261333...	2025-08-06T18:41:22.192328+00:00
<input type="checkbox"/>	carolisparo@gmail.com	0bfbe478eeeb9add1efbf...	2025-08-26T18:31:31.583872+00:00
<input type="checkbox"/>	rennan@fulano.com	4ecbc892d56ae8c68533...	2025-08-13T18:41:20.949200+00:00
<input type="checkbox"/>	rennan@fulano.com	7ea625a13f41b8261333...	2025-08-13T18:33:47.426887+00:00
<input type="checkbox"/>	demetrior@outlook.c...	26fc72c0205595aacec16...	2025-08-26T18:21:11.412589+00:00
<input type="checkbox"/>	demetrior@gmail.com	26fc72c0205595aacec16...	2025-08-26T18:17:46.857852+00:00

Fonte: Os autores (2025)

NEW GROUP LABS

Na tabela anterior (KnowDevices – Itens retornados), temos como objetivo armazenar os dispositivos que já passaram pela verificação de “score points” do SDK e são arquivados pelo banco de dados como “dispositivos conhecidos”.

Figura 27 – Banco de dados no AWS DynamoDB, apresentando a tabela “RuleWeights”

Tabela: RuleWeights - Itens retornados (8) Verificação iniciada em agosto 30, 2025, 20:46:31

<input type="checkbox"/>	rule_id (String)	description	peso
<input type="checkbox"/>	useragent_suspeito	User Agent suspeito (headless browser)	50
<input type="checkbox"/>	device_unknown	Dispositivo não reconhecido	40
<input type="checkbox"/>	idioma_nao_pt	Idioma não típico detectado	10
<input type="checkbox"/>	timezone_inesperado	Timezone inesperado	20
<input type="checkbox"/>	pais_nao_br	País de origem não é Brasil	80
<input type="checkbox"/>	ip_privado	IP privado/suspeito detectado	40
<input type="checkbox"/>	device_known	Dispositivo não reconhecido	10
<input type="checkbox"/>	abuseipdb_alto	IP com reputação ruim no AbuseIPDB	20

Fonte: Os autores (2025)

A tabela “RuleWeights” (figura 27) visa definir o peso (influência) para o cálculo dinâmico do score final. Já a tabela “ScoringRules” (figura 28) contém a faixa de score (mínima/máxima) e a ação correspondente (allow, review, deny) conforme a seção 4.2 Solução Proposta Inicial, sendo o score final limitado a 100 pontos.

Figura 28 – Banco de dados no AWS DynamoDB, apresentando a tabela “ScoringRules”

Tabela: ScoringRules - Itens retornados (3) Verificação iniciada em agosto 30, 2025, 20:52:15

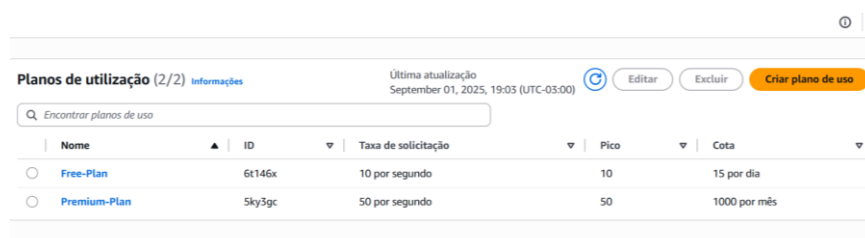
<input type="checkbox"/>	id (String)	action	max	min
<input type="checkbox"/>	r1	ALLOW	30	0
<input type="checkbox"/>	r2	REVIEW	75	31
<input type="checkbox"/>	r3	DENY	100	76

Fonte: Os autores (2025)

NEW GROUP LABS

Na busca por uma visão de negócios, é relevante reforçar que os pesos atribuídos a cada informação do dispositivo podem ser modificados conforme o interesse de cada cliente. Em adição, a equipe criou planos de uso para a API, que incluem o "Free Plan", um plano gratuito que permite 15 solicitações diárias, e o "Premium Plan", que permite 1000 solicitações mensais.

Figura 29 – Console da AWS na seção de Web API Gateway (Planos de utilização)



The screenshot shows the AWS API Gateway console's 'Usage Plans' section. It displays two plans: 'Free-Plan' and 'Premium-Plan'. The 'Free-Plan' has a request rate of 10 per second and a quota of 15 requests per day. The 'Premium-Plan' has a request rate of 50 per second and a quota of 1000 requests per month. The console includes a search bar, a table with columns for Name, ID, Request Rate, Throttle, and Quota, and buttons for 'Editar', 'Excluir', and 'Criar plano de uso'.

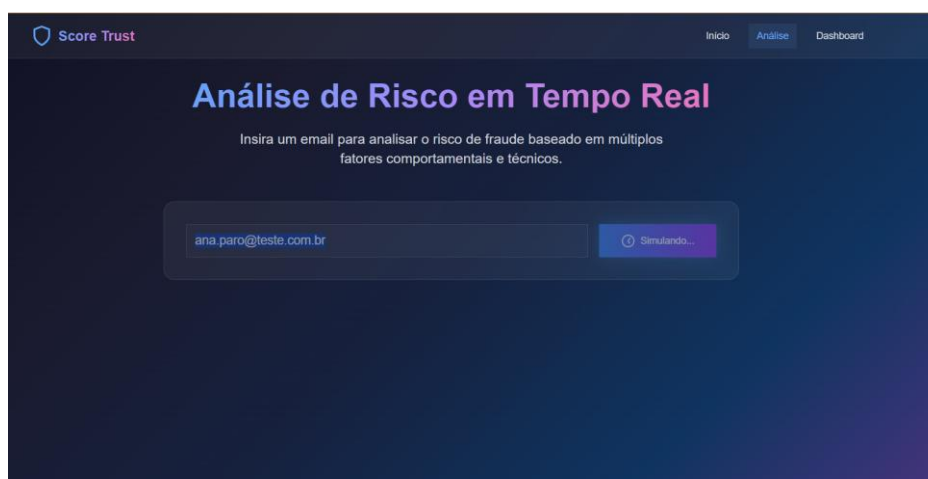
Nome	ID	Taxa de solicitação	Pico	Cota
Free-Plan	6t146k	10 por segundo	10	15 por dia
Premium-Plan	Sky3gc	50 por segundo	50	1000 por mês

Fonte: Os autores (2025)

6.2.1 DASHBOARD

A partir do SDK configurado e as conexões necessárias realizadas com o front-end, a equipe desenvolveu uma página para o dashboard da solução, oferecendo uma área de testes, o acompanhamento em tempo real dos eventos de risco e as informações recolhidas pelo SDK.

Figura 30 – Teste de acesso na página <https://scoretrust.com.br/analyze> na área de análise

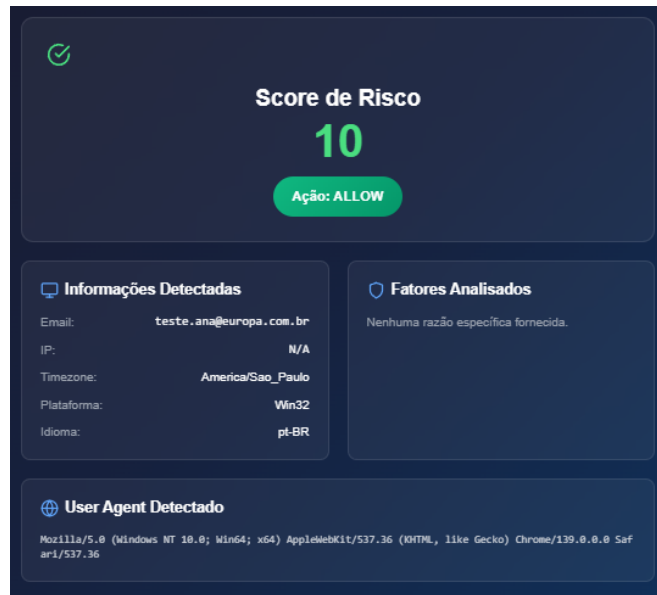


The screenshot shows the Score Trust dashboard's 'Análise de Risco em Tempo Real' section. It features a dark blue background with a white text input field containing the email 'ana.paro@teste.com.br'. To the right of the input field is a blue button labeled 'Simulando...'. Above the input field, there is a heading 'Análise de Risco em Tempo Real' and a subheading 'Insira um email para analisar o risco de fraude baseado em múltiplos fatores comportamentais e técnicos.' The dashboard also includes a navigation bar with 'Início', 'Análise', and 'Dashboard' links.

Fonte: Os autores (2025)

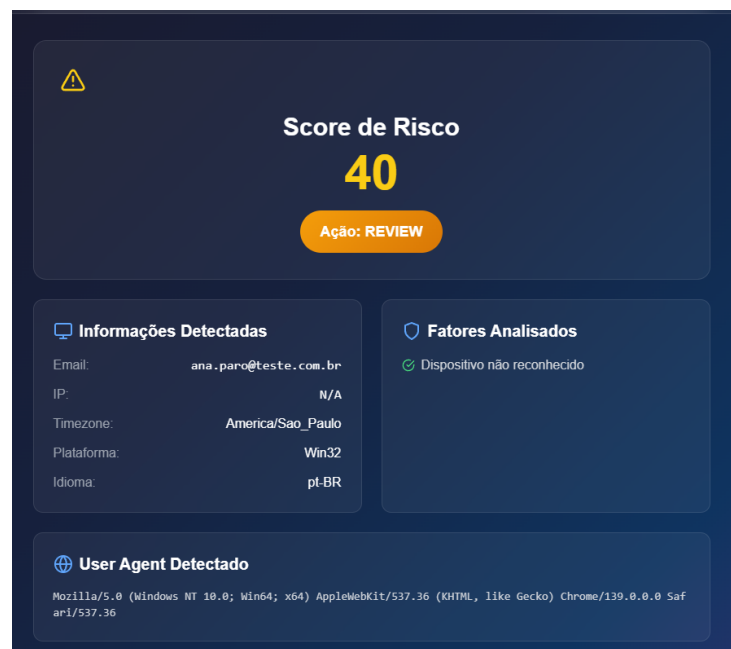
NEW GROUP LABS

Figura 31 – Teste de acesso na página <https://scoretrust.com.br/analyze> na área de análise utilizando um e-mail para testes



Fonte: Os autores (2025)

Figura 32 – Teste de acesso na página <https://scoretrust.com.br/analyze> na área de análise utilizando um e-mail para testes

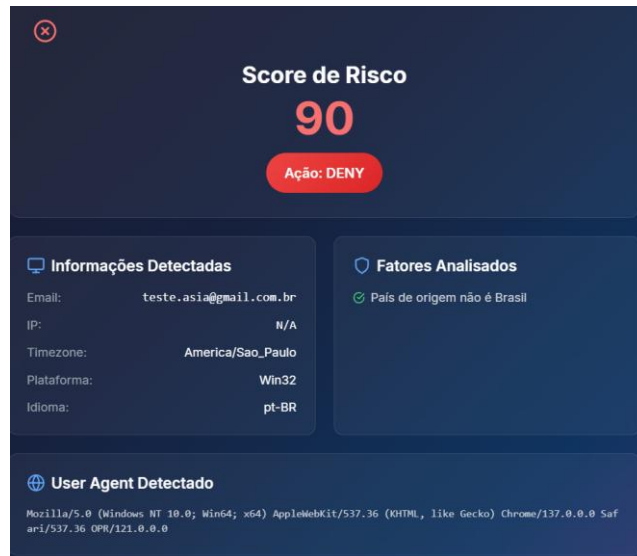


Fonte: Os autores (2025)

Nos testes exibido, a solução apresentou um score de 10 (ALLOW) — usuário já reconhecido no banco de dados — e de 40 (REVIEW), uma vez que o dispositivo não foi reconhecido, mas a localização é originária do Brasil.

NEW GROUP LABS

Figura 33 – Teste de acesso na página <https://scoretrust.com.br/analyze> na área de análise utilizando um e-mail para teste utilizando uma VPN.



Fonte: Os autores (2025)

Na figura 33, é apresentada a simulação de requisição utilizando uma VPN com origem na Ásia, mas com um dispositivo já reconhecido pelo banco de dados, que teve como resultado o score de risco 90, DENY. Já na figura 34, foi realizada a simulação de requisição utilizando uma VPN com origem europeia e que gerou como resultado o score de risco 100, DENY, baseado no país de origem e o dispositivo não ser reconhecido pelo banco de dados.

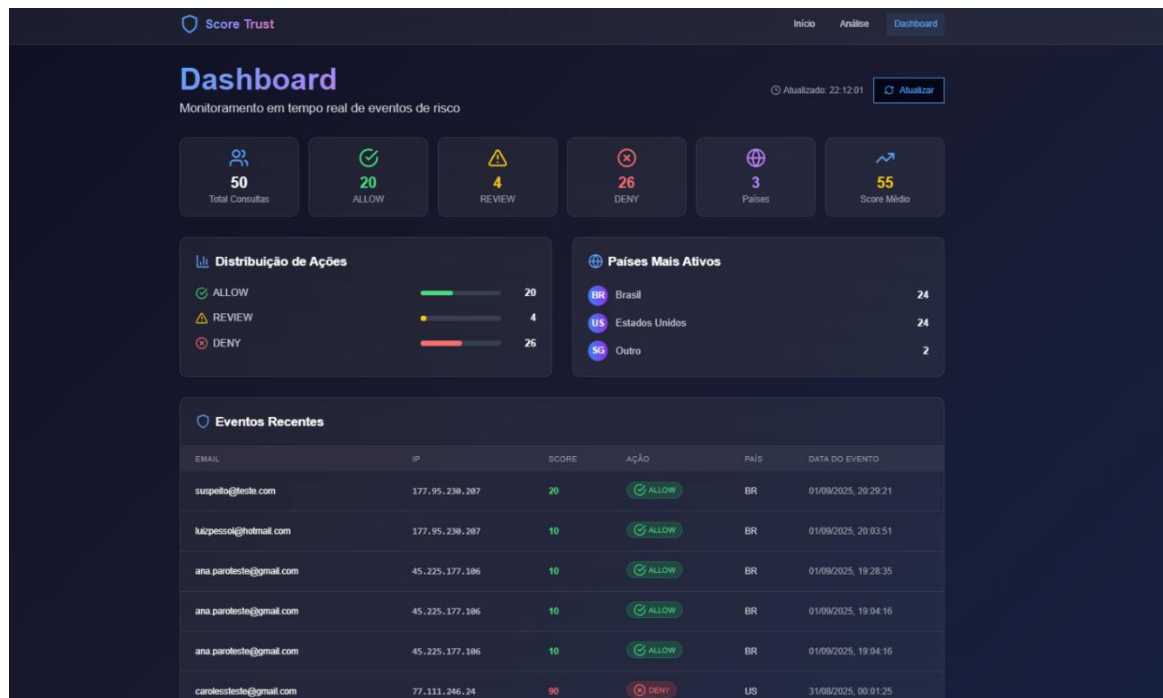
Figura 34 – Teste de acesso na página <https://scoretrust.com.br/analyze> na área de análise utilizando um e-mail para teste utilizando uma VPN.



Fonte: Os autores (2025)

NEW GROUP LABS

Figura 35 – Aparência de área de usuário para monitoramento dos eventos recentes



Fonte: Os autores (2025)

O SDK, ao ser carregado, consulta e valida o IP público do usuário por meio da API api.score-trust.com/identity/verify (figura 35), assim como realiza a formatação de informações como *user-agent*, linguagem, fuso horário e resolução da tela, formulando um payload para ser disparado com JSON.

Figura 36 – Teste de acesso na página <https://scoretrust.com.br/analyze>, requisição feita para a API, status code 200

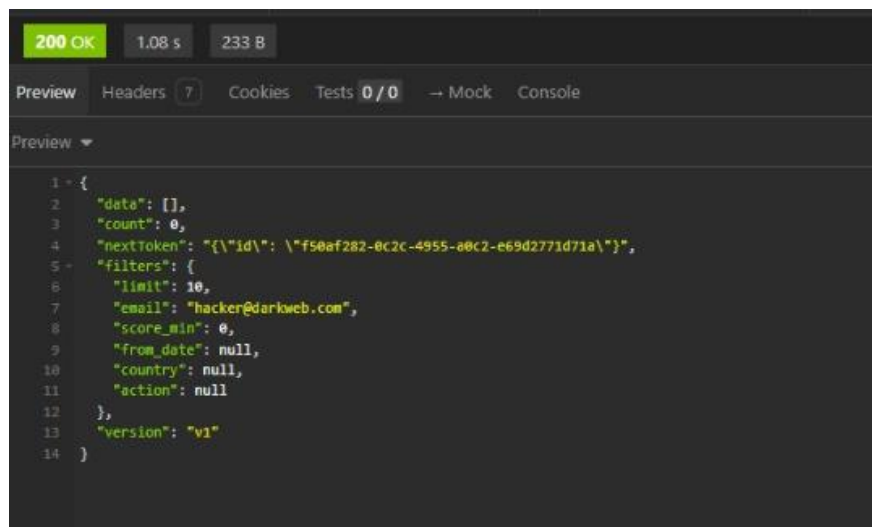
Name	×	Headers	Preview	Response	Initiator	>>
verify		▼ General				
verify		Request URL				
verify		https://api.score-trust.com/identity/verify				
verify		Request Method				
verify		OPTIONS				
verify		Status Code				
verify		200 OK				
verify		Remote Address				
verify		77.111.246.24:443				
verify		Referrer Policy				
verify		strict-origin-when-cross-origin				

Fonte: Os autores (2025)

6.2.2 TESTES DE REQUISIÇÃO NO INSOMNIA

Como apresentado na primeira etapa da solução (seção 4.4), o Insomnia foi novamente utilizado para testar novas chamadas de API, comprovando o funcionamento de cada módulo.

Figura 37 – Teste realizado no Insomnia efetuando uma consulta via e-mail



Fonte: Os autores (2025)

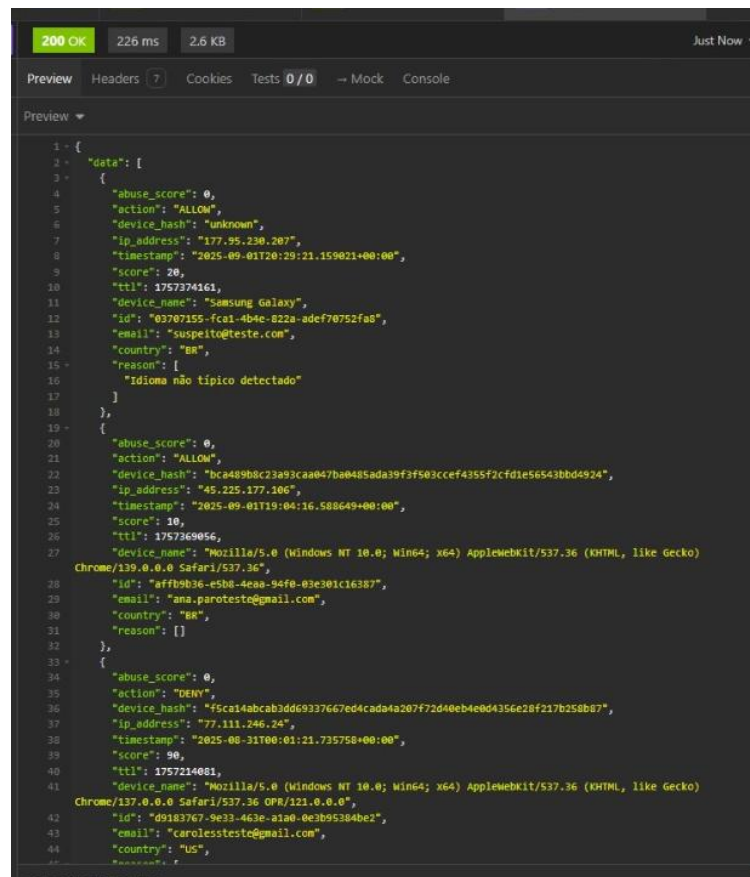
Figura 38 – Teste realizado no Insomnia consultando eventos pela data 31/08/2025



Fonte: Os autores (2025)

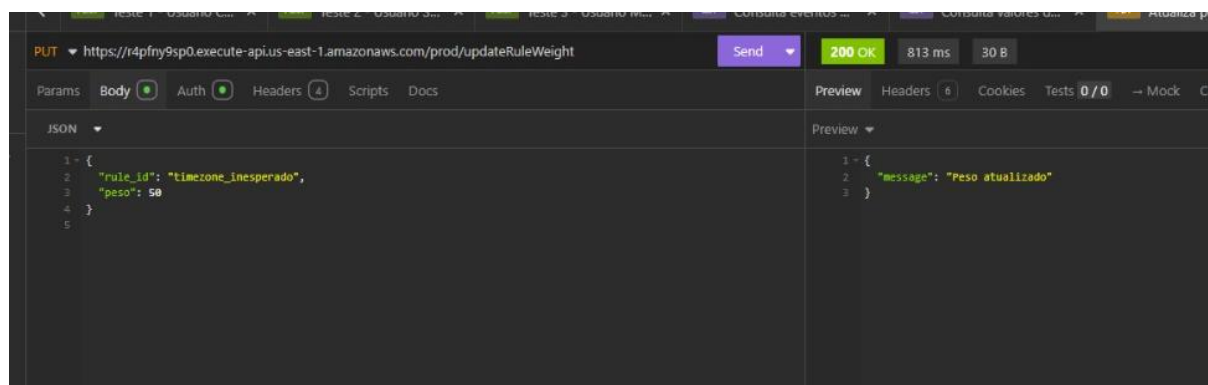
NEW GROUP LABS

Figura 39 – Teste realizado no Insomnia consultando pelo peso das regras



Fonte: Os autores (2025)

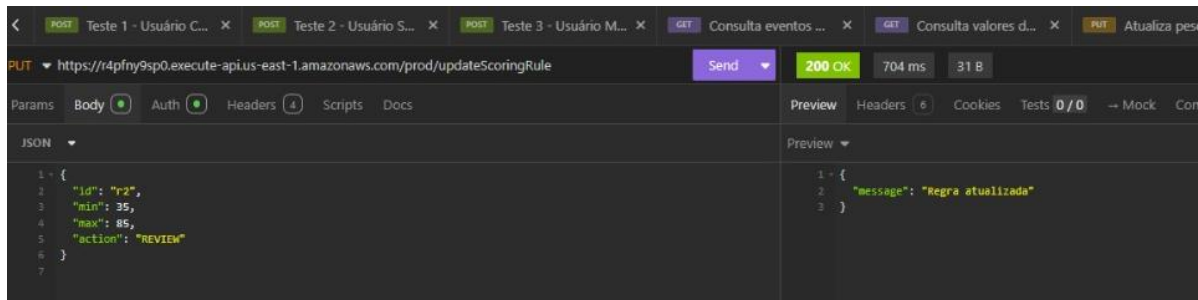
Figura 40 – Teste realizado no Insomnia realizando a alteração do peso “timezone” de 20 para 50



Fonte: Os autores (2025)

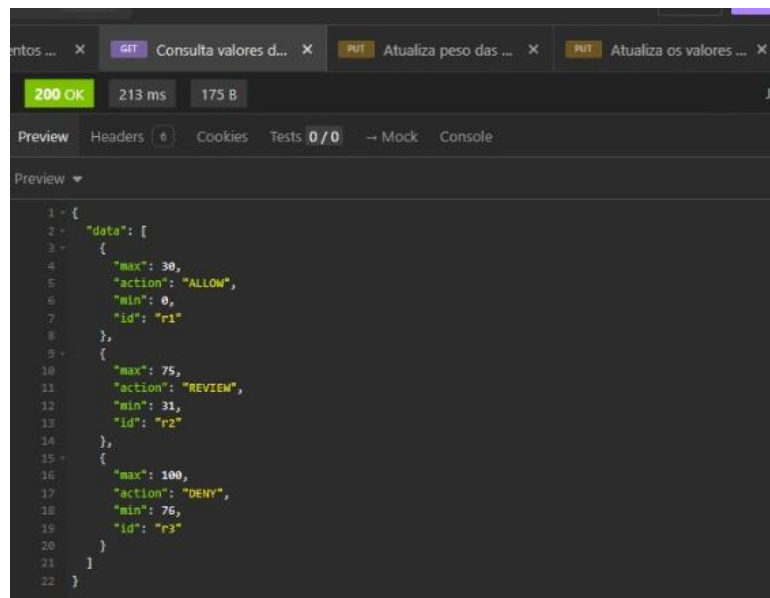
NEW GROUP LABS

Figura 41 – Teste realizado no Insomnia efetuando a alteração da regra “REVIEW”



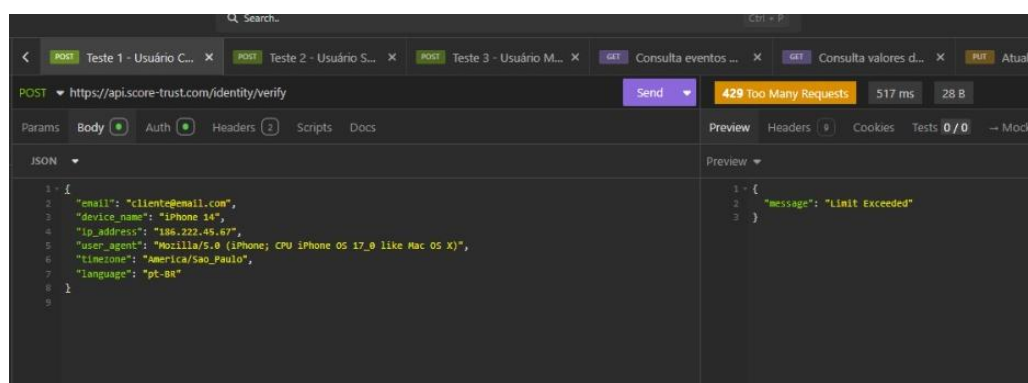
Fonte: Os autores (2025)

Figura 42 – Teste realizado no Insomnia realizando uma consulta pelos valores das ações



Fonte: Os autores (2025)

Figura 43 – Teste realizado no Insomnia representando o modo Free Plan consumido 100%



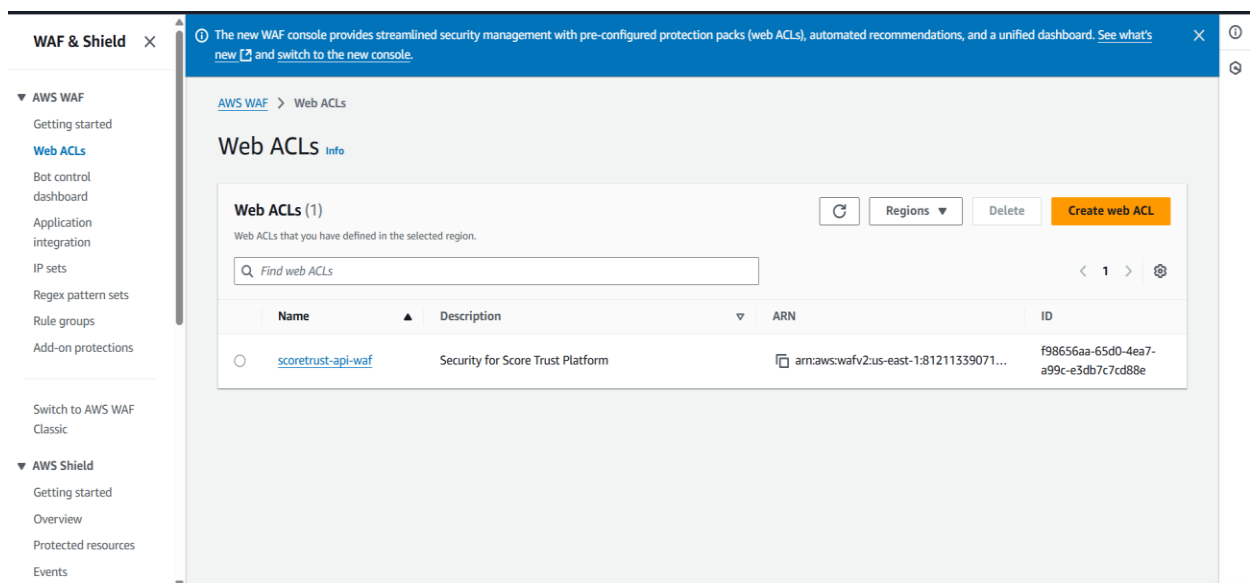
Fonte: Os autores (2025)

NEW GROUP LABS

6.2.3 ATUALIZAÇÕES DE SEGURANÇA

Como parte da evolução da arquitetura na Fase 3, a equipe passou a utilizar o AWS WAF (Web Application Firewall) para proteger a API da plataforma Score Trust contra ameaças na borda. Foi configurado um Web ACL chamado “scoretrust-api-waf”.

Figura 44 – Console do AWS WAF & Shield na seção de Web ACLs (Access Control Lists).



Fonte: Os autores (2025)

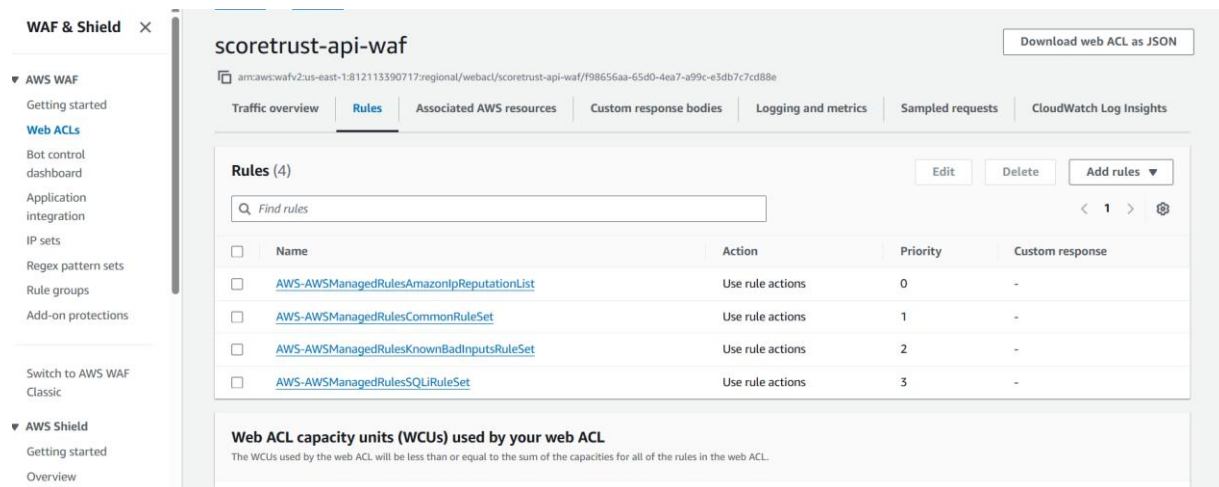
Na ACL, quatro regras gerenciadas pela AWS foram incluídas:

- **AWS-ManagedRulesAmazonIpReputationList:** bloqueia IPs com reputação maliciosa conhecida.
- **AWS- ManagedRulesCommonRuleSet:** protege contra-ataques comuns como XSS e injeções (as chamadas realizadas com o *Insomnia* foram bloqueadas por essa regra, já que simulamos um *HEADER* e ele identifica a ação como maliciosa).
- **AWS- ManagedRulesKnownBadInputsRuleSet:** detecta entradas malformadas ou suspeitas.
- **AWS-ManagedRulesSQLiRuleSet:** protege contra SQL Injection.

A utilização do WAF fortalece a segurança da aplicação, reduz riscos e melhora a confiabilidade do sistema frente a acessos maliciosos.

NEW GROUP LABS

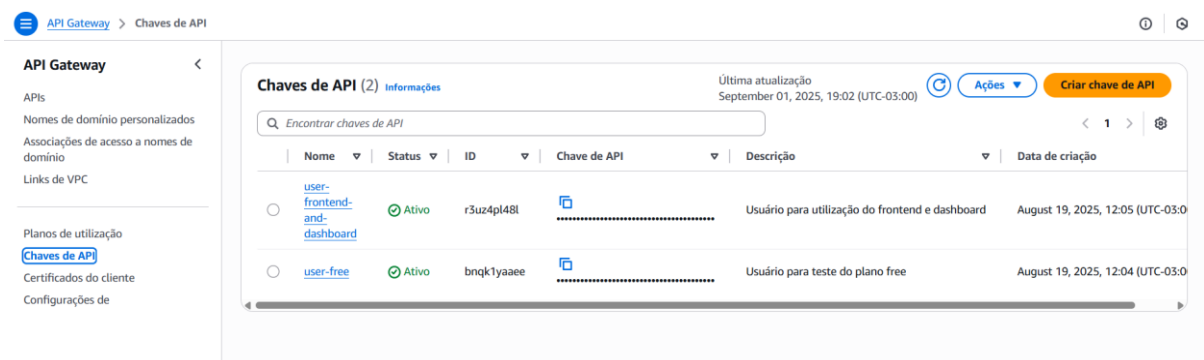
Figura 45 – Console do AWS WAF & Shield na seção de Web ACLs (Access Control Lists) Regras Gerenciadas



Fonte: Os autores (2025)

Foram acrescentados os ID's necessários para a segurança e gerenciamento das APIs utilizadas.

Figura 46 – Console do AWS na seção de Web API Gateway (Chaves de API)










Fonte: Os autores (2025)

6.2.4 DOCUMENTAÇÃO

Conforme solicitado pela Hakai Security, a equipe optou por criar a documentação do projeto Score Trust diretamente em um repositório do GitHub.

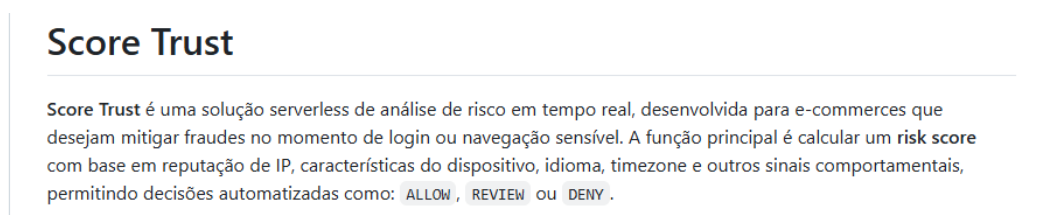
Figura 47 – Página inicial da documentação do projeto no GitHub (<https://github.com/luizpessol/score-trust/tree/main>) apresentando a estrutura do repositório

 luizpessol backup dos códigos backend	b376de2 · 4 days ago	🕒 55 Commits
 backend	backup dos códigos backend	4 days ago
 frontend	Update README.md	2 weeks ago
 img	Upload do desenho de arquitetura	last week
 insomnia	Update título documentação insomnia.yaml	3 weeks ago
 sdk	Documentação do código sdk.js	last week
 README.md	Update na documentação, detalhes dos recursos	last week

Fonte: Os autores (2025)

- **Backend:** Funções Lambda, integração com DynamoDB e lógica de avaliação de risco;
- **Frontend:** Interface em React para visualização e gestão dos eventos;
- **Insomnia:** Coleção de requisições para testar APIs;
- **SDK.js:** Biblioteca JavaScript para integrar o Score Trust em plataformas externas.

Figura 48 – Página inicial da documentação do projeto no GitHub (<https://github.com/luizpessol/score-trust/tree/main>) apresentando a estrutura do repositório



Fonte: Os autores (2025)

Na documentação, é possível ler sobre o funcionamento da solução Score Trust, a arquitetura utilizada na solução, principais componentes e funções, quais as regras determinadas para efetuar o cálculo do score, estrutura das tabelas AWS e exemplos de retorno da API, assim como informações sobre segurança, tecnologias aplicadas, observações e próximos passos.

7. Implementação Final da Solução

7.1. STACK DE DESENVOLVIMENTO DA FASE 4

Neste tópico, apresentamos a Stack de Desenvolvimento utilizado na Fase 4, ou seja, o conjunto de ferramentas e linguagens de programação aplicadas para a elaboração da solução.

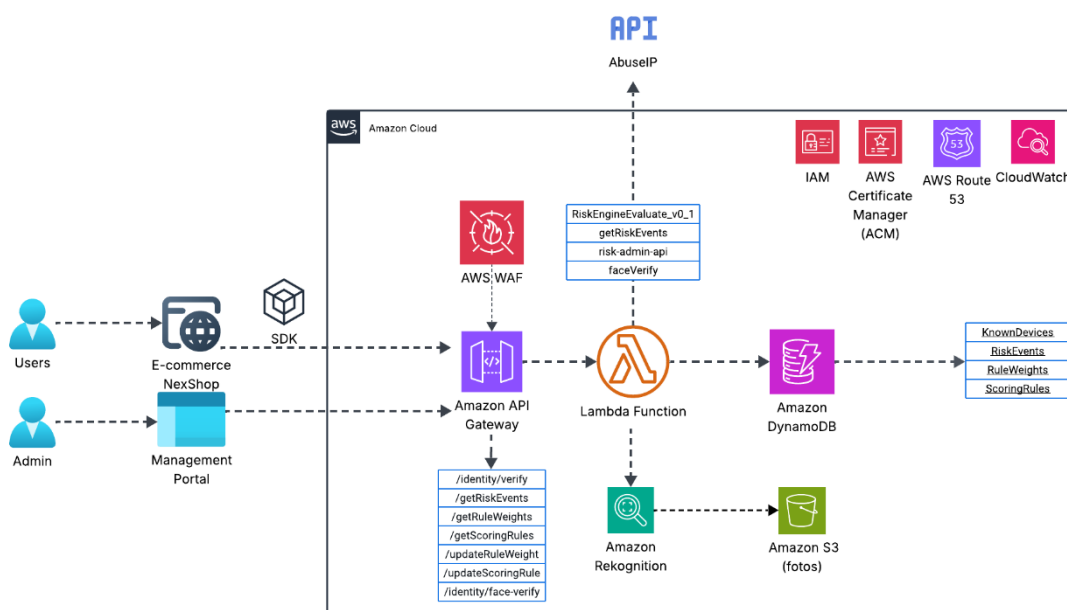
- Front-end (HTML, Java);
- SDK v1.2 (JavaScript);
- Backend (API em Python 3.13);
- AWS API Gateway (Method POST): envia dados ao servidor;
- AWS Lambda (backend em Python);
- AWS DynamoDB: serviço de banco de dados não relacional (NoSQL) que possui as seguintes tabelas: RiskEvents, Known Devices, RuleWeights, Scoring Rules.
- Dashboard (Backend: Python, Front-end: JavaScript);
- Route 53 (Registro e Gerenciamento do domínio);
- AWS WAF (Proteção de ataques na borda);
- AbuseIPDB API (external API);
- Certificate Manager (Criação e gerenciamento dos certificados SSL das API's);
- Cloud Watch (Registro de todos os logs);
- IAM (Gerenciamento de identidade dos usuários da AWS);
- AWS S3 (Armazenamento das imagens/fotos dos usuários);
- Amazon Rekognition (CompareFaces para verificação de identidade dos usuários).

Na imagem a seguir, é possível observar a arquitetura atualizada da solução, incluindo o que foi apresentado na fase 1 e fase 2 — Amazon API Gateway e Lambda Function SDK v1, front-end atualizado —, assim como o que foi introduzido na fase 3 — SDK v1.1, front-end

NEW GROUP LABS

atualizado, expansão da arquitetura com serviços gerenciados da AWS e o que foi proposto para a fase 4 — a inclusão de reconhecimento e registro facial.

Figura 49 – Arquitetura da solução “Score Trust” atualizada



Fonte: Os autores (2025)

- **Segurança e controle de acesso:** AWS WAF, IAM, ACM.
- **Gerenciamento de APIs e execução de lógica de negócio:** Amazon API Gateway e AWS Lambda.
- **Persistência de dados:** Amazon DynamoDB.
- **Monitoramento e DNS:** CloudWatch e Route 53.
- **Integração externa:** API AbuselP para validação de IPs suspeitos
- **Amazon S3 e Rekognition:** S3 para o armazenamento das imagens enviadas pelos usuários e Rekognition efetua a análise das imagens.

NEW GROUP LABS

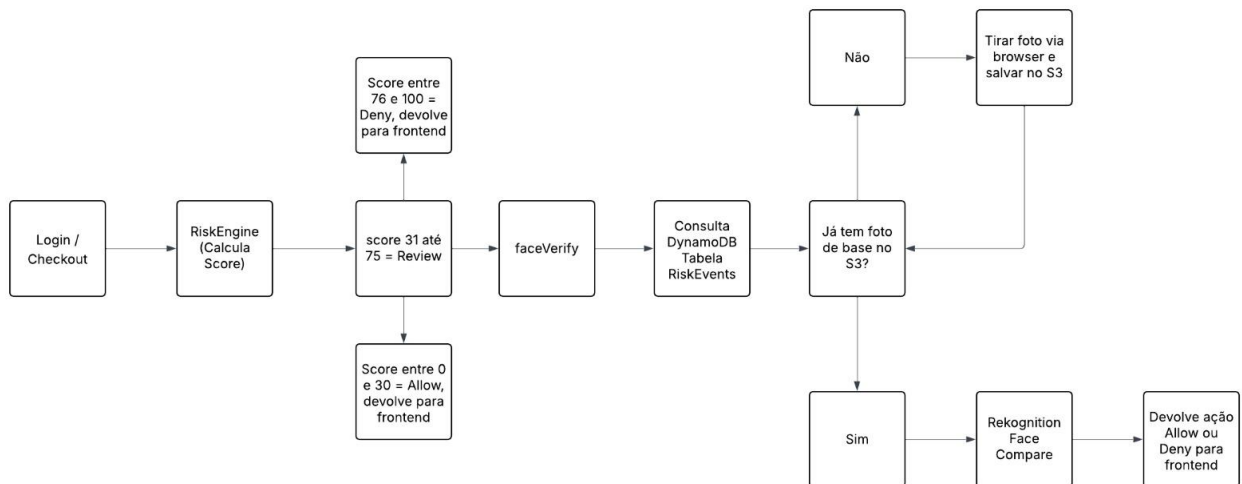
7.2. DEMO DA SOLUÇÃO – FASE 4

A partir de um estudo realizado na fase 1, juntamente com as implementações práticas da funcionalidade nas fases 2 e 3, o time evoluiu o seu projeto para a versão final, atualizando o seu SDK para a versão 1.2 e adicionando a API /identity/face-verify para o armazenamento de imagens e verificação biométrica.

Nesta fase, o SDK começou a recolher informações ainda mais detalhadas, como uma foto para o primeiro cadastro do usuário na plataforma, que servirá de base para outros acessos através do mesmo e-mail, assegurando uma entrada segura através de biometria facial.

A fim de demonstrar de forma mais clara a funcionalidade da solução, a equipe elaborou um fluxo da aplicação:

Figura 50 – Fluxo da aplicação do “Score Trust”

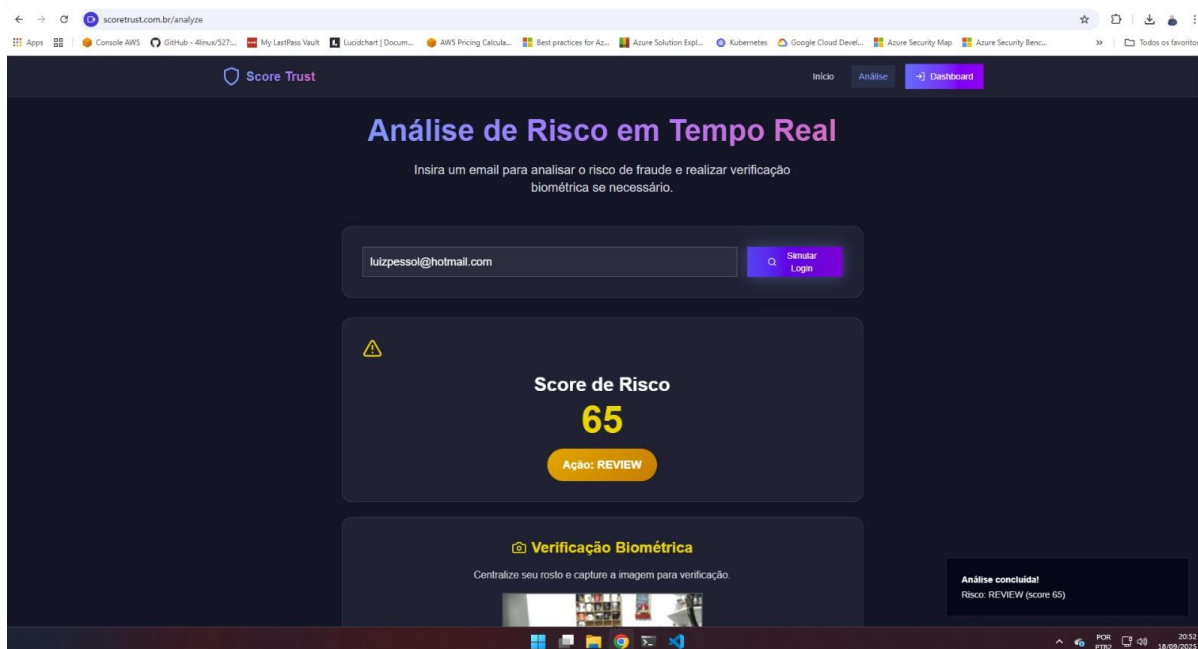


Fonte: Os autores (2025)

A nova funcionalidade do Score Trust é apresentada de imediato na área de testes. A partir da inserção de um novo e-mail e um dispositivo não reconhecido, a tela irá solicitar a autorização para o uso da câmera do utilizador.

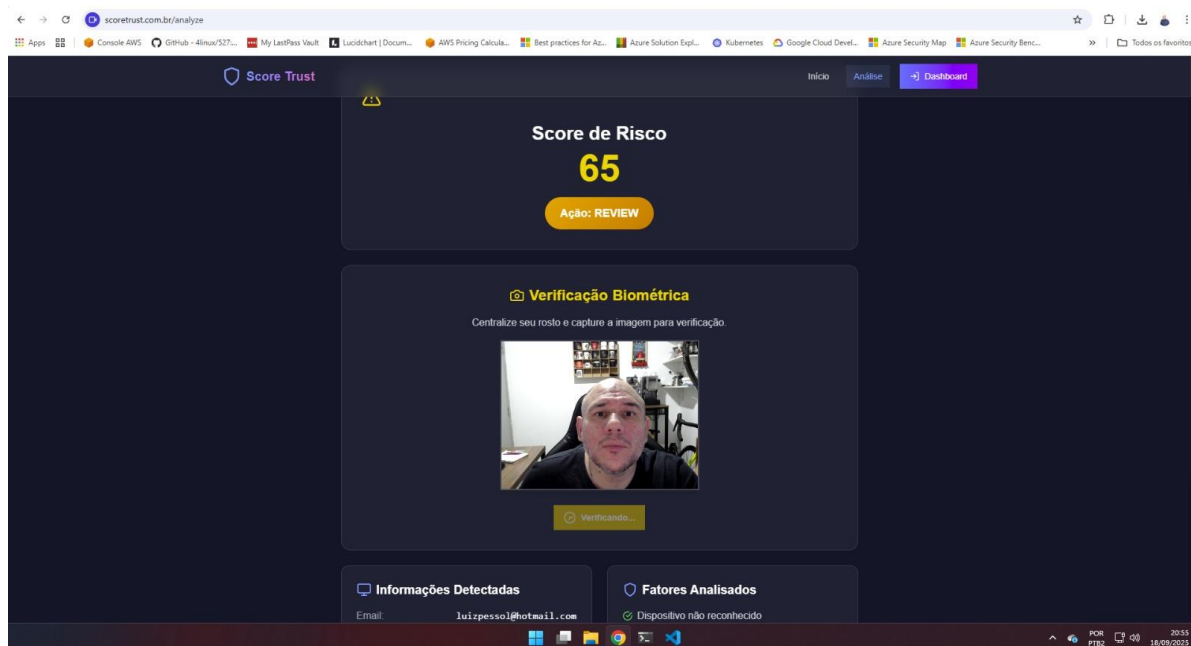
NEW GROUP LABS

Figura 51 – Primeira tentativa de login na plataforma <https://scoretrust.com.br>, dispositivo não conhecido + sem foto de base + sem verificação biométrica da foto



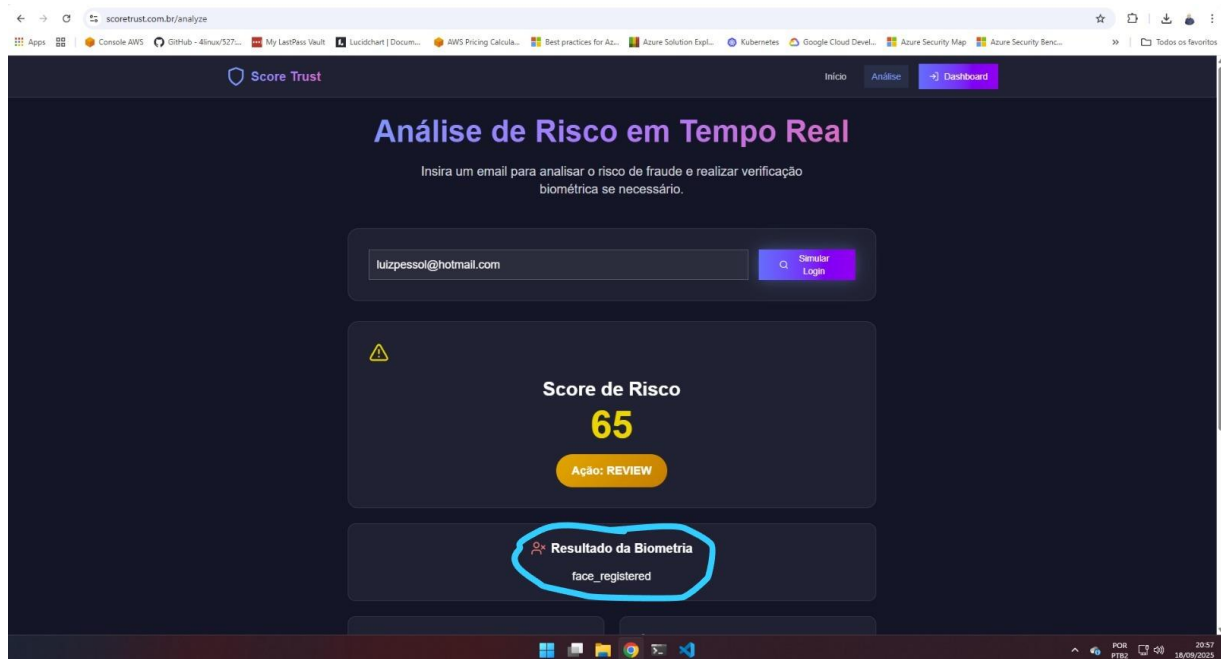
Fonte: Os autores (2025)

Figura 52 – Captura da foto de base para verificação biométrica



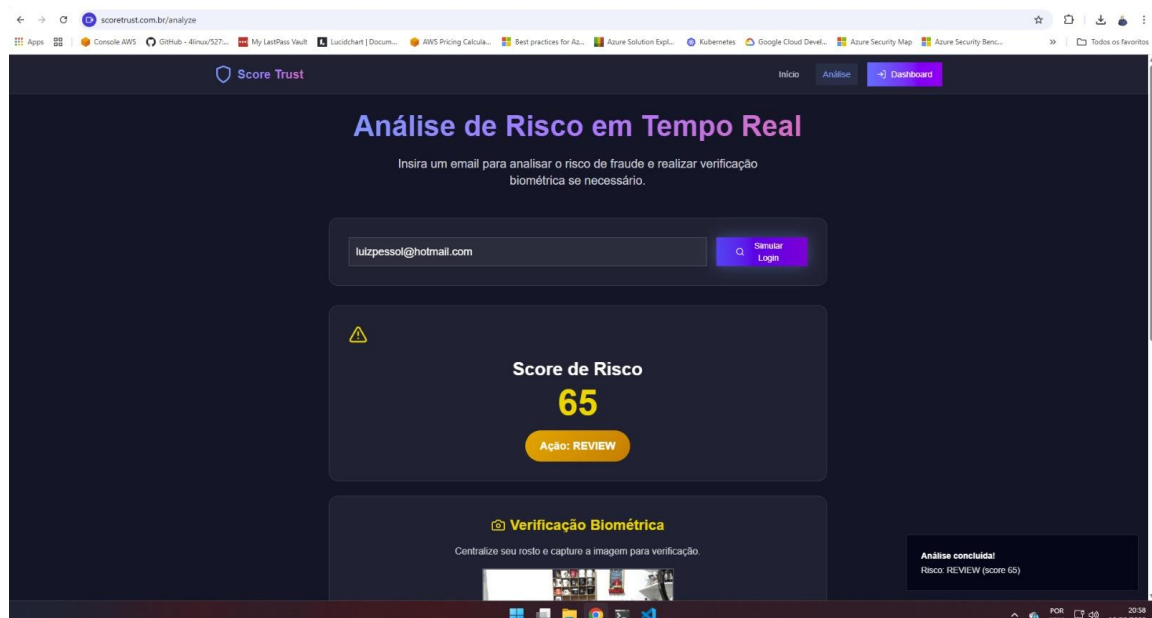
Fonte: Os autores (2025)

Figura 53 – Foto de base enviada para a plataforma



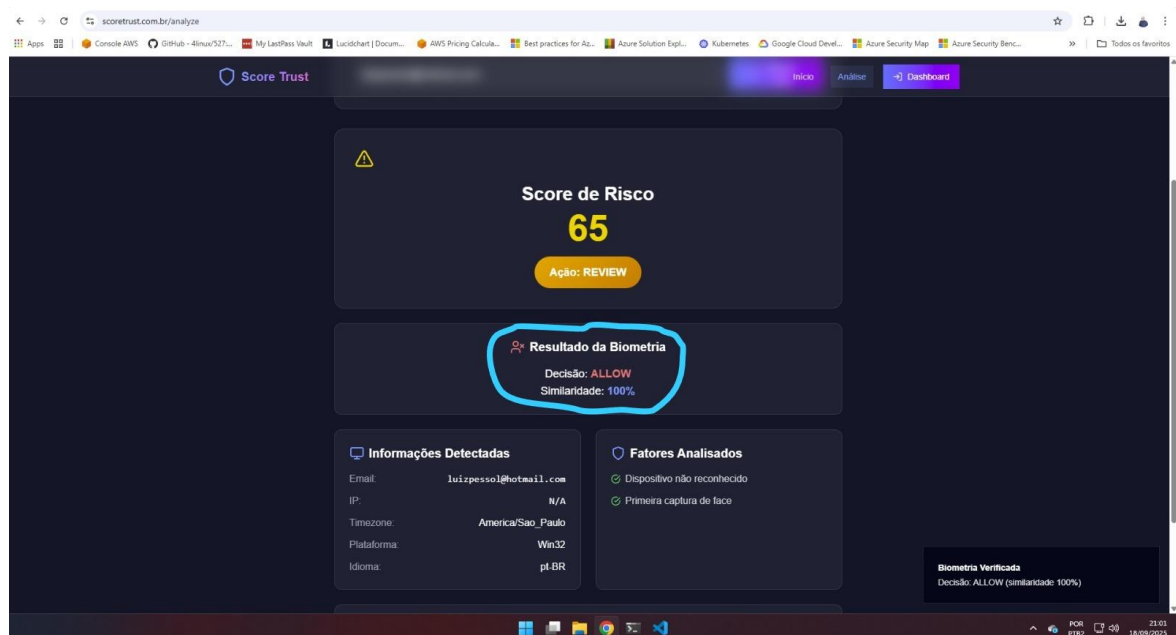
Fonte: Os autores (2025)

Figura 54 - Nova tentativa de login, verificação biometria facial necessária.



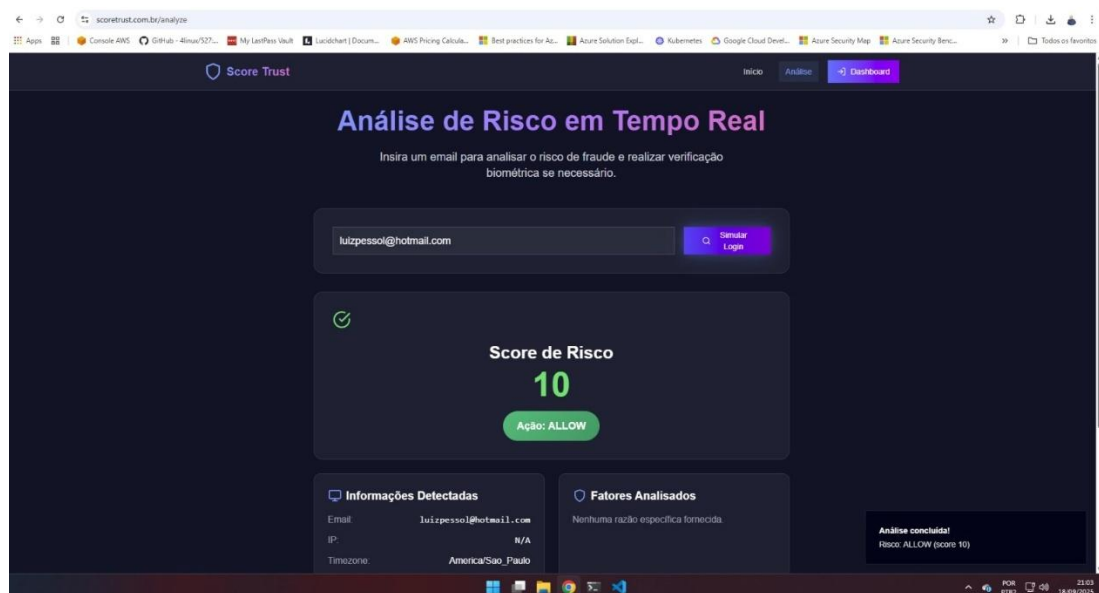
Fonte: Os autores (2025)

Figura 55 – Biometria facial verificada com sucesso.



Fonte: Os autores (2025)

Figura 56 – Novo acesso, o dispositivo foi reconhecido a foto de base registrada e a verificação da biometria facial realizada.



Fonte: Os autores (2025)

Como evidenciado nas figuras anteriores (51 – 57), a solução conta com uma camada adicional de segurança, exigindo uma foto para registro no banco de dados da Score Trust. A fotografia é comparada com imagens de futuros acessos, buscando máxima compatibilidade

NEW GROUP LABS

entre os traços característicos. Na tabela de eventos DynamoDB (figura 58), os dispositivos reconhecidos são registrados.

Figura 57 – Eventos registrados no DynamoDB, dispositivos conhecidos.

id (String)	action	biometricRequired	biometricSimilarity	biometricVerified	country	device_hash	device_name	email
af1486c8-2f15-42f4-...	REVIEW	true		false	BR	2d2f8068c27...	Mozilla/5.0 (...)	luizpessol@...
5449e085-7219-474f-...	REVIEW	true		false	BR	2d2f8068c27...	Mozilla/5.0 (...)	luizpessol@...
90324792-3148-4c01-...	REVIEW	true		false	BR	2d2f8068c27...	Mozilla/5.0 (...)	luizpessol@...
46b046cd-ba98-4ac8-...	ALLOW	false	99.999977118164	true	BR	2d2f8068c27...	Mozilla/5.0 (...)	luizpessol@...
586a2652-080e-4dba-...	ALLOW	false		false	BR	2d2f8068c27...	Mozilla/5.0 (...)	luizpessol@...

Fonte: Os autores (2025)

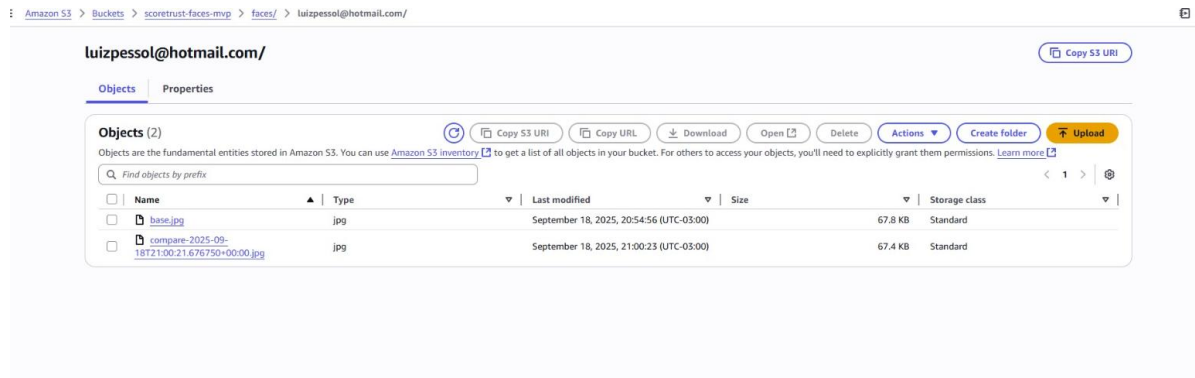
Figura 58 – AWS S3 com usuário que fizeram registro da foto de base e fotos da verificação biométrica facial.

Name	Type	Last modified	Size	Storage class
anacarolinaaraujo.paro@teste.com.br/	Folder	-	-	-
anacarolinaaraujoteste@outlook.com/	Folder	-	-	-
camille.costa@hyundai-brasil.com/	Folder	-	-	-
demetrior@drpp.com.br/	Folder	-	-	-
demetrior@outlook.com.br/	Folder	-	-	-
demetrior@terluss.com.br/	Folder	-	-	-
lpessol@gmail.com/	Folder	-	-	-
luizpessol@hotmail.com/	Folder	-	-	-
teste@example.com/	Folder	-	-	-

Fonte: Os autores (2025)

NEW GROUP LABS

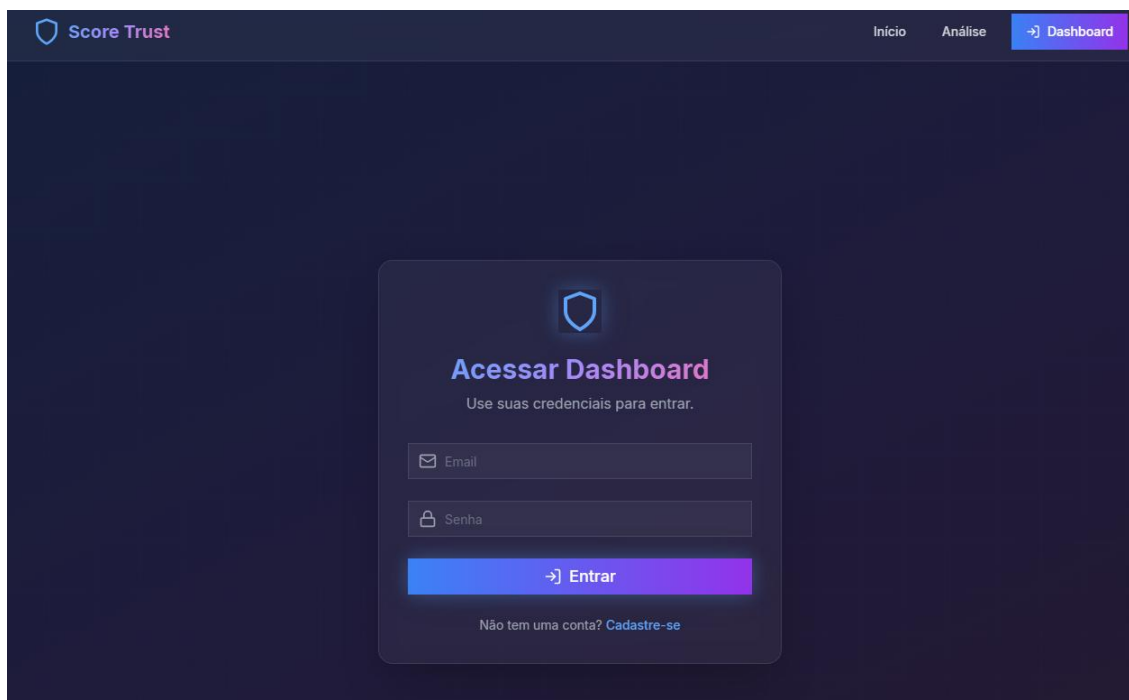
Figura 59 – Bucket do AWS S3 com a foto de base + verificação biométrica facial registrada



Fonte: Os autores (2025)

Contando com os serviços AWS, a solução registra as fotos do usuário em um banco de dados, assimilando cada credencial com a respectiva imagem, o e-mail e um hash característico do seu acesso. Para garantir que nenhuma informação está sendo utilizada por um ator mal-intencionado, a equipe adicionou uma tela de login para administradores na dashboard (figura 60).

Figura 60 – Tela de login para administradores



Fonte: Os autores (2025)

8. Conclusão

Em vista da proposta apresentada pela Hakai Security, a equipe New Group Labs realizou um estudo para viabilizar uma solução simples e leve que considerasse o conceito de Software Development Kit (SDK), buscando um projeto que avaliasse o login de usuários em um marketplace que estava sofrendo com constantes perigos cibernéticos. A partir de uma visão antifraude e baseada nas solicitações da Hakai Security — que envolvem o uso de scores, back-end, front-end —, a equipe desenvolveu a proposta *Score Trust*, uma funcionalidade cuja iniciativa é averiguar os acessos indevidos na plataforma de compras da NexShop, promovendo maior segurança para a empresa e seus clientes.

Com uma perspectiva estratégica, a New Group Labs desenvolveu uma dashboard funcional, permitindo a visualização dos dados das suas APIs e proporcionando ao cliente a oportunidade de realizar testes e solicitar um plano de utilização adequado para a realidade de cada negócio.

Através de testes simulados, a equipe comprovou a funcionalidade de seus códigos, frameworks e APIs, conforme evidenciado nas capturas de tela apresentadas, buscando evoluir seu projeto em uma solução sólida, segura e eficaz para atender às necessidades da NexShop no campo antifraude.