Blueprint para Integração de Sistemas com o Ping Identity

Controle de versão do documento:			
Versão	Item Revisado	Data	Responsável
1	Versão Inicial	26/05/2023	Marcos Damasceno
2	Inclusão de requisitos	21/03/2024	Marcos Damasceno
3	Inclusão de requisitos	30/06/2024	Marcos Damasceno
4	Inclusão de requisitos	22/05/2025	Marcos Damasceno
5	Inclusão de requisitos	03/11/2025	Marcos Damasceno

Nome	Órgão	E-mail
Marcos Damasceno	SIT	marcos.damasceno@claro.com.br
Guilherme Fidelis	NETBR	guilherme.fidelis@netbr.com.br
Rafael Campos	NETBR	rafael.campos@netbr.com.br

Sumário

1 OBJETIVO

O propósito deste documento é prover os insumos técnicos necessários, para que os centros de competência e os desenvolvedores dos sistemas da Claro, possam fazer as adequações e parametrizações necessárias para que seus respectivos sistemas possam ser integrados ao Ping Identity, nova plataforma de orquestração de processos de federação, autenticação de acesso e MFA (Múltiplo Fator de Autenticação) da Claro.

O Ping Identity é o substituir o Gateway de Autenticação atual de acesso a sistemas corporativos (OAM), e foi adquirindo como nova solução de mercado de Access Manager e Orquestração de Autenticação, de forma que possa se integrar ao ambiente corporativo da Claro e demais empresas do grupo, como um orquestrador/motor de todos os processos de autenticação de contexto corporativo (colaboradores internos e prestadores de serviços internos e externos).

Este orquestrador precisará se adequar a uma arquitetura meta já definida, onde o mesmo atenda como receptor do API Gateway Corporativo, tratando as requisições de acesso aos mais variados sistemas e provendo processo de autenticação segura, utilizando soluções de MFA e Biometria de acordo com o contexto de risco de acesso e segmento de atendimento prestado pelo usuário e principais operações de negócio corporativo.

2 DETALHAMENTO DO ESCOPO

A integração de aplicações (Service Providers) ao PingFederate (IdP) desempenha um papel fundamental na construção de sistemas seguros e eficientes de autenticação e autorização.

Este documento detalha as informações necessárias para a integração de aplicações com o PingFederate usando os protocolos OpenID Connect, SAML ou utilizando o método REST API.

Atualmente o OpenID Connect (OIDC) é protocolo mais moderno para autenticações, sendo ele o mais indicado para quando existir compatibilidade das aplicações.

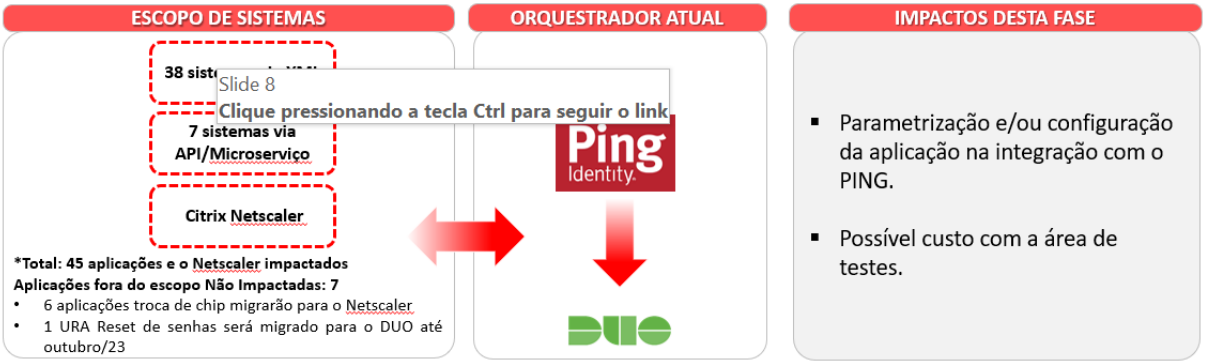
3 MIGRAÇÃO COM IMPACTO REDUZIDO

Nesta fase, todos os sistemas que estão integrados a solução atual de Access Manager (OAM/OAM), serão migrados AS-IS para o Ping Identity, utilizando os mesmos protocolos e métodos de integração que são utilizados atualmente na solução da Oracle.

Desta forma, o impacto nos centros de competência será reduzido, uma vez que nesta fase não haverá customização de códigos ou alteração no core das aplicações, impactando somente no nível de configuração/parametrização.

Nesta primeira versão do documento, serão disponibilizadas as informações necessárias para que os centros de competência donos dos sistemas impactados, possa estudar a solução e iniciar as parametrizações do lado dos provedores de serviço (service providers), que irão delegar seus processos de autenticação para o Ping. Nesta fase, será disponibilizado a documentação necessária para viabilizar as integrações com os sistemas que utilizam na solução atual, os protocolos abaixo:

- Federação SAML
- Federação OIDC
- REST API



SISTEMA	TIPO INTEGRAÇÃO	USUÁRIOS	GESTOR	CENTRO COMPETÊNCIA

JIRA ITSM	Federação SAML	44.348	FERNANDA SANTOS HUET DE BACELLAR	TI - GER OPERACOES E SERVICE DESK
HCM - CONECTE-SE	Federação SAML	35.000	PATRICIA HELENA MONTEIRO ALVES PEREIRA	TI - RH
ORHGANIZA ADM	Federação SAML	35.000	PATRICIA HELENA MONTEIRO ALVES PEREIRA	TI - RH
SALESFORCE-CEC	Federação SAML	30.163	JULIANA ALVES CASTRO PEREZ	TI - COORD VENDAS MOVEL
IW	Federação SAML	29.614	CARLOS RENATO ROCHA BUENO	TI - GER DE BI COMISION
SGD	Federação SAML	26.602	FELLIPE AUGUSTO DA CRUZ	87MZBR5D31 DIR DE REDE EXTERNA MZBR5
NICE CC1	Federação SAML	15.321	PITER SALOMAO ELIAS DA SILVA	TI - INFRA CALLCENTER
NICE CC2	Federação SAML	15.321	PITER SALOMAO ELIAS DA SILVA	TI - INFRA CALLCENTER
TOA	Federação SAML	3.715	FABIO MENEZES AMARAL	TI - M D O CAPEX TIPO G
PORTAL SCP	Federação SAML	1.844		
PORTAL_SIGO	Federação SAML	1.063	MARCIA MARIA MENDES NICOLICH	10MZBR6N28 SEGURANCA CORP
PCO - EXPERIANSP	Federação SAML	725	WALDYR LOURENCO DE FREITAS JUNIOR	TI - GER SISTEMAS POS PAGO
OCTANE	Federação SAML	508	ANA CRISTINA CHAGURI MADEIRA	10RJRE6A04 M D O CAPEX TIPOC ATIVO IMO
TOA RE	Federação SAML	477	FABIO MENEZES AMARAL	TI - M D O CAPEX TIPO G
SUBSÍDIOS MELHORIAS TRABALHISTAS	Federação SAML	350		
SUBSÍDIO TRABALHISTA DE TERCEIROS (BPM)	Federação SAML	350		
SAP ARIBA	Federação SAML	218	CARLOS RENATO ROCHA BUENO	TI - GER DE BI COMISION
PORTAL_SIXBELL_FEM	Federação SAML	137	CRISTIANO ALVES HARADA	10MZBR7H12 GER PLATAFORMAS DE ENGENHARIA
PORTAL_SIXBELL_TECNICO_ENG	Federação SAML	82	CRISTIANO ALVES HARADA	10MZBR7H12 GER PLATAFORMAS DE ENGENHARIA
PORTAL SIXBELL TECNICO	Federação SAML	82	CRISTIANO ALVES HARADA	10MZBR7H12 GER PLATAFORMAS DE ENGENHARIA

SAP COMMISSIONS	Federação SAML	32	CARLOS RENATO ROCHA BUENO	TI - GER DE BI COMISION
PORTAL GM	Federação SAML	12	CRISTIANO ALVES HARADA	TI - GER PLATAFORMAS DE IOT
TPKNOWLEDGE BLUEMIND	Federação SAML	2	ALESSANDRA GOMES MAZZETI	10MZBR6C48 GER TRIBUTOS FEDERAIS E RETENCOE
MARTECH	Federação SAML	1	RADAKIAN MAURITY SOUSA LINO	TI - GER MARTECH E ANALYTCS
DCI	Federação SAML	0		
PORTAL PIPEFY	Federação SAML	0		
PLANEJAMENTO LOGÍSTICO - SAP	Federação SAML	0		
QUALTRICS CX	Federação SAML	0	RADAKIAN MAURITY SOUSA LINO	TI - GER MARTECH E ANALYTCS
PORTAL RADAR	Federação SAML	0		
SISTEMA PLUGADOS	Federação SAML	0		
GRC PORTIFÓLIO	Federação SAML	0		
CONSUMO DE MATERIAIS	Federação SAML	0		
MVNO	Federação SAML	0	RAFAEL DA COSTA MOREIRA	TI - 10MZBR7S02 DIR MKT CLARO FLEX
NUVEM CISCO	Federação SAML	0		
NOVA LOJA ONLINE	Federação SAML	0		
JASPER GM AIRLINQ	Federação SAML	0	CRISTIANO ALVES HARADA	TI - GER PLATAFORMAS DE IOT
JURÍDICO DIGITAL CONSUMERISTA (SERVICENOW)	Federação SAML	0		
DATORAMA	Federação SAML	0	RADAKIAN MAURITY SOUSA LINO	TI - GER MARTECH E ANALYTCS
BLIPDESK	Federação OIDC	7.744	PITER SALOMAO ELIAS DA SILVA	TI - INFRA CALLCENTER
PORTAL SGE	Federação OIDC	59	EDUARDO GOMES DA SILVA CARDOSO	LE GER OPERACOES
FAP ONLINE	Federação API OAUTH	35.000	PATRICIA HELENA MONTEIRO ALVES PEREIRA	TI - RH
ORHGANIZA MOBILE	Federação API OAUTH	35.000	PATRICIA HELENA MONTEIRO ALVES PEREIRA	TI - RH

GESTOR DE COBRANÇA (GEVENUE)	Federação API OAUTH	6.500	RAFAEL RUSSO	10MZBR5H21 GER SIST BSS FAT CLARO FIXO E LD
PPM	Federação API OAUTH	300	ANA CRISTINA CHAGURI MADEIRA	10RJRE6A04 M D O CAPEX TIPOC ATIVO IMO
OMNI CHANNEL CPC	Federação API OAUTH	0	EDUARDO BASTOS PADILHA	TI - GER SISTEMAS OSS

4 BENEFÍCIOS ESPERADOS

- Agilidade no processo de integração de sistemas novos e legados, com o IDP de autenticação e MFA da Claro;
- Alta disponibilidade e resiliência de arquitetura.
- Funcionalidades novas, indisponíveis no produto atual, exemplo: login baseado em contexto do usuário e proteção de cookies de sessão de logon.
- Geração de logs de auditoria e rastreabilidade de acessos.

5 REQUISITOS DE IMPLEMENTAÇÃO VIA PING IDENTITY

Nesta fase, todos os sistemas que estão integrados a solução atual de Access Manager (OAM/OAM), serão migrados AS-IS para o Ping Identity, utilizando os mesmos protocolos e métodos de integração que são utilizados atualmente na solução da Oracle.

Desta forma, o impacto nos centros de competência será reduzido, uma vez que nesta fase não haverá customização de códigos ou alteração no core das aplicações, impactando somente no nível de configuração/parametrização.

6 INTEGRAÇÃO VIA OPENID (OIDC):

A integração entre o PingFederate e aplicações utilizando o protocolo OpenID Connect (OIDC) permite estabelecer um fluxo de autenticação seguro e interoperável.

O PingFederate atua como um provedor de identidade centralizado, permitindo que os usuários acessem várias aplicações usando suas credenciais de login em um único ponto de autenticação.

As informações desta seção são necessárias para integração da aplicação com o PingFederate utilizando protocolo OIDC.

6.1 Informações fornecidas pelo PingFederate (IdP)

Todos os parâmetros abaixo devem ser configurados no módulo de SSO do service provider, para posterior tratamento do access token enviado pelo IdP. Estes valores são padrões de uso de fluxos de autenticação, baseados em RFC.

Tabela 1 - Informações que deverão ser fornecidas pelo IdP (Ping).

Dado	Descrição	Exemplo
Issuer	Entidade que emite identificadores únicos e identifica o provedor de identidade (IdP) no contexto do OpenID Connect.	https://ipchml.claro.com.br
Authorization Endpoint	URL no IdP onde as aplicações clientes podem iniciar o processo de autenticação e solicitar autorização para acessar recursos protegidos em nome do usuário.	https://ipchml.claro.com.br /as/authorization.oauth2
Token Endpoint	URL no IdP onde as aplicações clientes podem trocar um código de autorização (Authorization Code) obtido no Authorization Endpoint por um token de acesso (Access Token) e, possivelmente, um	https://ipchml.claro.com.br /as/token.oauth2

	token de atualização (Refresh Token).	
Userinfo Endpoint	URL no IdP onde as aplicações clientes podem solicitar informações adicionais sobre o usuário autenticado.	https://ipchml.claro.com.br/idp/userinfo.openid
Client ID	Identificador único atribuído à aplicação cliente (SP) pelo IdP.	Ex.: gevenue
Client Secret	Senha atribuído à aplicação cliente (SP) pelo IdP.	XPTOXyZ
Certificado	Os certificados digitais usados para assinatura e/ou criptografia.	Anexo

6.2 Informações necessárias da aplicação (Service Provider)

Todos os parâmetros abaixo devem ser configurados no módulo de SSO do IdP (identity provider), para prover os redirecionamentos entre as URLS e estabelecer confiança entre as partes, assim como o envio de atributos de autenticação (Claims).

Tabela 2 - Informações que deverão ser fornecidas pelo SP (Sistema Centro de Competência).

Dado	Descrição
URL de Callback	URL utilizada para direcionar o usuário de volta à aplicação após o processo de autenticação no provedor de identidade.

Claims	Representa uma declaração sobre um usuário autenticado, que é fornecida pelo provedor de identidade (IdP) no formato de um token de identidade (JWT - JSON WebToken), que o SP necessita. As claims contêm informações específicas sobre o usuário, como seu nome, endereço de e-mail, papel ou função, data de nascimento, entre outras.

7 INTEGRAÇÃO VIA OPENID (OIDC) FLUXO COM AUTHORIZATION CODE:

7.1 Objetivo:

Este documento possui o objetivo de oferecer um descritivo da integração de aplicação via OpenID no fluxo Authorization Code, será apresentado os parâmetros, header, body) das chamadas OpenID.

A partir destas configurações realizadas, todo processo de Autenticação e Autorização de usuários deverá ser controlado pelo Ambiente de IAM.

Para iniciar será demonstrado todo o fluxo do Authorization Code, detalhando cada step e sua finalidade, para posteriormente apresentar todas as chamadas necessárias para integração utilizando esse protocolo.

6.2 AuthN - PingFederate:

Abaixo iremos apresentar uma visão geral com todo o fluxo que demonstra a implementação do protocolo OpenID seguindo o fluxo Authorization Code.

Conforme Ilustrado na Figura 1, a jornada inicia a partir do momento que o usuário acessa a aplicação via browser (1).

A aplicação recebe a chamada, monta a URL e nesse momento a chamada será redirecionada para o PingFederate (2), com seus devidos parâmetros.

O PingFederate devolve no browser a tela de autenticação, o End-User insere suas credenciais e é realizado a validação das credenciais, para que no sucesso seja disponibilizo o Code, na URL (3).

Em posse do Code a aplicação deverá chamar via API o PingFederate (com os parâmetros necessários), realizando a troca do Code para obter o Access Token (4).

Nessa etapa é realizada a validação do Code no PingFederate, se OK o Access Token é disponibilizado para aplicação (5).

De posse do Access Token a aplicação pode realizar o introspect para validar e abrir o Token e assim coletar as informações do usuário em questão que realizou a Autenticação (6).

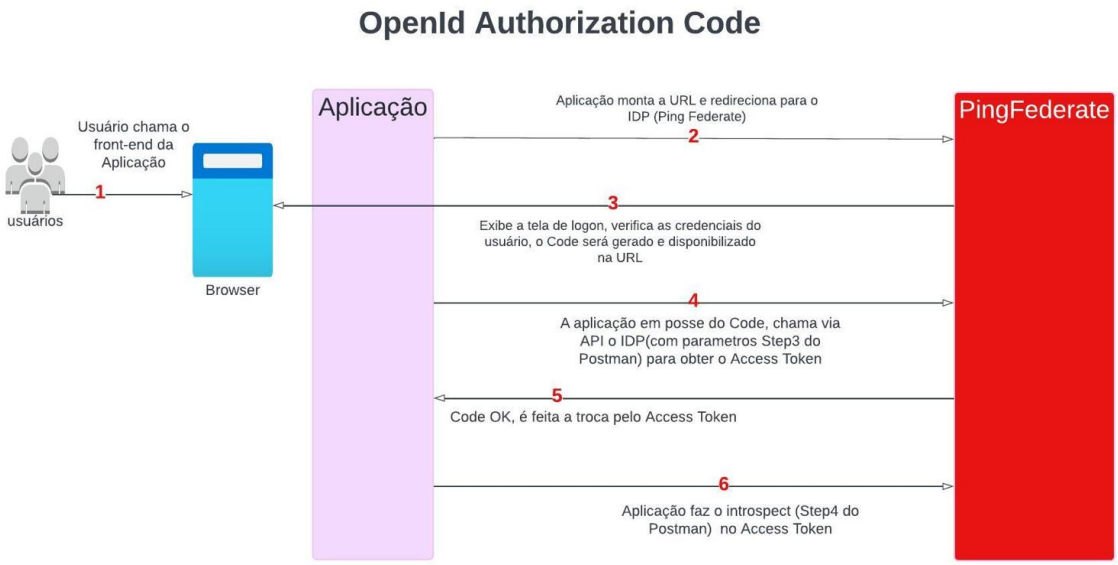


Figura 1 - Visão geral do Fluxo Authorization Code

7.2 Descritivo Das Chamadas:

Nos tópicos a seguir será demonstrado cada chamada que será necessário ser realizado pela aplicação.

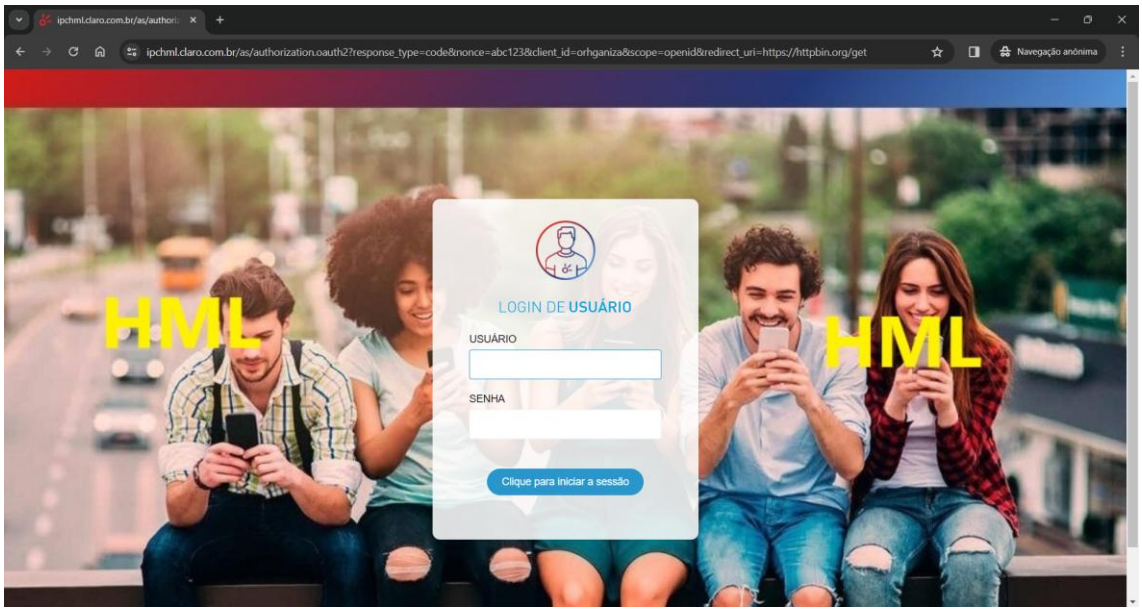
7.2.1 Redirect para o Ping Federate (step 2 da Figura 1):

Para dar início ao processo de autenticação será necessário realizar um GET para o recurso /as/authorization.oauth2. Os parâmetros descritos abaixo são necessários:

Chamada	
GET https://ipchml.claro.com.br/as/authorization.oauth2?response_type=code&client_id={{client_id}} &scope=openid&redirect_uri={{redirect aplicação}}	
Parâmetros	
client_id	{{client_id}}
response_type	code
scope	openid
redirect_uri	{{redirect aplicação}}

Exemplo de resultado esperado:

O PingFederate irá receber a chamada com seus devidos parâmetros e estando OK/Correto irá exibir a tela de login no browser do usuário.



Após a autenticação do usuário, o Code será gerado e disponibilizado na URL para a aplicação.

7.2.2 Get Access Token (step 4 da Figura 1):

Com um authorization code recebido via URL pelo PingFederate, para se obter um acess token, deve-se realizar um POST no recurso /as/token.oauth2 e enviar os parâmetros descritos abaixo:

Chamada	
POST https://ipchml.claro.com.br/as/token.oauth2	
Parâmetros	
client_id	{{client_id}}
grant_type	authorization_code
code	{{code}}
redirect_uri	{{redirect aplicação}}
client_secret	{client_secret}}

A resposta desta chamada deverá ser um Status OK 200 e irá conter um access token ("access_token"), um id token ("id_token"), o tipo de token ("token_type") e o tempo de expiração ("expires_in") desses tokens. Conforme exemplo abaixo:

Exemplo de resposta Get Access Token:

Resposta (Exemplo) Status 200 OK


```
{
"access_token":
"eyJhbGciOiJSUzI1NiIsImtpZCI6ImsxIiwicGkuYXRtIjoiYzZoMS
J9.eyJzY29wZSI6Im9wZW5pZCIsImNsaWVudF9pZCI6Im9tbmkiLCJV
c2VybmFtZSI6IlAyMDIwMjAwIiwiYXBwTmFtZSI6Im9tbmkgSW50ZXJ
uYWwiLCJwaS5zcmkiOiJWczlBUk1PVURqVGtObkhUWkg2MnpQX3h5a1
UuLnp0TmsiLCJvdU5hbWUiOiJvbW5pSW50ZXJuYWwiLCJleHAiOjE3M
DgwNjA1OTN9.EYi92OXWzDO5XgPzXkRSZ_VwFVNuJUcqOpmU2GH38xc
BRjq9-
ifx0tiwwt9IgjbQVphzU3BACTyJrA_nxsD2lirlR1vOU2IKImVMm3qZ
```

```
9HkAk3j_qcK0B0EfKIoroV0j7pzahTKa_BXQ_l1JLJ-
oEnw5uAa5t5_pzNTxM4-
TYOS6Xe24Nuv56bFMB_aSRa5p8X9GOCV76UOp956qxKCbeYotFjh_Zz
EHHysoN0bQX1Vx-
Rcj8FErutBdfp2LkS_s3pDHLH3bMri7c_NIzBOtzwCYL8NCF9SphQhv
UNi2m1wb9HCgvyYuW-_UGc6rY86pDUIxHdBZdVng1s-OUnJjhQ",
"id_token":
"eyJhbGciOiJSUzI1NiIsImtpZCI6InZOUjNSU1piX1FUZFhoVmN5OT
FYVF9HLVp3SSJ9.eyJzdWIiOiJQMjAyMDIwMCIsImF1ZCI6Im9tbmki
LCJqdGkiOiI3dEhqMEN3SUxvcG93NzJkdGJIUm5EIiwiaXNzIjoiaHR
0cHM6Ly9pYW0tZnFhLXBmLmludGVybmFsLnRpbWJyYXNpbC5jb20uYn
IiLCJpYXQiOjE3MDgwNTMzNTYsImV4cCI6MTcwODA1MzY1NiwiYXV0a
F90aW1lIjoxNzA4MDUzMzU2LCJwaS5zcmkiOiJWczlBUk1PVURqVGtO
bkhUWkg2MnpQX3h5a1UuLnp0TmsifQ.Uz4kDsTMOq-
3aC4pO0TgG5PARJHwMFeGbcpyKjlkzw6-bGeThkk46C-
avGLJkL80xPaqkF2mSpQnlX7slzGfIPmbNyJQg2PLgFr9qcS5H2VYex
3s5yYVI6yTfUctD9RLV4KEbNwx_bb0RuQfA6hn4cJR8VARFSgDtSus1
0Bq82GSToVhBySgi7VFm4P9lYZpW9IeFIklbkyj8qVVidceYpHP9lvQ
wuxe9lj8zko2UJk5_uGBMRrRspsxVdiowkbFS8_tZ20VcwpC8tbk5x4
k1qKIM_AWw14QpLbPHDw_Npenr4UsrX8mQ7fBBSmLfRhIlt8iv73AW_
nVqspzpELiWw",
"token_type":       "Bearer",
"expires_in": 7199
}
```


7.2.3 Introspect (step 6 da Figura 1):

Em posse do access token a aplicação pode realizar uma verificação de um determinado token (por exemplo, um access token), deve-se realizar um POST no recurso /as/introspect.oauth2 e enviar no body a key token={{access_token}}, juntamente com o client_id e client_secret. Abaixo os parâmetros dessa chamada:

Chamada	
POST https://ipchml.claro.com.br/as/introspect.oauth2	
Parâmetros	
client_id	{{client_id}}
token	{{access_token}}

client_secret	{{client_secret}}

A resposta desta chamada deverá ser um Status OK 200 e irá conter um as informações do usuário, detalhes do Token e sessão. As informações do usuário serão disponibilizadas no token conforme solicitação/necessidade da Aplicação. Conforme exemplo abaixo:

Exemplo de resposta - Introspect:

Resposta (Exemplo) Status 200 OK


```
{
    "Username": "P2020200",
    "appName": "NOME APP",
    "scope": "openid",
    "pi.sri": "k-6xo14AsRJZ3rQW291-OuuP3M8..01Mw",
    "active": true,
    "token_type": "Bearer",
    "exp": 1708355431,
    "client_id": "NOME_CLIENT",
    "ouName": "NOME APP"
}
```

8 INTEGRAÇÃO VIA SAML:

SAML (Security Assertion Markup Language) é um padrão de autenticação aberto que torna possível o logon único (SSO) para aplicativos da web. O SSO permite que os usuários façam login em vários aplicativos e serviços baseados na Web usando um único conjunto de credenciais. Projetado para simplificar as experiências de logon do usuário, o SAML é mais amplamente usado em organizações empresariais e permite que os usuários acessem aplicativos e serviços pelos quais eles pagam.

Mais importante, as experiências de logon SAML são seguras porque as credenciais do usuário nunca são transmitidas. Em vez disso, eles são gerenciados por provedores de identidade (IdPs) e provedores de serviços (SPs):

- O IdP armazena todas as credenciais e informações do usuário necessárias para autorização e as fornece ao SP, quando solicitado. O trabalho dos IdPs é dizer: “Eu conheço essa pessoa e ela deve ter acesso a esses recursos”.

- O SP hospeda os aplicativos e serviços que os usuários desejam acessar. Esses aplicativos ou serviços podem incluir plataformas de e-mail, como Google ou Microsoft Office, ou aplicativos de comunicação, como Slack ou Skype. É trabalho dos SPs dizer: “Você pode acessar esses aplicativos ou serviços por um período de tempo especificado sem precisar se conectar novamente”.

Quando os usuários tentam acessar esses aplicativos ou serviços, o SP solicita ao IdP que verifique suas identidades. O IdP emite asserções SAML, ou tokens, que contêm as informações necessárias para confirmar as identidades do usuário, incluindo a hora em que as asserções foram emitidas e as condições que tornam as asserções válidas. Depois de recebidos, o SP dá aos usuários acesso aos recursos solicitados.

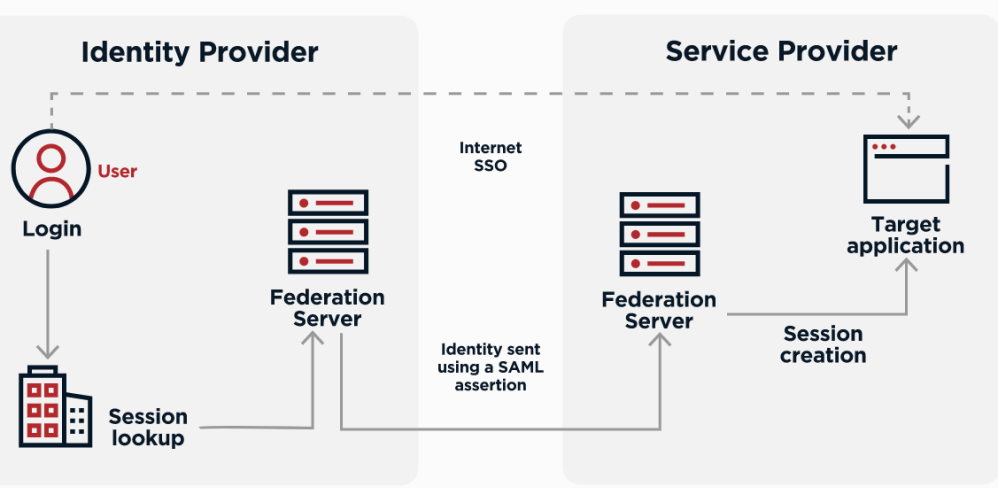
A integração entre o PingFederate e aplicações usando o protocolo SAML (Security Assertion Markup Language) permite estabelecer uma solução de Single Sign-On (SSO) para autenticação e autorização seguras. Neste protocolo o PingFederate também atua como um Identity Provider (IdP) centralizado, permitindo que os usuários acessem várias aplicações usando as suas credenciais de login em um único ponto de autenticação.

As informações desta seção são necessárias para integração da aplicação com o PingFederate utilizando protocolo SAML 2.0.

8.1 Como funciona o SAML?

SAML é uma estrutura baseada em XML, o que significa que é extremamente flexível, pode ser usado em qualquer plataforma e pode ser transmitido por uma variedade de protocolos, incluindo HTTP e SMTP. Os parceiros da federação podem optar por compartilhar qualquer informação que desejarem em uma asserção SAML, desde que as informações possam ser representadas em XML.

Um processo típico de autenticação SAML funciona da seguinte maneira:



1. O usuário se conecta e solicita acesso ao aplicativo ou serviço da Web de destino do SP.
2. O SP solicita informações de autenticação do usuário do IdP.
3. O IdP cria uma resposta assinada digitalmente no formato SAML que autentica o usuário. Essa resposta pode estar na forma de uma asserção SAML ou de um token SAML. A resposta também pode incluir informações sobre os privilégios do usuário.
4. O IdP assina a asserção e a envia para o SP.
5. O SP recupera a asserção, garante sua validade e autêntica o usuário.
6. O usuário acessa o aplicativo ou serviço do provedor de serviços.

As experiências de logon SAML são seguras porque as credenciais do usuário nunca são transmitidas. Asserções SAML, ou tokens, são usadas em seu lugar.

8.2 O que são asserções SAML?

Asserções SAML são documentos XML enviados de um IdP para um SP que identificam usuários, contêm informações pertinentes sobre eles e especificam seus privilégios no aplicativo ou serviço de destino. Essas mensagens também fornecem garantias de que as informações são válidas e especificam por quanto tempo os usuários podem acessar esses recursos sem precisar se conectar novamente.

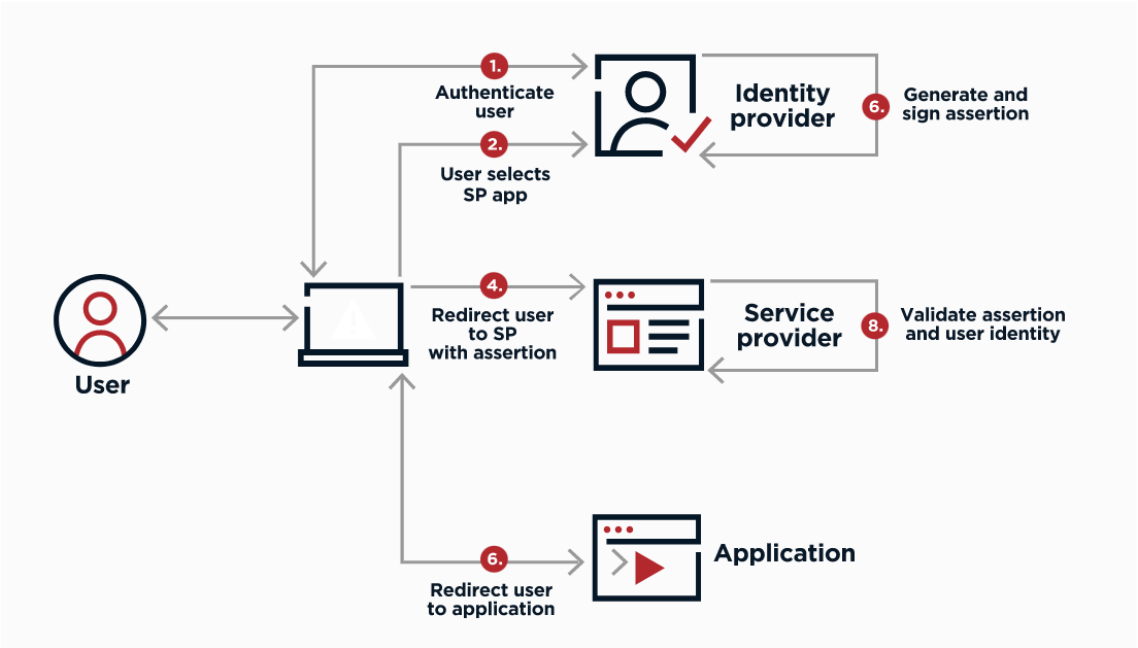
As asserções SAML são usadas principalmente para fins de autenticação, mas também podem incluir informações de autorização:

- **Asserções de autenticação:** identifique os usuários e forneça informações de logon, incluindo a hora em que o logon ocorreu e o método usado para autenticação.
- **Asserções de atributo:** contêm informações de atributo do usuário que existem nos diretórios IdP e SP e são correspondidas durante o processo de autenticação.
- **Asserções de autorização:** Indica se o usuário está autorizado a acessar o aplicativo ou serviço.

O SAML é amplamente usado em organizações empresariais para compartilhar informações de identidade entre sistemas IAM existentes e aplicativos da web. As formas como esses processos são implementados dependem das formas como os processos de logon são iniciados -- por meio do IdP ou do SP.

8.3 SSO iniciado por IdP

O SSO iniciado por IdP costuma ser encontrado em soluções de força de trabalho. As etapas envolvidas nesse tipo de processo são descritas no diagrama a seguir.

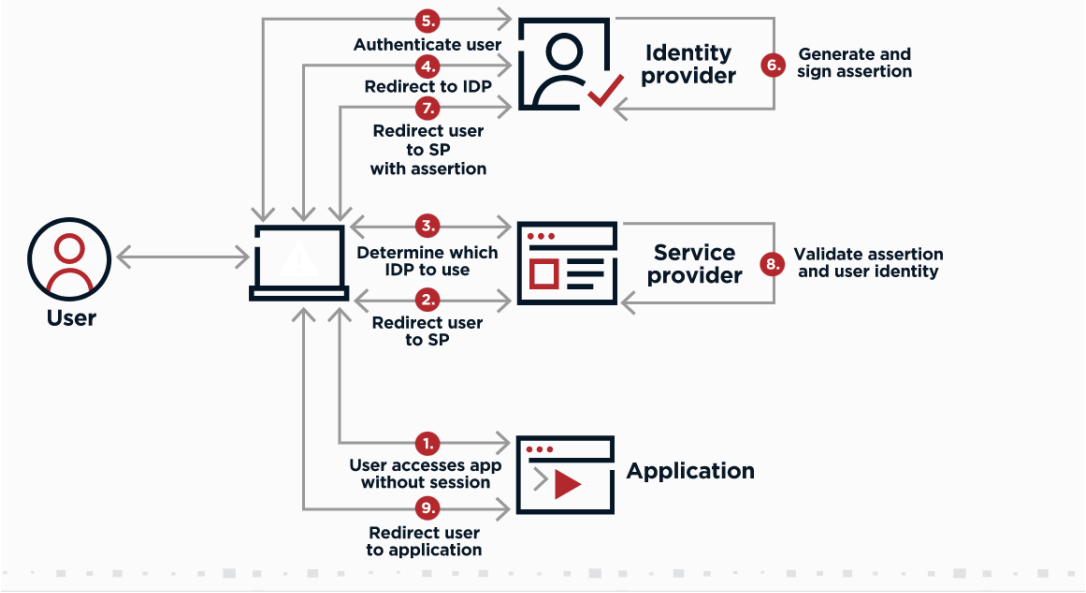


1. Um usuário se conecta ao sistema com um nome de usuário e senha e é apresentado a um catálogo de aplicativos que exibe ícones que representam os aplicativos e serviços baseados na Web que podem acessar.
2. O usuário clica em um ícone para acessar um desses aplicativos ou serviços.
3. O IdP cria e assina uma asserção SAML que inclui informações sobre a identidade do usuário, juntamente com qualquer outra informação de atributo que o IdP e o SP concordaram em compartilhar para autenticar os usuários.
4. O IdP envia a asserção ao SP por meio de um navegador ou envia uma referência à asserção que o SP pode usar para recuperar a asserção com segurança.
5. O SP valida a assinatura para garantir que a asserção SAML realmente veio de seu IdP confiável e que nenhum dos valores na asserção foi modificado. Ele também extrai as informações de identidade, atributo e autorização necessárias

para determinar se o acesso deve ser concedido e quais privilégios o usuário terá.
6. Os usuários acessam o aplicativo ou serviço.

8.4 SSO iniciado por SP

O SSO iniciado pelo SP começa quando um usuário tenta acessar um aplicativo ou serviço diretamente, em vez de autenticar primeiro por meio do IdP. As etapas envolvidas nesse tipo de processo são descritas no diagrama a seguir.



1. O usuário tenta acessar o aplicativo ou serviço.
2. O SP redireciona o usuário de volta ao IdP para ser autenticado e fornece ao usuário uma URL de recurso para acessar o aplicativo ou serviço após a autenticação.
3. O SP determina qual IdP o SP deve usar para autenticar o usuário. Os métodos comuns incluem:

    - O SP pode solicitar o endereço de e-mail do usuário e usar o domínio do e-mail, como bill@pingidentity.com, para identificar o IdP apropriado.
    - O SP pode exibir uma lista de IdPs com suporte e solicitar que o usuário selecione o apropriado.
    - A URL do recurso pode ser específica para um IdP.

- O SP pode ter colocado um cookie contendo informações do IdP no navegador do usuário na primeira vez que o usuário se conectou com sucesso a partir do IDP e usará essas informações para acesso subsequente.

4. O SP redireciona o usuário para o IdP apropriado.
5. O IdP autentica a identidade do usuário.
6. O IdP cria e assina uma asserção SAML baseada em XML que inclui informações sobre a identidade do usuário, junto com qualquer outra informação de atributo que o IdP e o SP concordaram em compartilhar para autenticar os usuários.
7. O IdP envia a asserção ao SP por meio de um navegador ou envia uma referência à asserção que o SP pode usar para recuperar a asserção com segurança.
8. O SP valida a assinatura para garantir que a asserção SAML realmente veio de seu IdP confiável e que nenhum dos valores na asserção foi modificado. Ele também extrai as informações de identidade, atributo e autorização necessárias para determinar se o acesso deve ser concedido e quais privilégios o usuário terá.
9. Os usuários acessam o aplicativo ou serviço.

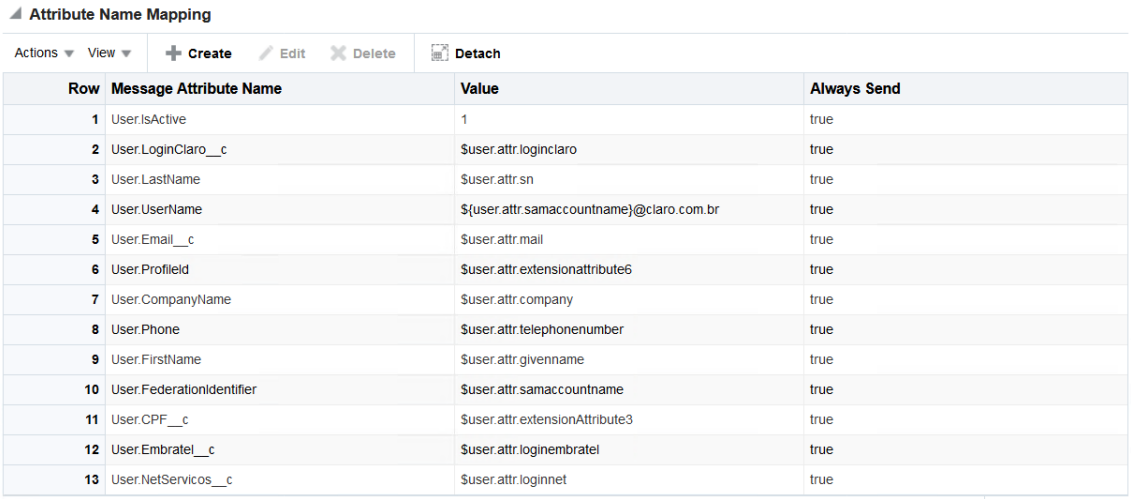
8.5 Informações fornecidas pelo PingFederate (IdP)

Em uma integração SAML (Security Assertion Markup Language), o metadata desempenha um papel fundamental, ele serve como um meio de comunicação padronizado para compartilhar informações sobre a configuração e os recursos de autenticação entre o IdP e o SP.

Tabela 3 - Informações que deverão ser fornecidas pelo IdP (Ping).

Dado	Descrição
Metadata (XML ou URL)	Documento XML que contém informações sobre as entidades envolvidas na integração, incluindo Entity ID, certificados, URL de endpoints, atributos, ACS e SLO.
Certificado	Os certificados digitais usados para assinatura e/ou criptografia das mensagens SAML.
Endpoint de SLO (Single Logout)	URL que permite encerrar sessões de autenticação em várias aplicações de forma simultânea quando um usuário faz logout de uma delas.

OBS.: Parâmetros seguem os mesmos padrões existentes no IdP atual (OAM), conforme exemplo abaixo:



Attribute Name Mapping

Actions View Create Edit Delete Detach

Row	Message Attribute Name	Value	Always Send
1	User.IsActive	1	true
2	User.LoginClaro__c	$user.attr.loginclaro	true
3	User.LastName	$user.attr.sn	true
4	User.UserName	${user.attr.samaccountname}@claro.com.br	true
5	User.Email__c	$user.attr.mail	true
6	User.ProfileId	$user.attr.extensionattribute6	true
7	User.CompanyName	$user.attr.company	true
8	User.Phone	$user.attr.telephonenumber	true
9	User.FirstName	$user.attr.givenname	true
10	User.FederationIdentifier	$user.attr.samaccountname	true
11	User.CPF__c	$user.attr.extensionAttribute3	true
12	User.Embratel__c	$user.attr.loginembratel	true
13	User.NetServicos__c	$user.attr.loginnet	true

8.6 Informações necessárias da aplicação (Service Provider)

Tabela 4 - Informações que deverão ser fornecidas pelo SP com metadata.

Dado	Descrição
Metadata (XML ou URL)	Documento XML que contém informações sobre as entidades envolvidas na integração, incluindo Entity ID, certificados, URL de endpoints, atributos, ACS e SLO.

Caso o metadata da aplicação não esteja disponível, as informações podem ser disponibilizadas de forma individual, detalhando os campos essenciais para a integração:

Tabela 5 - Informações que deverão ser fornecidas pelo SP caso não exista metadata.

Dado	Descrição
Entity ID	Um identificador único que identifica exclusivamente o IdP ou SP.
Endpoints	URLs que indicam os pontos de extremidade específicos para a comunicação SAML.

URL ACS	Endpoint/URL do Assertion Consumer Service, onde o IDP devolverá o token com os atributos de autenticação/autorização do usuário.
Atributos	Detalhes sobre os atributos que podem ser compartilhados entre o IdP e o SP durante a autenticação, como nome, e-mail, função, etc.
Certificados	Os certificados digitais usados para assinatura e/ou criptografia das mensagens SAML

Exemplo de XML de metadata SAML

```
<?xml version="1.0" encoding="UTF-8"?>
<EntityDescriptor xmlns="urn:oasis:names:tc:SAML:2.0:metadata" entityID="https://sp.example.com/metadata">
  <SPSSODescriptor AuthnRequestsSigned="true" WantAssertionsSigned="true">
    <NameIDFormat>urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified</NameIDFormat>
    <AssertionConsumerService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
Location="https://sp.example.com/acs" />
    <SingleLogoutService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect"
Location="https://sp.example.com/slo" />
   <KeyDescriptor use="signing">
    <KeyInfo xmlns="http://www.w3.org/2000/09/xmldsig#">
     <X509Data>
      <X509Certificate>MIIC...CertificateData...==</X509Certificate>
     </X509Data>
    </KeyInfo>
   </KeyDescriptor>
  </SPSSODescriptor>
</EntityDescriptor>
```

9 INTEGRAÇÃO OAUTH2:

A integração entre o PingFederate e aplicações utilizando protocolo Oauth2 envolve a utilização de recursos e padrões comuns para permitir a autenticação e autorização seguras dos usuários na API.

As chamadas descritas nesta seção seguem uma jornada de autenticação OpenID.

9.1 (Step 1) Get FlowID

O PingFederate associa um flowID para cada transação SSO que utiliza a sua API de autenticação. Esse flowID é utilizado pelo PingFederate para determinar o estado atual da transação. Logo, para solicitar um flowID e dar início ao processo de autenticação, deve-se realizar um GET para o recurso /as/authorization.oauth2.

Na Tabela 6, apresenta-se um exemplo de uma chamada Get FlowID realizada para o PingFederate.

Tabela 6 - Exemplo de chamada Get FlowID

Chamada	
GET https://{{pfhost}}/as/authorization.oauth2?client_id={{client_id}}&response_type=code&response_mode=pi.flow&scope=openid&nonce=abc123	
Parâmetros	
client_id	{{client_id}}
response_type	code
response_mode	pi.flow
Scope	Openid
Nonce	abc123
Authorization	
`Este request não possui Authorization`	
Header	
`Este request não possui Header específicos`	
Body (Exemplo)	
`Este request não possui body`	

A resposta desta chamada deverá ser um Status OK 200 e irá conter um FlowID (`"id"`) e o atual estado da transação (`"status"`).

Na Tabela 7, um exemplo de resposta é apresentado.

Tabela 7 - Exemplo de resposta Get FlowID

Resposta (Exemplo)	Status 200 OK


```
{
	"id": "cbEQ0sRzx7",
	"pluginTypeId": "7RmQNDWaOnBoudTufx2sEw",
	"status": "USERNAME_PASSWORD_REQUIRED",
	"showRememberMyUsername": false,
	"showThisIsMyDevice": false,
	"thisIsMyDeviceSelected": false,
	"showCaptcha": false,
	"rememberMyUsernameSelected": false,
	"_links": {
		"self": {
			"href": "https://{{pfhost}}/pf-ws/authn/flows/cbEQ0sRzx7"
		},
		"checkUsernamePassword": {
			"href":
https://dev-auth-pf.clarobrasil.mobi/pf-ws/authn/flows/cbEQ0sRzx7
		}
	}
}
```


9.2 (Step 2) Get code

Com um flowID estabelecido, para se obter um authorization code, deve-se logar no PingFederate através da sua API de autenticação.

Sendo assim, deve-se realizar um POST no recurso /pf-ws/authn/flows/{{flowID}} (FlowID recebido no Step 1) e enviar um username e password no body no formato de um json. Isso faz com que o PingFederate conclua o processo de autenticação e envie o authorization code.

Na Tabela 8, é apresentado um exemplo desta chamada para o PingFederate.

Tabela 8 - Exemplo de chamada Get code

Chamada

POST https://{{pfhost}}/pf-ws/authn/flows/{{flowId}}

Parâmetros
Este request não possui Parâmetros

Authorization
Este request não possui Authorization

Header	
Content-Type	application/vnd.pingidentity.checkUsernamePassword+json
X-XSRF-Header	PingFederate

Body (Exemplo)

```
{
	"username":"34037168057",
	"password":"secret"
	}
```

A resposta desta chamada deverá ser um Status OK 200 e irá conter o atual estado da transação ("status") com o valor "COMPLETED" e um *authorization code* ("code").

Na Tabela 9, um exemplo de resposta é apresentado.

Tabela 9 - Exemplo de resposta Get code

Resposta (Exemplo) Status 200 OK


```
{
	"id": "RWcqQ2PAHx",
	"pluginTypeId": "7RmQNDWaOnBoudTufx2sEw",
	"status": "COMPLETED",
	"authorizeResponse": {
		"code": "z52R_UrhJJTfWvtkEpL_jC9vBVTbTazAaKNBmQXY"
	},
	"user": {
```

```
            "id": "34037168057",
            "username": "34037168057"
      },
      "_links": {
            "self": {
                  "href":"https://{{pfhost}}/pf-
                  ws/authn/flows/RWcqQ2PAHx"
            }
      }
}
```

9.3 (Step 3) Get access_token

Com um *authorization code* recebido no *Step* 2 ({{code}}), para se obter um *access token*, deve-se realizar um **POST** no recurso /as/token.oauth2 e enviar no body diversas *keys,* entre elas, as *keys grant_type=authorization_code* e *code*={{code}}. Na

Tabela 10, um exemplo desta chamada para o PingFederate.

Tabela 10 - Exemplo de chamada Get access_token

Chamada
POST https://{{pahost}}/as/token.oauth2
Parâmetros
Este request não possui Parâmetros
Authorization
Este request não possui Authorization
Header
Este request não possui Header específicos
Body (Exemplo)

client_id	{{client_id}}
grant_type	authorization_code
Code	{{code}}
redirect_uri	{{redirect_uri}}
client_secret	{{client_secret}}

A resposta desta chamada deverá ser um Status OK 200 e irá conter um *access token* ("access_token"), um *refresh token* ("refresh_token"), um *id token* ("id_token"), o tipo de *token* ("token_type") e o tempo de expiração ("expires_in") desses *tokens*.

Na Tabela 11, um exemplo de resposta é apresentado.

Tabela 11 - Exemplo de resposta Get access_token

Resposta (Exemplo)	Status 200 OK

```
{       "access_token": "1f2fS6PCgd2cxnY1wm71pMtzz1J2Hm48",
       "refresh_token": "bHFkV2UeZywVky5wM9ESIpi8iaUj8yxlybsHXMzfrG",
       "id_token": "eyJhbGciOiJS…7Krw",
       "token_type": "Bearer",
       "expires_in": 7199
}
```

9.4 (Step 4) Refresh

Com um *refresh token* recebido no *Step* 3 ({{refresh_token}}), para se obter um novo *access token*, deve-se realizar um **POST** no recurso /as/token.oauth2 e enviar no *body* diversas *keys*, entre elas, o *grant_type=refresh_token* e o *refresh_token*={{refresh_token}}.

Na Tabela 12, um exemplo desta chamada para o PingFederate.

Tabela 12 - Exemplo de chamada Refresh

Chamada	
POST https://{{pfhost}}/as/token.oauth2	
Parâmetros	
Este request não possui Parâmetros	
Authorization	
Este request não possui Authorization	
Header	
Este request não possui Header específicos	
Body (Exemplo)	
client_id	{{client_id}}
response_type	refresh_token
refresh_token	{{refresh_token}}
client_secret	{{client_secret}}
redirect_uri	{{redirect_uri}}

A resposta desta chamada deverá ser um Status OK 200 e irá conter um novo access token ("access_token").

Na Tabela 13, um exemplo de resposta é apresentado.

Tabela 13 - Exemplo de resposta Refresh

Resposta (Exemplo)	Status 200 OK

```
{       "access_token": "1f2fS6PCgd2cxnY1wm71pMtzz1J2Hm48",,
        "token_type": "Bearer",
        "expires_in": 7199
}
```

9.5 (Step 5) User Info

Com um *access token* recebido no *Step* 3 ou 4 ({{access_token}}), para se obter informações sobre o usuário, deve-se realizar um **GET** no recurso /idp/userinfo.openid e enviar o *token* através de um *Authorization Bearer Token* {{access_token}}.

Na Tabela 14, um exemplo desta chamada para o PingFederate.

Tabela 14 - Exemplo de chamada User Info

Chamada	
GET https://{{pfhost}}/idp/userinfo.openid	
Parâmetros	
Este request não possui Parâmetros	
Authorization	
Bearer Token	{{access_token}}
Header	
Este request não possui Header específicos	
Body (Exemplo)	
Este request não possui body	

A resposta desta chamada deverá ser um Status OK 200 e irá conter as informações do usuário.

Na Tabela 15, um exemplo de resposta é apresentado.

Tabela 15 - Exemplo de resposta User Info

Resposta (Exemplo)	Status 200 OK

```
{
      "sub": "34037168057"
}
```

9.6 (Step 6) Introspect

Para realizar uma verificação de um determinado token (por exemplo, um *access token*), deve-se realizar um **POST** no recurso /as/introspect.oauth2 e enviar no *body* a *key* t*oken*={{access_token}}, juntamente com um *client_id* e *client_secret*.

Tabela 16, um exemplo desta chamada para o PingFederate.

Tabela 16 - Exemplo de chamada Introspect

Chamada
POST https://{{pfhost}}/as/introspect.oauth2
Parâmetros
Este request não possui Parâmetros
Authorization
Este request não possui Authorization
Header
Este request não possui Header específicos
Body (Exemplo)

client_id	{{client_id}}
Token	{{access_token}}
client_secret	{{client_secret}}

A resposta desta chamada deverá ser um Status OK 200 e irá conter algumas informações do usuário e o campo `"active"` no valor **`true`**. Na Tabela 17, um exemplo de resposta é apresentado.

Tabela 17 - Exemplo de resposta Introspect

Resposta (Exemplo)	Status 200 OK


```
{
        "sub": "34037168057",
        "scope": "openid",
        "pi.sri": "w2242KpzlUvfAIdK4NuaUOd0e1Y..VIjM",
        "active": true,
        "token_type": "Bearer",
        "exp": 1666491634,
        "client_id": "troubleshooting_pi"
}
```


9.7 (Step 7) Revogação

Por fim, para realizar uma revocação e tornar um determinado *token* inválido, deve-se realizar um **POST** no recurso /as/revoke_token.oauth2 e enviar no *body* a *key* t*oken*={{access_token}}, juntamente com um *client_id* e *client_secret*. O campo *token_hint* pode ser utilizado para informar o tipo de *token* que deseja revogar. Embora opcional, esse campo possui a função de acelerar o processo de revogação.

Na Tabela 18, um exemplo desta chamada para o PingFederate.

Tabela 18 - Exemplo de chamada Revoke

Chamada
POST https://{{pfhost}}/as/revoke_token.oauth2

Parâmetros	
Este request não possui Parâmetros	
Authorization	
Este request não possui Authorization	
Header	
Este request não possui Header específicos	
Body (Exemplo)	
client_id	{{client_id}}
token_hint	refresh_token
Token	{{access_token}}
client_secret	{{client_secret}}

A resposta desta chamada deverá ser apenas um Status OK 200.

10 REFERÊNCIAS:

Guia de Referências: https://docs.pingidentity.com/r/en-us/pingfederate-112/help_dependencyerrormanagementtasklet_dependencyerrormanagementstate

SAMLv2:https://www.pingidentity.com/en/resources/identityfundamentals/authentication-authorization-standards/saml.html

OIDC (OpenID Connect): https://www.pingidentity.com/en/resources/identity-fundamentals/authentication-authorization-standards/openid-connect.html

REST API: https://apidocs.pingidentity.com/pingdirectory/directory/v1/api/guide/