Trabalho Conclusão Curso

André Azevedo Vargas

DESENVOLVIMENTO E USO DE UM FORMATO DE POLÍTICAS DE PRIVACIDADE NO CONTROLE DE ACESSO.

Florianópolis

2017

Universidade Federal de Santa Catarina
Departamento de Informática Estatística
Curso de Ciências da Computação

André Azevedo Vargas

DESENVOLVIMENTO E USO DE UM FORMATO DE POLÍTICAS DE PRIVACIDADE NO CONTROLE DE ACESSO.

Trabalho de Conclusão de Curso apresentado à disciplina do Curso de Ciências da Computação da Universidade Federal de Santa Catarina para a obtenção do título de bacharel.
Orientador: Prof. Dr. Carla MerkleWestphall
Co-orientador: Jorge Werner

Florianópolis

2017

André Azevedo Vargas

DESENVOLVIMENTO E USO DE UM FORMATO DE POLÍTICAS DE PRIVACIDADE NO CONTROLE DE ACESSO.

Este trabalho de conclusão de curso julgado aprovado para a obtenção do Título de "Bacharelado em Ciências da Computação", e aprovado em sua forma final pelo Programa de Graduação em Ciências da Computação.

Florianópolis, 01 de Junho 2017.

_____

Profa. Dra. Carla Merkle Westphall

Coordenadora

Banca Examinadora:

_____

Doutorando Jorge Werner

Coorientador

_____

Prof. Dr. Carlos Becker Westphall

_____

Mestrando Gerson Luiz Camillo

Este trabalho é dedicado aos meus colegas, professores e aos meus queridos pais.

AGRADECIMENTOS

Gostaria de agradecer a Deus por manter meu equilíbrio apesar das dificuldades que passei nos últimos anos, aos meus familiares, minha mãe Maria Teresa, minha irmã Bárbara, Luiz Mário Machado e sobretudo meu pai Décio Gabriel pelo apoio nos momentos mais conturbados do trabalho e minha avó Solange pelo carinho ao longo do trabalho.

Gostaria de agradecer alguns colegas em particular, sendo esses Marcos Schead, Luca Campelli pela contribuição em parte das ferramentas usadas.

Por fim agradeço aos caros colegas do LRG, em especial a Professora Carla e o Jorge pela motivação, orientações e paciência e o professor Carlos pela paciência e contribuições e por fim ao Gerson que incansavelmente me deu o suporte em grande parte do desenvolvimento do trabalho.

RESUMO

Muitas aplicações na web que oferecem serviços á pessoas ou empresas na Internet precisam de dados para identificar estas entidades ou realizar serviços de forma automatizada e mais prática. Estas aplicações tem o dever proteger o acesso a esses dados e o responsável por isso são sistemas de controle de acesso. Muitas vezes disponibilizamos dados em ambientes em nuvem e acabamos não sabendo quais dados são coletados, como são utilizados ou quem fará uso deles. A privacidade tem sido um desafio para a segurança da informação e já vem sendo discutida e avaliada antes mesmo do advento da Internet, que é um dos ambientes onde mais ocorrem ataques a ela. Portanto a ideia deste trabalho será desenvolver e utilizar políticas de privacidade através das linguagens de controle de acesso e utilizando dados de preferencias definidas pelo dono dessas informações.

Palavras chaves: Privacidade, políticas de privacidade, controle de acesso.

ABSTRACT

Many web applications that offer services to people or businesses on the Internet need data to identify these entities or perform services in an automated and more practical way. These applications have the duty to protect the access to this data and the responsible for this are access control systems. We often make data available in cloud environments and we end up not knowing what data is collected, how it is used or who will make use of it. Privacy has been a challenge for information security and has already been discussed and evaluated before the advent of the Internet, which is one of the most attacking environments. Therefore the idea of this work will be to develop and use privacy policies through access control languages and using preference data defined by the owner of this information.

Keywords: Privacy, privacy policies, identity management systems, access control.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS

PII – Personal Information Identify.

P3P – Platform for Privacy Preferences Project.

W3C – World Wide Web Consortium.

IdM – Identity Manager.

IdP- Identity Provider.

SP – Service Provider.

AC – Access Control

PEP – Policy Enforcement Point

PDP – Policy Decision Point

PAP – Policy administrator Point

PIP – Policy Information Point

XML – eXtensible Markup Language

XACML – eXtensible Actcess Control Markup Language

JSON – JavaScript Object Notation

RBAC – Role-Based Access Control

ABAC – Attribute-Based Access Control

DAC – Discritionary Access Control

MAC – Mandatory Access Control

PI – Personal Information

PCP – Personal Caracteristics and Preferences

AH – Activities and Habbits

RS – Relationships

LO – Location

SI – Service Improvement

SC – Scientific

CO – Comercial

PP – Personal Party

TP – Third Party

JWT – JSON Web Tokens

SSO – Single Sign-On

EPAL – Enterprise Privacy Authorization Language

P3P – Platform for Privacy Preferences

SQL – Structured Query Language

OIDC – OpenID Connect

SUMÁRIO

1 INTRODUÇÃO

Nos últimos anos houve um grande crescimento no número de utilizadores da Internet assim como na quantidade de informação disponível, afirma MACEDO (2013). Isso se dá graças as facilidades que a Internet proporciona e aos serviços que pode-se acessar nela e portanto ISHITANI (2013) acredita que a privacidade na web é uma questão que tem levantado, atualmente, várias discussões.

A computação em nuvem está modificando a forma de usar a Internet já que os serviços e recursos podem ser utilizados sob demanda, de acordo com a necessidade dos usuários. A computação em nuvem pode oferecer poder de processamento, oferecer serviços de software e de armazenamento ou mesmo uma infraestrutura de dados distribuídos.

Apesar de todas as vantagens da computação em nuvem e do grande número de empresas que oferecem serviços, os usuários relutam em armazenar seus dados na nuvem, pois têm dúvidas quanto à segurança dos ambientes. A privacidade, segundo LANGHEINRICH (2002), já era uma preocupação no século 19, devido a problemas relacionados à imagem, pois houve o surgimento da fotografia.

Para empresas pode existir uma potencial proliferação de identidades e credenciais necessárias para acessar os serviços. Existe necessidade de limitar corretamente os domínios e os fluxos de informações entre os provedores de serviços nas nuvens, já que podem existir fluxos diversos entre vários domínios para que um serviço seja fornecido.

À medida que cresce o número de consumidores que participam de atividades on-line, torna-se cada vez mais imperativo que as organizações expressem suas práticas de privacidade de forma precisa, acessível e útil. Isso permite que os consumidores avaliem as práticas de uso e manipulação de informações de uma organização antes de determinar se desejam ou não se envolverem em transações com a organização (STUFFLEBEAM et al., s.d).

STALLINGS (1999) comenta que a privacidade em relação aos dados disponibilizados na nuvem, pode ser vista como uma questão de controle de acesso, em que é assegurado que os dados armazenados estarão acessíveis apenas para pessoas, máquinas e processos autorizados.

Muitos usuários, segundo ISHITANI (2003) não sabem como uma invasão de privacidade pode ocorrer ou o que se deve fazer para proteger sua privacidade.

A proposta deste trabalho visa criar politicas com regras de controle de acesso visando conservar a privacidade dos dados de um determinado usuário. Estas regras terão como alvo

os atributos e dados disponibilizados por um *token* de atributos de privacidade, que terá a tarefa de selecionar as preferências de uso de dados que já foram devidamente agrupados. Assim só pode-se ter acesso ao recurso com os devidos atributos se ás preferências dadas por este *token* forem respeitadas. Ao fim, existirá um sistema que simulará um sistema que gerencia o controle de acesso, mais especificamente autorização, para testar essas políticas.

1.1 OBJETIVOS

O objetivo desse trabalho é desenvolver e usar politicas de privacidade utilizando a linguagem para controle de acesso XACML que possibilite o usuário criar regras bem definidas de politica de privacidade, a fim de utilizá-lo em um simulador de um sistema de gerenciamento de identidade.

1.1.1 Objetivos Específicos

Realizar análise bibliográfica sobre políticas de privacidade e do funcionamento de um sistema de gestão de identidade baseado no modelo de IdM do OpenID Connect.

Realizar estudos para analisar melhores e possíveis maneiras de utilizar atributos e propriedades para a definição das regras que serão disponibilizadas para o usuário do sistema criar a política, tentando abranger ao máximo de regras possíveis para sistemas de computação em nuvem e compatíveis com formatos utilizados por sistemas de gerenciamento de identidades.

Criar politicas de privacidade através de uma linguagem de politica de controle de acesso XACML para simular em um sistema de controle de acesso baseado em ABAC.

Pesquisar e estudar sobre diferentes bibliotecas em Java que realize a simulação dos pontos de acesso de controle de uma politica de controle de acesso.

Construir uma aplicação *Web* que gere um *token* que contenha as preferências de privacidade do usuário a serem usadas para tratar a privacidade no acesso aos dados de um usuário com uma identidade digital.

Construir um sistema simulador de controle de acesso para simular políticas de privacidades criadas previamente e utilizando o *token* de privacidade.

1.2 MOTIVAÇÃO E JUSTIFICATIVA

Grande parte dos sistemas de computação em nuvem e sistemas *Web* muitas vezes coleta e utiliza informações pertencentes a usuários para realizar seus serviços. Esses dados muitas vezes são informações que podem causar algum tipo de constrangimento quando compartilhado ou usado de forma indevida. Neste caso, como seria a forma mais apropriada de lidar com estas informações para evitar possíveis problemas relacionados ao uso desses dados que podem vir a ferir a privacidade deste indivíduo?

A computação em nuvem vem sendo bastante usada por usuários através de aplicações que prestam serviços na Internet. Para lidar com uma grande quantidade de dados como é o caso de ambientes em nuvem, a necessidade da utilização de sistemas de gerenciamento de identidade (IdM) para administrar e gerenciar identidades desses sistemas é cada vez maior. No entanto, apesar dos benefícios e facilidades providos de serviços da nuvem, WERNER e WESTPHALL (2016) afirmam que problemas de segurança e privacidade surgem na integração de nuvem e IdM.

As políticas de controle de acesso definem quais sujeitos podem acessar quais objetos e em qual modo, isto é, restringe o que o usuário pode fazer ao usar uma aplicação ou serviço. Essas políticas melhoram a segurança dos ambientes de nuvem regulando os fluxos de informação. Porem para manter a privacidade de usuários de um sistema *Web* em um sistema de gerenciamento de identidades é necessário que os dados pessoais nos provedores de identidade sejam disponibilizados, acessados ou utilizados com o consentimento e a ciência de quais dados, quais finalidades e quem usará esses dados, dando ou não permissão de acesso de acordo com os privilégios e preferências deste usuário.

A utilização de dados de usuários para fins secundários segundo IYILADE e VASSILEVA (2014**),** representam riscos para privacidade de forma significativa para o usuário. Com o aumento de dispositivos que fazem uso da internet, houve uma necessidade de proteger esses dados coletados e armazenados.

O aumento da utilização da Internet para atividades diárias está trazendo novas ameaças à privacidade pessoal (GOLDBERG, 1997). Isso vem sendo uma grande preocupação para usuários que disponibilizam suas informações, muitas destas confidenciais, para determinados sistemas pertencentes a entidades que de alguma maneira obtém e utilizam esses dados para prestar algum determinado serviço.

ISHITANI (2013) alega que muitas pessoas ainda evitam usufruir plenamente de seus serviços, por recearem ter sua privacidade invadida**.** Esse receio citada por ISHITANI (2013) é compreensível, já que nem sempre é possível configurar as preferências de privacidade de tal forma que se cria um desconforto e insegurança por parte de usuários e enviar dados para um ambiente em nuvem.

A necessidade de possuir a privacidade é que fez com que políticas de privacidade fossem necessárias para estas entidades, a fim de garantir ao usuário segurança e confiança na entidade que provêm serviços e declarando os fins que terão os dados disponibilizados.

Sánchez indicado por WEINGÄRTNER (2014), ressalta que assim que os dados dos usuários são enviados a provedores de identidades, o controle sobre como esses dados são divulgados, armazenados e utilizados é perdido (LANGHEINRICH, 2002). Portanto, é desejável que dados sejam controlados pelo usuário dono destes sugere WERNER e WESTPHALL (2016) .

Para a grande maioria dos sistemas em nuvem ou sistemas *Web* que lidam com diferentes entidades ou usuários, é necessário coletar dados destes, a fim de prover algum serviço e além de tudo diferenciar um usuário de outro. Para tal, é sabido que o ideal seria gerar políticas de privacidade com regras bem definidas e com transparecia para o usuário.

ISHITANI (2013) comenta que houve tentativas de se buscar uma solução para o problema, através da disponibilização de ferramentas para proteção de privacidade de usuário, no entanto não se obteve êxito de maneira satisfatória.

Foram encontradas poucas referências que mostram como definir essas regras para essas políticas e para cada formato de acordo com o sistema de gestão de identidade escolhido, e uma grande contribuição seria se houvesse uma ferramenta capaz de definir essas regras de forma simples, sem precisar abrir um documento XML ou JSON e conhecer a estrutura ou linguagem para poder montar essas políticas.

1.3 METODOLOGIA DE PESQUISA

Primeiramente será feito uma coleta de materiais para começar a fazer um estudo sobre temas relacionados de grande importância como privacidade, gerenciamento de identidades e controle de acesso. Com isto, será possível escrever algum documento mais direcionado aos objetivos da proposta do trabalho e o conhecimento estará estruturado de forma suficiente para realizar o desenvolvimento prático.

Para o desenvolvimento ser iniciado é necessária a instalação e configurações das ferramentas de implementação. Começando por uma versão mais atual do Eclipse com o Java web. O sistema fará uso do Framework Spring boot para o desenvolvimento de novas ferramentas que serão utilizadas. Será necessário estudar uma forma de simulação do sistema de controle de acesso e uma maneira de controlar a privacidade do usuário

Após a instalação, será estudado o JSON, que é um formato leve de troca de dados entre sistemas e é o formato de troca de mensagens utilizado pelo OpenID Connect, que apesar de não usado diretamente neste trabalho é o IdM no qual será baseada estrutura de controle de acesso. O JSON poderá ser utilizado dentro do desenvolvimento de alguma ferramenta de requisição.

Realizar um estudo e a construção de políticas de privacidade utilizando a linguagem XACML que é uma das linguagens usadas para controle de acesso.

Depois da estrutura da proposta completa e o desenvolvimento e configurações estiverem de acordo, serão realizados os testes e casos de uso para chegarmos a conclusão do trabalho proposto.

1.4 ESTRUTURA DO TRABALHO

Este trabalho será dividido em cinco capítulos:

- O Capítulo 2 é o capítulo que contem toda a fundamentação teórica, com os quatro tópicos envolvidos neste trabalho, sendo estes temas a Segurança da informação, Privacidade, Gerenciamento de Identidades , Controle de acesso e trabalhos relacionados.
- O Capítulo 3 começa o desenvolvimento. Um tópico sobre a proposta, outro sobre as ferramentas utilizadas para sua concepção e por fim como foram instaladas e configuradas as ferramentas e como foi construído o modelo proposto.
- O Capítulo 4 possui os resultados e testes gerados através dos casos de uso realizados a partir do desenvolvimento realizado.
- O Capítulo 5 é a conclusão do trabalho como um todo, possuindo tópicos como trabalhos futuros e agradecimentos.

Por fim temos as referências utilizadas ao longo do trabalho e anexos com partes de códigos e configurações desenvolvidas.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo veremos alguns conceitos que serão abordados na proposta deste trabalho e, portanto precisamos conhecer alguns aspectos para compreender melhor o que será realizado no desenvolvimento da proposta. A fundamentação é dividida em 5 seções. O primeiro será segurança da informação, depois sobre privacidade, seguido de gerenciamento de identidades e, por fim, controle de acesso.

2.1 SEGURANÇA DA INFORMAÇÃO

A informação é o dado com uma interpretação lógica ou natural dada a ele por seu usuário na percepção de REZENDE e ABREU indicado por LAUREANO (2005)**.** Devido a sua natureza contributiva, possui grande valor e oferecer vantagens a entidades que a possui e podem-se ajudar muitas entidades proprietárias dos dados, sendo facilitadores de utilização de serviços em um ambiente computacional por exemplo.

Em contrapartida os sistemas de segurança da informação existente em ambientes computacionais, devem também ter em conta as ameaças que hoje se colocam às liberdades individuais, à proteção dos dados pessoais e consequentemente à privacidade, objeto de estudo deste trabalho. Dependendo da forma de uso dessas informações, o mau uso delas pode gerar bastantes problemas a quem pertence essas informações assim como facilidades.

A segurança da informação é a proteção dos ativos informacionais de uma organização, em relação às perdas, exposição indevida ou dano (BRAGANÇA et. al., 2010). É um conceito que se aplica a toda informação armazenada, manipulada ou transmitida em uma ou entre organizações. A norma ISO/IEC 17799:2005 (2005) ressalta que a segurança da informação é a preservação da confidencialidade, da integridade e da disponibilidade da informação; adicionalmente, outras propriedades, tais como autenticidade, responsabilidade, não repúdio e confiabilidade, podem também estar envolvidas.

SÊMOLA indicada por LAUREANO (2005) afirma que segurança da Informação também é definida como uma área do conhecimento dedicada à proteção de ativos da informação contra acessos não autorizados, alterações indevidas ou sua indisponibilidade**.**

2.1.1 Princípios de Segurança

Para o padrão ISO/IEC 17799:2005 (2005), sistemas que prezam a garantia da segurança da informação precisa seguir alguns princípios, conhecidos também em algumas bibliografias como tríade CIA (*Confidentiality*, *Integrity* e *Availability*).

A confidencialidade é a limitação do acesso a um recurso por quem realmente possui autorização, a fim de impedir que pessoas não autorizadas tenham acesso. A integridade é a propriedade que garante que a informação recebida seja legítima do envio até a recepção e armazenamento, ou seja, uma proteção dos dados contra modificações intencionais ou acidentais de uma parte. A disponibilidade por sua vez é a propriedade que garante que a informação deve estar disponível no momento em que a mesma for necessária para o uso do usuário permitido.

Evidentemente existem outros princípios que devem ser respeitados e não são menos importantes.

- **Autenticidade:** garante que a informação ou o usuário da mesma é autêntico.
- **Não repúdio:** a impossibilidade de negar o envio ou recepção de uma informação ou dado.
- **Privacidade:** segundo RAULINO (2013)**,** difere um pouco do conceito de confidencialidade, porém facilmente confundível, apesar de uma informação poder ser considerada confidencial, mas não necessariamente privada. Um dado privado deve ser gerido somente pelo dono. Garante ainda que a informação não será disponibilizada para outras pessoas. É a capacidade de um usuário realizar ações em um sistema sem que seja identificado;

2.1.2 Políticas de Segurança

O tipo de controle de seguranças estudados neste trabalho é de controles lógicos, dispensando qualquer contato com o controle físico. O mecanismo utilizado nesse trabalho é o controle de acesso utilizando de políticas de segurança que será enfatizado no capítulo 2.4.

Uma política é pode ser definida como um conjunto de normas ou regras que é determinada por uma instituição ou entidade. As políticas de segurança, por sua vez intenções e diretrizes globais formalmente expressas por uma descrição que orienta o que e como devem ser feitos uma política, conhecidas como diretrizes (ISO/IEC 17799, 2005). Estas diretrizes estabelecidas determinam regras que devem ser seguidas pela instituição para que seus recursos e informações sejam garantidos.

2.2 PRIVACIDADE

Como já mencionado, a privacidade é alvo de grande preocupação em computação em nuvem e sistemas *Web*. Essa preocupação existe, pois grande parte dos sistemas na web necessita utilizar dados de seus usuários para identificar e prestar serviços de forma adequada e aceitável e precisam seguir uma legislação aplicada por seu país para privacidade. Os usuários também têm preocupações tanto quanto empresas que necessitam coletar dados para prestar serviços, já que nem sempre sabem para qual fim, quem utiliza e quais são os dados utilizados.

A falta de consciência e muitas vezes de consentimento gera grande polêmica e dessa forma o estudo da privacidade e noção de maneiras de protegê-la é fundamental para aplicações web ou em nuvem, visto a facilidade do acesso à informação que existe graça ao advento da Internet.

Portanto neste capítulo serão abordados os diferentes pontos de vista que definem a privacidade de modo geral, conceitos básicos e princípios segundo a norma ISO/ IEC 21900 (2011) e referências, as definições de políticas de privacidade, direcionando para o contexto de sistemas *Web*, e também temas que vem sendo estudados e discutidos pela comunidade científica.

2.2.1 Definições e Conceitos

Existem muitas fontes que definem de sua própria maneira o conceito de privacidade, pois esta pode ser observada de diversos pontos de vista e contextualizada em diferentes áreas.

A privacidade segundo Alan Westin (1967) citado por (EDUCAUSE, 2013) é o controle dos dados pessoais que os usuários possuem. Portanto a tomada de decisão e ter consciência de como são manipuladas as informações devem ser do usuário proprietário ou de quem tem sua autorização.

No ponto de vista direcionado ao indivíduo, a privacidade é segundo LANDWEHR (2012) como o controle da liberação de dados pessoais que os usuários possuem. Portanto a tomada de decisão e ter consciência de como são manipuladas as informações devem ser do usuário proprietário e de quem tem sua autorização.

Privacidade da Informação é a proteção da coleta, armazenamento processamento de divulgação dos dados sensíveis e de PII (identificação de informações pessoal) para terceiros sem o consentimento do mesmo (SANTOS, 2011).

KALEMPA (2009) declara privacidade na visão do direito, a privacidade é um direito de qualquer pessoa, e muitas nações têm em suas constituições leis que garantem ao cidadão o direito de possuí-la.

Os dados armazenados em sistemas colocados na internet ou em nuvens no ponto de vista de privacidade podem ter uma série de definições. Algumas dessas definições fazem parte do conceito e termos utilizados quando se trata de privacidade em relação aos tipos possíveis de dados. Alguns dos termos doravante são definidos a norma ISO 29100 (2011), as informações podem ser definidas nos seguintes tipos:

- **Informações de identificação pessoal (PII):** qualquer informação que pode ser usada para identificar uma entidade, podendo ser diretamente ou indiretamente ligada a entidade.
- **Preferências de privacidade:** São dados que o usuário define para usar em conjunto a uma política a fim de proteger os dados sensíveis escolhidos pelo usuário.
- **Controlador de dados**: parte que, de acordo com a legislação, é competente para decidir sobre o conteúdo e o uso de dados pessoais, independentemente de esses dados serem coletados, armazenados, processados ou divulgados por essa parte ou por um agente.

Em sua pesquisa WEINGÄRTNER (2014) declara que existem legislações internacionais e nacionais que visam proteger a privacidade dos usuários no âmbito de sistemas da informação, e reforçado por KALEMPA(2009) que diz que existem diferentes formas de lidar com a privacidade no âmbito global.

Assuntos legais relacionados a utilização da nuvem envolvem a proteção da informação e sistemas computacionais e contramedidas para tratar as violações de segurança e privacidade por intrusão, vazamento, revelação ou divulgação de dados protegidos(MARCON et. al., 2010).

A privacidade é definida por leis, técnicas e por mecanismos, geralmente políticas. Em sistemas computacionais é necessário seguir alguns princípios usados como base e diretriz para a aplicação da privacidade.

2.2.2 Princípios da Privacidade

Existem muitas vertentes que declaram de sua maneira os princípios de privacidade, pois seus estudos levam em consideração estudos realizados anteriormente das leis desenvolvidas por vários países e organizações internacionais. Com o crescente desenvolvimento de sistemas computacionais existente, esses princípios acabaram por ser definidos como um guia. Os princípios segundo OECD (2013) são os seguintes:

1. **Princípio de abertura**: Deve haver uma política geral de abertura sobre desenvolvimentos, práticas e políticas em relação aos dados pessoais. Os meios devem estar prontamente disponíveis para estabelecer a existência e a natureza dos dados pessoais, e os principais fins de sua utilização, bem como a identidade e residência habitual do controlador dos dados.
2. **Princípio da qualidade dos dados:** Os dados pessoais devem ser relevantes para os fins para os quais estão sendo usados, e, na medida necessária para esses fins, deve ser preciso, completo e mantido atualizado.
3. **Princípio de participação individual:** Os indivíduos devem ter o direito de obter de um controlador dos dados, ou de outra forma, a confirmação de se o controlador de dados tem ou não os dados relacionados e ter comunicado a eles de forma clara. Dar razões se um pedido feito nos termos for negado e o poder de contestar sobre os dados relacionados a eles.
4. **Princípio da especificação do propósito:** Os fins para os quais os dados pessoais são coletados devem ser especificados até no máximo o momento da coleta de dados e o uso subsequente limitado ao cumprimento desses propósitos ou outros que não sejam incompatíveis com esses propósitos e como são especificados em cada ocasião de mudança de propósito.
5. **Princípio da limitação de coleta:** Deve haver limites para a coleta de dados pessoais e quaisquer dados devem ser obtidos por meios legais justos e, quando apropriado, com o conhecimento ou o consentimento da pessoa em causa.
6. **Princípio de limitação de uso:** Os dados pessoais não devem ser divulgados, disponibilizados ou de outra forma a ser usado para fins diferentes dos especificados

no acordo com exceção do consentimento da pessoa em causa ou pela autoridade da lei.

7. **Princípio de salvaguardas de segurança:** Os dados pessoais devem ser protegidos por garantias de segurança razoáveis contra riscos como perda ou acesso não autorizado, destruição, uso, modificação ou divulgação de dados.
8. **Princípio da Responsabilidade:** Um controlador de dados deve ser responsável pelo cumprimento das medidas que dão efeito aos princípios acima mencionados.

Através destes princípios, especialistas que prezam a privacidade de seus usuários buscam se orientar para o desenvolvimento e gerenciamento de seus sistemas devido a vulnerabilidade que existe na Internet.

2.2.3 Preocupações e Desafios da Privacidade na Internet

A ISHITANI (2003) alega que a *Web* aumenta os riscos de invasão de privacidade, pois facilita a coleta, monitoramento e análise de informações sem que os usuários percebam que isso esteja ocorrendo. Muitos ataques conhecidos em torno do mundo ocorrem devido a esse tipo de vulnerabilidade, o que pode ser causa de preocupações e constrangimentos para as pessoas, pois esses dados poderão sempre estar em poder dos atacantes e ser usado de forma maliciosa em qualquer momento no futuro.

Com aumento do número de usuários na Internet, a necessidade de protegê-los de todos os riscos em relação as suas ações e a possíveis fraudes por uso indevido de dados era cada vez maior. PEASON e CHARLESWORTH (2009) afirma que novos usuários da Internet geralmente não percebem que todas as postagens que eles fazem para um grupo de notícias, todos os e-mails que enviam, todas as páginas *WorldWideWeb* acessadas, e todos os itens que eles compram on-line podem ser monitorados ou registrados por algum terceiro não visto. Esse tipo de situação torna usuários mais inseguros com relação confiança em aplicações que oferecem serviços, sobretudo os ambientes em nuvem.

É inegável que essas tecnologias e serviços são de grande necessidade e facilidade para as pessoas e como descreve CHADWICK e FATEMA (2012) a computação em nuvem oferece a oportunidade de armazenar uma enorme quantidade de dados de forma relativamente barata, dando a usuários acesso a serviços ou aplicativos independentemente da sua localização ou dispositivo de computação que eles usam e apesar de todos esses

benefícios, problemas de privacidade e segurança ainda são grandes desafios da computação em nuvem.

RAO e ROHATGI (2000) comentam que a ausência de proteção de privacidade na infraestrutura básica da *Web*, junto ao aumento da migração de atividades humanas de interações reais de palavras para interações baseadas na web, representa uma grave ameaça à privacidade das pessoas.

Com esse problema evidenciado deveria ser possível dar a opção ao usuário sobre o uso de dados e o que pode ou não ser coletado ou compartilhado de seus dados, porém nem todos os sistemas oferecem esse controle a usuários ou o deixam a par das ações realizadas.

2.2.4 Políticas de Privacidades

A especificação e a aplicação da política de privacidade tornam-se alvo e fonte de atividades de pesquisa ao longo dos últimos anos, já que o uso da Internet vem aumentando em todo o mundo (ISO/IEC27799, 2005).

Existem dois tipos de documentos que são definidos como políticas privacidade. Ambos os documentos tratam do mesmo objetivo, porém para fins e usos diferentes. MARCON et. al. (2010) define uma politica voltada para a ciência e compreensão dos usuários. Ele declara que as políticas de privacidade on-line são documentos disponibilizados por organizações através de seus sites que explicam como as informações pessoais dos consumidores serão coletadas, usadas e armazenadas pela organização. Essas políticas devem ser escritas de uma forma clara e bem definidas.

Entretanto, em sistemas computacionais, precisamos de um documento que realize o controle de privacidade e garanta essa proteção das informações pessoais. BERTINO e TAKAHASHI (2011) declaram que políticas de privacidade referem-se a políticas implementadas por uma organização sobre o uso de PII pela organização**.**

Pode ser definida como um conjunto de regras criado por uma entidade para determinar quais outras entidades ou serviços podem ter acesso as suas informações e que ações podem ser realizadas para este fim. O importante é que o usuário dono do dado esteja consciente dos dados usados, dos destinos e destinatários e finalidades de uso além do controle sobre essas preferências ao usuário.

Essas políticas são a maneira como o controle de acesso pode definir as proteções para os usuários. Em combinação com outras propriedades como a autenticação e uso de

criptografia em base de dados onde são armazenadas essas informações são fundamentais para fortalecer a proteção e manter os dados confidenciais.

Algumas linguagens criadas com o intuito de proteger o controle de acesso e privacidade pode-se ver a seguir:

EPAL: É uma linguagem formal para escrever políticas de privacidade empresarial que regem as práticas de tratamento de dados em sistemas de TI de acordo com direitos de autorizações positivas e negativas de grão fino. (IBM **,** 2003)

P3P: é uma linguagem que permite que sites expressem suas práticas de privacidade em um formato padrão que pode ser recuperado automaticamente e interpretado facilmente pelos agentes do usuário (W3C, 2006). O usuário P3P permite que usuários sejam informados das práticas do site de forma clara a fim de para automatizar a tomada de decisões com base nessas práticas quando apropriado.

A proposta é relacionada a outra linguagem, o XACML, uma linguagem de controle de acesso que será detalhada no capítulo 2.4.

2.3 GERENCIAMENTO DE IDENTIDADES

Com a crescente quantidade de aplicações desenvolvidas para a *Web*, houve uma necessidade de criação de uma maneira segura e inteligente de gerir as informações oriundas de usuários para prestação de serviços. A concepção do gerenciamento de identidades existiu, pois houve necessidade de organizar e gerenciar dados sobre muitos usuários, para que possam ter acesso a recursos de forma segura, prática e justa.

A tecnologia digital de gerenciamento de identidade é fundamental para personalizar e melhorar a experiência do usuário, protegendo a privacidade, sustentando a responsabilidade nas transações e interações, e em conformidade com os controles regulamentares (LEANDRO, 2012).

Porém um dos grandes desafios do gerenciamento de identidades é a segurança de dados dos usuários. Muitos desses sistemas não oferecem ao usuário controle dos dados que estão sendo usados e não deixam claro quais dados estão sendo utilizados.

Nesta seção será apresentada uma visão geral sobre de gerenciamento de identidades, a fim de esclarecer alguns conceitos importantes que fazem parte deste processo e entidades que constituem o mesmo, modelos e ferramentas de IdM (gerenciamento de identidades) .

O gerenciamento de identidades, para JOSANG et al. (s.d) refere-se ao processo de representação e reconhecimento de entidades como identidades digitais em redes de

computadores. Sua importância cresce conforme crescem os serviços que precisam utilizar autenticação e controle de acesso de usuários.

Este trabalho visa por proteger privacidade dos usuários em um sistema de gerenciamento de identidade. Sabendo disso, em BIRREL e SCHNEIDER descreve três características principais para IdM:

- **Indetectabilidade:** é a maneira de ocultar as transações realizadas pelo usuário, não sendo possível encontrar ações realizadas por ele em um determinado sistema;
- **Não vinculação:** as identidades do usuário ocultam a ligação do usuário com suas ações e transações;
- **Confidencialidade:** é o controle dado aos usuários do compartilhamento dos próprios atributos.

2.3.1 Conceitos Básicos

Gerenciamento de identidades possui algumas terminologias que necessitam ser apresentadas, alguns conceitos e termos que estão diretamente ligados a este tema e aos componentes que serão citados ao longo deste capítulo. Um elemento de grande importância são as entidades, que são representações de uma pessoa, organização, servidor ou até mesmo um conjunto destes os quais são reconhecidamente distintos entre si.

Em um sistema web precisa de uma forma de ser reconhecido e diferenciado por um IdM. Para isso ele precisa possuir algo que o torne único dentro de um sistema. A identidade digital é uma ou mais informações sobre uma entidade usada por sistemas de computadores que o representam em um determinado domínio. A ISO/IEC 24760-1 (2011) define identidade como um conjunto de atributos relacionados a uma entidade. Essa entidade pode vir a possuir uma ou mais dessas identidades e diversas entidades podem ter a mesma identidade.

Esses dados podem ser de qualquer natureza, desde que possuam alguma relação de forma direta com usuário. Os dados coletados de uma identidade por um sistema podem ser oriundos de documentos pessoais (CPF, RG), dados médicos do usuário ou até dispositivos eletrônicos ligados a ele e uma série de outras fontes. Existem 3 tipos de dados que compõe uma identidade digital.: identificador, credenciais e atributos (ZANELATO , 2014).

Os identificadores são as informações que representa uma pessoa, local ou coisa de maneira exclusiva, única e não duplicada em um dado contexto. Já credenciais é definida um conjunto de atributos e afirmações de identidade sobre uma determinada entidade que é

emitida por um IdP. E atributos seria um conjunto de dados que descreve as características de uma entidade. Estes atributos podem ser usados para descrever seu estado em um determinado domínio.

Essas informações são imprescindíveis para ajudar resolver algumas tarefas de sistemas sem a necessidade de uma ação manual e presencial do usuário. Portanto isso acaba por evitar questionamentos sucessivos ao usuário sobre uma determinada ação que necessitaria de informações para realizá-la e deixa os sistemas mais dinâmicos. Alguns dos sistemas que fazem uso desses dados são os sistemas de gerenciamento de identidades, através dos provedores de identidade contidos em sua estrutura.

Esses processos podem representar ameaça á privacidade do usuário, pois nem sempre todos os dados que são solicitados são necessariamente usados e também nem sempre o usuário estará ciente de quais dados serão usados, quem terá acesso e qual a finalidade de uso.

2.3.2 Componentes de um IdM

Os sistemas de gerenciamento de identidade são os *frameworks*, que permitem aos usuários gerenciar adequadamente suas PIIs. Assim, eles permitem que os usuários acessem recursos e serviços usando dados de identificação armazenados em provedores de identidade, dos quais um subconjunto dos atributos de identificação pode ser divulgado aos provedores de serviços (WERNER et. al, 2017).

Para que um sistema de gerenciamento de identidades possam exercer suas funções é necessário ter atores ou elementos o constituam. Cada parte ou elemento tem seu papel dentro do IMS:

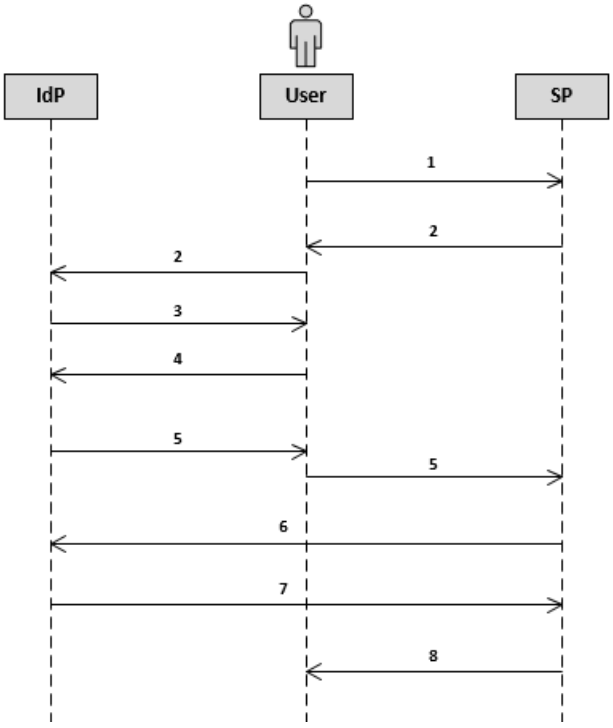
- **Usuário:** entidade que possui uma identidade que o represente que deseja acessar algum recurso oferecido pelo sistema;
- **Provedor de Identidades:** está é a parte que é responsável por armazenar e gerenciar as informações sobre os usuários. Ele emite a identidade de um usuário e após da autenticação do mesmo e recebe a identidade deste que é reconhecida e aprovada pelos provedores de serviço.
- **Provedor de Serviços:** oferece serviços a usuários autorizados, após verificar a autenticidade de sua identidade e após comprovar que a mesma carrega todos os atributos necessários para o acesso; Esses serviços são de acesso restrito e de acordo com atributos do usuário pode conceder privilégios de acesso. É responsável pela delegação da autenticação e autorização que acessam seus serviços a um IdP.

Esses componentes interagem entre si de acordo com modelos de IdM. Um dos modelos mais utilizados são os modelos federados, o qual é utilizado como padrão em sistemas de gerenciamento de identidade como Shibboleth e OpenID Connect. O Shibboleth é um sistema que utiliza o XML como troca de mensagens e é utilizada pela Federação CAFe, uma comunidade cujo objetivo é intuições congregar todas as instituições de pesquisa brasileira. Já o OpenID Connect é um protocolo baseado em OAuth2.0 e cuja a troca de mensagens no formato JSON.

2.3.3 OpenID Connect

O OpenID Connect é um protocolo de autenticação e gerenciamento de identidades (IdM) baseado no OAuth 2.0. Permite que os usuários verifiquem a identidade do usuário final com base na autenticação realizada por um Servidor de Autorização, bem como para obter informações de perfil básico sobre o usuário final de forma interoperável e *REST*. (OPENID, 2017). A Figura 1 é uma visão geral do funcionamento da autenticação do OIDC.

Figura 1 - Fluxo simplificado do OpenID Connect (BODNAR et. al, 2017).



Primeiramente temos que ter em mente que existem passos a serem seguidos que não constam no esquema da Figura 1, porém o intuito é entender o fluxo entre as partes.

1. O usuário tenta um acesso, fazendo uma requisição para o SP para acesso deu um recurso protegido.
2. A autenticação do usuário é reencaminhada para o servidor de autenticação (IdP) pois a aplicação cliente confia no resultado da mesma.
3. O IdP solicita dados credenciais, senha e *login*, para confirmar que o usuário seja quem diz ser.
4. O usuário envia os dados necessários para sua autenticação.
5. Os dados recebidos do usuário pelo IdP é validado por um sistema de autenticação e o IdP envia um *token* de autorização, comprovando sua autenticação.
6. O usuário quer acessar um recurso e ele solicita ao SP o acesso enviando o *token* de autorização que por sua vez pede a permissão ao servidor de autorização do controle de acesso.
7. O IdP envia o *token* I informações (*claims*) de atributos do usuário.
8. É realizado o controle de acesso e autorização ao recurso desejado e retornado o acesso ao usuário.

Claro que existem passos como a escolha do IdP com os dados do usuário, passos detalhado do controle de acesso não foram descritos com detalhes, mas também existem para que o OpenID Connect realize a autenticação e autorização a um recurso pertencente a uma parte.

2.3.4 Protegendo a Privacidade em IdM

Sistemas seguros como foi visto no capítulo anterior, precisam possuir uma série de princípios de segurança. Neste trabalho o objetivo lida com a privacidade e, portanto IdM precisam também cobrir essa propriedade em particular para que seja seguro e para proteger os dados de seus usuários. Segundo a norma ISO/IEC 24760-1 (2011) certos tipos de entidades, seja um indivíduo ou uma empresa, possui o direito de proteção a privacidade.

Visando esta preocupação existem algumas maneiras de IdM proteger seus e respeitar a jurisdição que define essas diretrizes. Uma forma é através da divulgação seletiva de dados, que seria o controle dos dados pertencentes ao dono dos dados, podendo limitar as informações de identidade que são coletadas ou compartilhadas com uma terceira parte. Além

dessa maneira existe também uma forma de proteção que é a divulgação mínima, que restringe uma requisição ou emissão de informações da identidade de um usuário para uma terceira parte, de forma que esta tenha o mínimo de informações necessário para a finalidade de uso.

Neste capítulo vimos mais a parte referente ao gerenciamento de identidades e uma visão superficial sobre a comunicação realizada para autenticação e autorização. No capítulo a seguir teremos uma visão mais detalhada do controle de acesso e das políticas neles contida.

2.4 CONTROLE DE ACESSO

Em sistemas de gerenciamento de identidade e acesso (IAM), provedores de identidade e provedores de serviços são geralmente componentes IdM que lidam com autenticação e autenticação. (ROSSET, 2004) A autenticação precede a autorização em um IdM .

A autenticação não é o mesmo que controle de acesso ou autorização. O controle de acesso ou autorização é a decisão de permitir ou negar a um sujeito acesso a objetos do sistema. Uma autorização estabelece quais ações são permitidas ou proibidas em um determinado sistema com a determinação dos direitos de acesso de um sujeito a um objeto computacional específico (ROSSET, 2004).

Portanto controle de acesso é um monitor de referência que controla o acesso de um usuário a recursos. Esses controles funcionam como limitações de acesso. Essas limitações são controladas de acordo com privilégios e autorizações desse sistema para acesso do usuário a um recurso específico. Para isso é importante que este usuário esteja autenticado e seja reconhecido pelo sistema que precisa acessar. Estas limitações são realizadas por tomadas de decisão realizadas por políticas de segurança, com a finalidade de prevenir que o sistema deixe de ser seguro. ROSSET (2004) resalta que um acesso não autorizado, representa um grande problema em potencial para aplicações de sistemas computacionais.

Controle de acesso deve controlar cada solicitação para um sistema e determinar, com base em regras especificadas pelas políticas, se o pedido deve ser concedido ou negado. A definição de um sistema de controle de acesso é baseada em três conceitos: políticas de controle de acesso, modelos de controle de acesso e mecanismos de controle de acesso.

Geralmente são utilizadas linguagens próprias, dependendo do sistema, para definição das políticas de controle, tornando assim um fator limitante para a concepção de sistemas

distribuídos e abertos. Para facilitar o gerenciamento e manutenção do controle de acesso, as políticas de controle de acesso estão cada vez mais escritas em linguagens de especificação como o XACML (HU et. al., 2007).

2.4.1 Modelos de Controle de Acesso

Os modelos de controle de acesso são classificados em discricionários, obrigatórios, baseados em papéis e baseado em atributos como descrevem WESTPHALL (2000) e ROSSET (2004) baseados em atributos segundo HU et. al (2007).

- **DAC:** é o controle de acesso discricionário, um modelo que permite que usuário consiga determinar de que forma as informações pode ser acessada por uma terceira parte. Neste modelo o usuário dono da informação determina quem pode ou não acessar os dados sob seu controle. Segundo MACEDO **(**2013**)** o DAC tem a possibilidade de ser combinado com outras políticas de controlo de acessos.
- **MAC:** é o controle de acesso obrigatório ou mandatório, que é um modelo que define acesso as informações usando parâmetros determinados por um sistema administrador. É determinado usando rótulos vinculados a sujeitos e objetos, comparando esses rótulos chamados de *Security Classification*. Assim a autorização poderá ou não ser concedida para o sujeito realizar uma determinada ação no objeto, através da comparação entre esses rótulos do sujeito e o do objeto (HU et. al , 2007)
- **RBAC (*Role-based Access Control*):** é um controle de acesso baseado em papéis que determina acesso as informações de um objeto de acordo com o papel ou função de um determinado usuário. Na utilização do RBAC hierárquico, os papéis, associam-se a um conjunto de operações que podem ser realizadas por um usuário ou grupo a um ou mais objetos.
- **ABAC:** um modelo de lógica AC que controla o acesso a objetos, avaliando regras em relação aos atributos de entidades (sujeito e objeto), ações e o ambiente relevante para uma solicitação HU et. al. (2017). É um modelo de controle de acesso na qual sua estrutura é construída em forma de predicados, utilizando atributos O ABAC evita a necessidade de que as permissões sejam ligadas diretamente a um usuário ou papel. Assim, quando uma requisição de acesso é solicitada, uma *engine* realiza a decisão baseado nos atributos do usuário, recurso, ambiente, entre outros. As políticas de controle de acesso que podem ser implementadas no ABAC são limitadas apenas pela linguagem computacional e pela riqueza dos atributos disponíveis. Esta flexibilidade

> permite que a maior variedade de sujeitos acesse a maior amplitude de objetos sem especificar relações individuais entre cada sujeito e cada objeto (NIST, 2013)

Os sistemas ABAC podem aplicar os conceitos de seus modelos anteriores, utilizando de suas melhores características. Neste trabalho que possui maior relevância por ser o modelo a qual foi baseado na linguagem XACML.

2.4.2 Politica de Controle de Acesso

Como comentado na seção 2.1, existem mecanismos de segurança que se usa para proteger a segurança de um usuário, as mais usadas delas são as políticas.

A política de controle de acesso é um exemplo dos mecanismos que como o próprio nome já diz, realiza o controle de acesso a recursos através de uma ação de um sujeito dando a autorização ou não de acordo com regras bem definidas nela estabelecida (SILVA et al , 2014).

Existem diversas linguagens que podem ser utilizadas para descrever uma política em um sistema de controle de acesso, uma delas é a linguagem XACML.

2.4.3 Linguagem XACML

O XACML é um padrão que determina uma linguagem declarativa, que descreve políticas de controle de acesso. É uma linguagem é uma proposta do OASIS STANDARD (2013) para modelar, armazenar e distribuir políticas descritivas de controle de acesso que utiliza o formato do XML, porém com a finalidade de tratar o controle de acesso, realizando consultas que verificam a permissão de uma ação realizada para acessar um determinado recurso. Sua estrutura pode ser utilizada para a criar políticas, contudo também é possível utilizar para fazer requisições neste formato e as respostas do sistema.

A importância de cada uma dessas partes e de suas estruturas será abordada nesse capítulo. SILVA et. al (2014) declara que no padrão XACML, é possível tratar políticas com mais flexibilidade utilizando um formato padrão de troca de mensagens, também aproveitando as técnicas de avaliação de políticas que auxiliam no controle de acesso.

2.4.3.1 Request

A requisição ou *request* no contexto de controle de acesso são mensagens emitidas por alguma aplicação de pedido de autorização para acesso aos recursos ou atributos destes para realização de alguma ação por algum sujeito em um determinado ambiente.

Pode-se observar a estrutura de uma requisição na Figura 2.

Figura 2 - Requisição no formato XACML.

```
<Request xmlns="urn:oasis:names:tc:xacml:3.0:core:schema:wd-17"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="urn:oasis:names:tc:xacml:3.0:core:schema:wd-17
                             http://docs.oasis-open.org/xacml/3.0/xacml-core-v3-schema-wd-17.xsd"
        ReturnPolicyIdList="false"
        CombinedDecision="false">
        <Attributes Category="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject">
            <Attribute IncludeInResult="false"
                       AttributeId="subject:subject-id">
                <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">"subAtt1"</AttributeValue>
            </Attribute>
            <Attribute IncludeInResult="false"
                       AttributeId="subject:age">
                <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#integer">"subAtt1"</AttributeValue>
            </Attribute>
        </Attributes>

        <Attributes Category="urn:oasis:names:tc:xacml:3.0:attribute-category:resource">
            <Attribute AttributeId="resource:resource-id">
            <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#anyURI">"rscAtt"</AttributeValue>
            </Attribute>
        </Attributes>

        <Attributes Category="urn:oasis:names:tc:xacml:3.0:attribute-category:action">
            <Attribute AttributeId="action:action-id">
            <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">"actAtt"</AttributeValue>
        </Attribute>

        <Attributes Category="urn:oasis:names:tc:xacml:3.0:attribute-category:environment">
            <Attribute  AttributeId="environment:time">
                <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#integer">"envAtt"</AttributeValue>
            </Attribute>
        </Attributes>
</Request>
```

Além dos agrupamentos mencionados nesse capítulo, é possível criar outros tipos não padronizados. Pode-se observar que o exemplo da Figura 2, existem as 4 categorias de atributos citadas anteriormente. A primeira é a de *subject*, que representa o sujeito que está usando o sistema. Na requisição podem ir dados relacionados a ele. A segunda categoria é a *action* relacionada a ação do usuário. A terceira é a categoria r*esource*, relacionada ao recurso a ser acessado de acordo com a permissão. E por fim, o *environment*, que é um conjunto de atributos que são relevantes para uma decisão de autorização e são independentes de um sujeito, recurso ou ação.

Todas essas categorias podem possuir um ou mais atributos, cada atributo possui um único tipo de dado (*data type*) e um identificador (*AttributeId*) e podem possuir um ou mais valores.

Para que seja possível gerar uma *bag* ou conjunto de dados, pode-se utilizar a estrutura de um tipo *bag* através de uma função *Match* e passar todos os dados necessários. Porem existe outra maneira, que é criar um valor para um mesmo *AttributeId* de mesmo tipo e mesma categoria. Essa última é uma opção melhor, pois é possível utilizar numa requisição no formato JSON.

Existe também uma função *includeResult opcional* para utilizar com o fim de mostrar na resposta o resultado da combinação do dado passado pela requisição e pelo *target* da política, comparada pelo PDP.

2.4.3.2 Response

Uma Resposta ou response consiste em um ou mais resultados após a avaliação da política. Sua estrutura é bastante simples conforme a Figura 3. A resposta assim como os demais arquivos XACML possui um cabeçalho com dados que costumam ser padrão.

Figura 3 - *Response* no formato XACML.

```
<Response xmlns="urn:oasis:names:tc:xacml:3.0:core:schema:wd-17"
          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
          xsi:schemaLocation="urn:oasis:names:tc:xacml:3.0:core:schema:wd-17
                              http://docs.oasis-open.org/xacml/3.0/xacml-core-v3-schema-wd-17.xsd">
    <Result>
        <Decision>Permit</Decision>
        <Status>
            <StatusCode Value="urn:oasis:names:tc:xacml:1.0:status:ok"/>
        </Status>
    </Result>
</Response>
```

Deny, *Not Applicable* ou *Indeterminate*, que ocorrem respectivamente quando é concedida, negada, ou não é aplicada, ou contém algum erro no sistema como quando não se encontra uma política através do *target*. O campo *Status* resulta no valor "*Ok"* caso o resultado seja um resultado válido ou ele apontará um erro caso exista um problema, por exemplo, quando uma indeterminação por não achar uma política.

2.4.3.3 Política

Uma política criada na linguagem XACML possui uma estrutura bem definida para que seja possível ser processada. Para seu funcionamento, também precisa de um cabeçalho padronizado, contudo, não está ilustrado na Figura 4, mas podemos observar as principais estruturas que a compõe.

Figura 4 - Conjunto de política no formato XACML.

```
<PolicySet PolicySetId="..." policyCombiningAlgId="...">
    <Target>
            <Subjects>    ...  </Subjects>
            <Resources>   ...  </Resources>
            <Actions>     ...  </Actions>
            <Environment>...   </Environment>
    </Target>
    <Policy PolicyId="..."  RuleCombiningAlgId="...">
        <Target>
            <Subjects>    ...  </Subjects>
            <Resources>   ...  </Resources>
            <Actions>     ...  </Actions>
            <Environment>...   </Environment>
        </Target>
        <Rule RuleId="..." Effect="...">
            <Target> ...</Target>
            <Condition FunctionId="...">...</Condition>
        </Rule>
        <Obligation>
            <Obligation ObligationId="..." FulfillOn="...">...</Obligation>
        </Obligation>
    </Policy>
</PolicySet>
```

Existem alguns elementos que precisam estar presentes ou que podem vir a fazer parte do formato de uma política de privacidade no XACML. As duas estruturas que podem iniciar um documento de política e possui o cabeçalho em sua composição é a ***Policy*** e ***PolicySe****t*. Em sua composição precisa conter um algoritmo de combinação e um identificador da politica ou conjunto de politicas. É esta estrutura que o PDP deverá procurar através do *target* dentro dele. A política possui uma série de regras e um conjunto de políticas pode possuir um conjunto de políticas e políticas.

Dentro de uma estrutura de política podem existir as estruturas ***Rule*** que é a regra responsável por retornar para a política uma decisão de acordo após avaliações feitas em seu escopo com atributos passados por requisição e dados pré-definidos no documento. Como o escopo da regra só recebe valores booleanos, ela precisa além de um identificador de um a definição de efeito chamado ***Effect.*** Este efeito é o que deverá retornar para a política caso as verificações retornem *true* ou *false*, ela pode possuir os valores *Permit* ou *Deny*, que são os valores possíveis esperados pelo usuário. Ela existe apenas no escopo da regra.

Outro elemento grande importância são os Algoritmos de combinação, que são definidos em dois tipos: ***Policy Combining algorithm*** e ***Rule Combining algorithm*****.** Eles são algoritmos que são declarados, mas estruturas de políticas ou conjunto de políticas para combinar a decisão e as obrigações de múltiplas políticas ou regras, a fim de escolher como aplicar o acesso, as politicas e conjunto de politicas possuem alguns algoritmos de combinação de politicas ou de regras consecutivamente. Os principais algoritmos definidos pela especificação da OASIS (OASIS STANDARD, 2013) e estão na Tabela 1.

Tabela 1 - Tabela de algoritmos de combinação do XACML

Algoritmo	Descrição
Deny Overrrides	A decisão retornará "Deny" se, pelo menos, um elemento negar a autorização.
Permit Overrrides	A decisão retornará "*Permit*" se, pelo menos, um elemento permitir a autorização.
First Applicable	A primeira regra ou política aplicável será avaliado e o resultado final da decisão dada por ele. Neste caso a ordem das políticas faz a diferença.
Only one applicable	Este caso só funciona para combinação de políticas. Se dentro de um conjunto de politica existe apenas uma politica aplicável, retorna seu resultado, caso contrário devolve indeterminado a decisão.
Permit unless deny	A decisão retornará "*Permit*" quando um elemento permitir a autorização. Diferente do *permit-overrides*, este algoritmo desconsidera os casos de indeterminação e não aplicação de um elemento.

Fonte: OASIS STANDARD, 2013.

O ***Target*** é o elemento usado para que o PDP possa localizar a política, através da comparação do valor e tipo de dado definido por atributo contido na categoria a qual ele pertence. As categorias padrões são as mesmas que existem na requisição, *Subject*, *Action*, *Resource* e *Environment*. Ela pode existir ou não num conjunto de política, política ou regra. Quando ela não é definida, qualquer dado passado pela requisição pode usar a política. Os *targets* são o controle de acesso a politica, regras ou conjunto de políticas.

Uma ***Function*** são as funções são as diferentes ações que podem ser realizadas com os atributos de dados recebidos ou dados definidos na política. Esses dados possuem tipos definidos. Esses tipos e essas funções são oferecidos no próprio escopo de estruturas como *Condition*, *match* e *apply*, pois esses realizam suas aplicações.

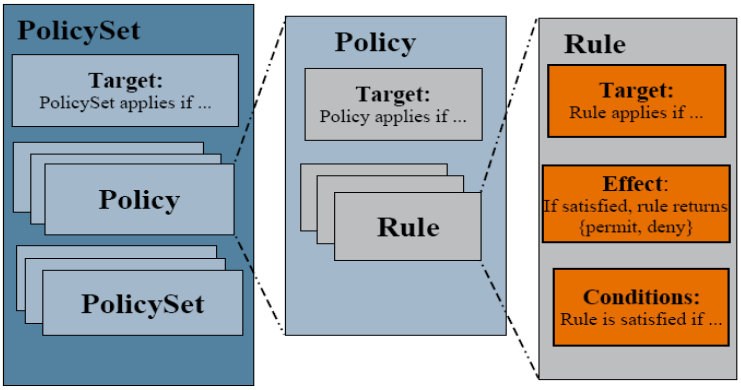
A condição ou ***Condition*** é uma estrutura existente dentro de uma a regra que realiza a ação de uma determinada função. Ela retorna *True* ou False de acordo com a execução da função que está sendo executada. Ela pode definir o efeito de uma regra com seus resultados, porém não é um item que precisa constar numa regra de forma obrigatória.

Um dos objetivos do XACML é fornecer um controle de acesso de nível mais fino do que uma simples autorização e negar decisões (ROSSET, 2004). Opcionalmente algumas políticas podem utilizar estruturas ***obligation*** em seus conjuntos de politicas, politicas ou regras. Essas obrigações precisam ser cumpridas para que o acesso ao recurso seja permitido ou negado. As obrigações são ações que podem vir a ser realizadas pelo PEP em junto a execução de uma decisão de autorização

Um conceito importante é a estrutura das *bags*, que segundo explicado pela OASIS STANDARD (2013), é uma coleção desordenada de valores, na qual pode haver valores duplicados. A *bag* é um conjunto de dados de um determinado tipo que podem ser passados por uma requisição ou podem ser declarados em uma política para realizar ações geradas pelas funções.

Uma representação diferente de como poderia criar uma política de forma mais didática é como ilustra a Figura 5.

Figura 5: Estrutura simples de uma política XACML (SINNEMA, 2011).



Este modelo representa de forma um pouco mais clara a relação entre alguns dos elementos principais da estrutura de uma política na linguagem XACML. É importante salientar que apesar da falta dos algoritmos de combinação na imagem, eles são fundamentais para o funcionamento do modelo.

2.4.5 Componentes da Arquitetura de Controle de Acesso

Primeiramente é importante entendermos a função dos elementos que compõe o controle de acesso das políticas XACML, para que possam ser entendidos os passos e o fluxo geral realizado para a obtenção da decisão a partir da descrição oferecida.

Com a noção dada no capítulo anterior sobre a linguagem XACML , é possível compreender a estrutura e como a requisição enviada retorna uma resposta para o usuário.

Começando com o ponto de administração da política (PAP), que é a entidade do sistema responsável por criar, gerenciar e armazenar as políticas ou conjunto de políticas.

O ponto de decisão política, o PDP, é um dos elementos mais importantes, pois é ele quem avalia as políticas e processa uma decisão de autorização de acesso a um recurso, caso política exista no PAP ou tenha aplicabilidade para o fim desejado.

O ponto de aplicação de uma política ou PEP também tem uma grande importância. Ele recebe as requisições de, as envia para o PDP e, por fim, também é o responsável por aplicar as decisões de autorização de acesso avaliado pelo PDP. Além de encaminhar as requisições ele também é responsável pela interpretação da resposta obtida

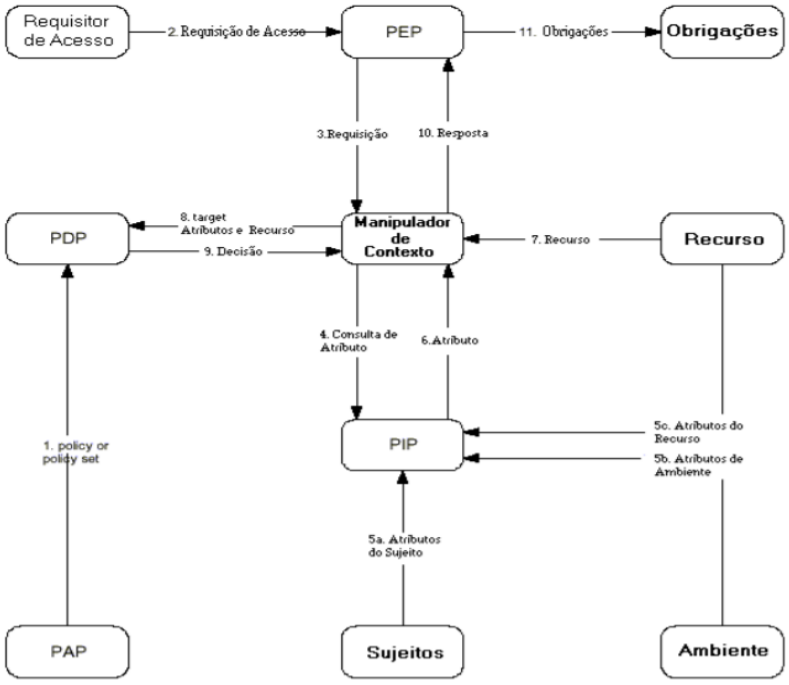
O ponto de informações de política ou PIP atua como uma fonte aonde é possível coletar os valores dos atributos de um sujeito, recursos e ambiente. Ele pode estar definido num sistema como dados de bancos SQL ou até mesmo estruturas XML. Quando uma requisição não possui todos os valores suficientes em seu escopo, o PDP pode utilizar dados fornecidos pelo PIP para resolver.

Por fim o *Context Handler*, é uma espécie de intermediário entre o PDP , PEP e PIP, e segundo pela OASIS STANDARD (2013), uma entidade responsável pela conversão das solicitações de decisão no formato de solicitação nativo para o um contexto do XACML e vice-versa, além de coordenar com PIP o envio de valores de atributo ao contexto de requisição quando necessário.

2.4.6 Fluxo de Controle de Acesso

O usuário após a autenticação em uma aplicação gerida por um IdM, necessita estar autorizado para acessar o recurso desejado. A maneira adotada neste trabalho segue esta arquitetura e fluxo. Ao entender os componentes dessa arquitetura relacionada na **Figura 6** o é possível acompanhar o fluxo do processamento das politicas de controle de acesso.

Figura 6 - Fluxo do controle de acesso na politica XACML (OASIS STANDARD, 2013).



O fluxo do controle de acesso, baseando-se no modelo do OASIS STANDARD (2013) através das seguintes etapas de acordo com a Figura 6.

1. O **PAP** escreve as políticas e as disponibiliza para que o **PDP** as utilize.
2. O usuário (um *subject*) envia uma requisição de acesso ao **PEP**.
3. O **PEP** envia a requisição ou ***Context Handler*** em seu formato de solicitação nativo.
4. O ***Context Handler***, por sua vez, monta o contexto da requisição XACML, podendo adicionar atributos e envia ao **PDP**.
5. O **PDP** solicita atributos adicionais do sujeito, recurso, ação ou ambiente do ***Context Handler***.
6. O ***Context Handler*** solicita os atributos do **PIP**.
7. O **PIP** coleta os atributos solicitados.
8. O **PIP** retorna os atributos solicitados para o ***Context Handler***.
9. O ***Context Handler*** pode incluir o recurso no contexto.
10. O ***Context Handler*** realiza a emissão dos atributos solicitados e outros dados que podem vir a estar no contexto para o **PDP,** que por sua vez, avalia a política.
11. O **PDP** retorna o contexto de resposta (com decisão de autorização) ao ***Context Handler***.

12. O ***Context Handler*** monta o contexto de resposta para o formato de resposta nativa do **PEP** o retorna a resposta.
13. O **PEP** cumpre as obrigações caso existam.
14. De acordo com o resultado o **PEP** nega ou permite o acesso ao usuário solicitante.

2.5 TRABALHOS RELACIONADOS

Neste capítulo serão citadas algumas referências de trabalhos relacionados com o controle de acesso e privacidade de dados em um ambiente de computação em nuvem. Portanto será teremos uma visão geral de cada trabalho a fim de fazer um comparativo com o que será proposto neste trabalho.

Com o intuito de uma melhor gestão e aplicação do controle de acesso visando a privacidade sobre dados pessoais e sensíveis coletados e armazenados pelas empresas, MONT (2011) propõe e trata confidencialidade no controle de acesso sobre os dados pessoais e sensíveis voltados para empresas. Ele gerencia o controle de acesso de uma maneira diferente da forma utilizada neste projeto, abordando mais o tratamento da privacidade em nível de banco de dados, ou melhor, trata especificamente o problema da aplicação de políticas de privacidade em dados pessoais armazenados em uma ampla variedade de repositórios de dados dentro das empresas, podendo ter os dados pessoais acessados por diferentes tipos de solicitantes (aplicativos, pessoas, serviços). No trabalho proposto possui-se o mesmo objetivo trata a privacidade com consentimento e consciência do usuário, no entanto a forma como foi tratado controle de privacidade na proposta precisa apenas do *token* cadastrado e seu processamento é apenas no SP, diferente da proposta de MONT (2011), ele inclui aspectos relacionados de modelagem de dados gerenciados e criação de políticas de privacidade.

Devido a inexistência do controle por parte do usuário do compartilhamento de seus dados pessoais armazenados em provedores de serviços, o trabalho proposto por VILLAREAL (2017) apresenta um modelo de definição de privacidade para sistemas de gerenciamento de identidade, o qual permite a um usuários cadastrados em um IdP configurar suas preferências de privacidade de acordo com uma série perfis padrões oferecidos pré-definidos ou personalizado por ele, gerando uma estrutura nomeada pela autora de *token* de privacidade. Esse mecanismo desenvolvido por VILLAREAL (2017) foi crucial para o desenvolvimento das políticas de privacidade desenvolvidas no trabalho desenvolvido. A

ideia do *token* foi mantida com as mesmas características, atributos e objetivo, porém com um formato diferenciado para melhor coleta de dados na aplicação desenvolvida.

Para uso de serviços fornecidos pelos provedores de serviços, precisa-se muitas vezes uso de dados além da identidade para que possam atender as necessidades das políticas no controle de acesso, a proposta de CAMILLO et al (2017) , é uma arquitetura que permite um individuo acessar algum recurso ou e filtrando os dados pessoais desnecessários para tal operação. Para isso as políticas de controle de acesso sã o enviadas para o IdP tirando a responsabilidade de fazer a decisão do SP. Com a análise dos dados no lado do IdP garante que apenas os dados que realmente serão úteis para o serviço estarão presentes nele, protegendo dessa forma os dados sensíveis do usuário. Podemos ver que trata da privacidade em relação a dados coletados, porém na proposta deste trabalho, tratam as políticas de privacidade em relação ao consentimento do usuário e não a necessidade de uso de dados no serviço, porém ambos tem a mesma proposta de proteção de privacidade.

O trabalho de WERNER e WESTPHALL (2016), propõe um modelo de gestão de identidade com privacidade para ambientes de nuvem usando o OpenID Connect como IdM com objetivo de reduzir os riscos de violação de privacidade. A proposta aborda três questões: a falta do controle dos dados sensíveis de um usuário por ele, a falta de modelos para auxiliar os usuários na disseminação de dados durante a interação e a falta de garantias de preferência dos usuários no SP. Tendo isso em vista, podemos relacionar a proposta deste trabalho como uma parte a ser usada para gerar politicas de privacidade bem definidas a fim de participar no controle dos dados sensíveis em conjunto com o trabalho proposto por VILLAREAL (2017).

3 DESENVOLVIMENTO PRÁTICO

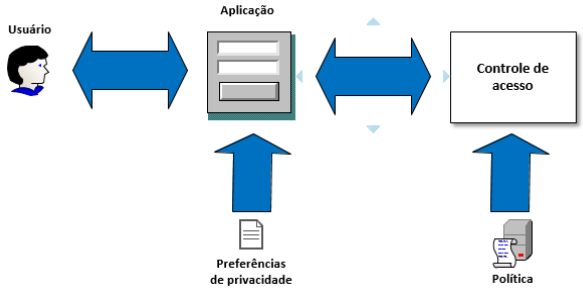
Este capítulo descreve o desenvolvimento prático do trabalho. Primeiramente será apresentada a proposta para entendimento do que será esperado no fim deste trabalho. Também serão citadas as ferramentas que serão utilizadas e os passos que serão realizados para as instalações, configurações e uso das ferramentas existentes e desenvolvidas no decorrer do projeto.

3.1 PROPOSTA

Para este projeto, é importante entender que para obter uma permissão de acesso é preciso que **sujeitos** realizem **ações** para ter acesso a **recursos.** Porém as ações nem sempre utilizam todos os dados coletados e esses dados muitas vezes são sensíveis. O dono dos dados podem não saber quais dados estão sendo usados, nem para qual finalidade e quem fará uso desses dados.

Considerando a privacidade, gerenciamento de identidades e a parte que trata de controle de acesso, será possível entender o que será feito neste trabalho. A proposta se resume em modelar e utilizar políticas de privacidade utilizando uma linguagem de controle de acesso para sistema de autorização de um sistema de gerenciamento de identidades, para qualquer aplicação *Web* que faca uso de informações pessoal (PII) de uma determinada entidade. Para que isto seja possível precisa-se de vários componentes.

Figura 7 - Proposta de controle de privacidade que será desenvolvida.



O esperado, conforme a Figura 7, é que ao fim deste projeto o usuário consiga enviar uma requisição para acesso a dados através de uma aplicação e receber uma resposta de permissão ou negação de acesso de acordo com as preferências de privacidade coletadas do usuário, dono do recurso a ser acessado, para serem usadas para comparação com as regras de privacidade construídas antecipadamente pelo usuário em um sistema controle de acesso.

Os problemas de privacidade relacionados á finalidade de uso, tipos de dados coletados pelo usuário e quem é este usuário beneficiário necessita de controle para a parte que armazena estes dados. Para este modelo, não será usado o IdP; os dados necessários para que sejam feitas as comparações e testes serão gerados por ferramentas criadas nesse trabalho que deverão estar presentes na requisição. Portanto, as trocas de mensagens do SP ou sistema de autorização com o IdP não está no escopo deste trabalho, mas serão consideradas e estudadas para fins de implementação.

As ferramentas desenvolvidas geram as preferências de privacidade e fazem a leitura dessas preferências para gerar uma requisição a ser enviada pela aplicação.

Esta proposta fará uso de um sistema que realiza controle de acesso, que poderá ser adotado por um IdM, como o OpenID Connect. Desta maneira, poderão ser aplicadas politicas de privacidade no formato XACML como diretriz para a tomada de decisão de acesso.

3.2 FERRAMENTAS E PADRÕES UTILIZADOS

Foi necessário instalar algumas ferramentas e entender alguns padrões para garantir a implementação correta e garantir uma aplicabilidade da proposta.

A linguagem escolhida para a implementação é o Java devido a frameworks e bibliotecas que serão usadas, além de que algumas ferramentas são compatíveis apenas com esta linguagem. Ferramentas que foram usadas, porém podem ser usadas outras, é o Eclipse Neon 3 (ECLIPSE NEON, 2017), o Jetty (JETTY, 2017) através do *plugin*, o Maven contido no Eclipse,

3.2.1 O Formato JSON

JSON (*JavaScript Object Notation*) é um formato leve de intercâmbio de dados baseado em um subconjunto da linguagem de programação JavaScript (ECMA-404, 2011).

Sabemos que o formato JSON é um padrão utilizado para muitos fins, uma delas é troca de mensagens para alguns sistemas de gerenciamento de identidade. É de fácil utilização para humanos e para máquinas.

Esta proposta se baseia no IMS OpenId Connect, foi escolhido o JSON como formato para os *Tokens* e Requisições geradas além de ser um formato simples para se trabalhar e com bibliotecas bem consolidadas em Java.

As requisições padronizadas em Java para controle de acesso têm menos recursos XACML por falta de algumas funções que poderiam ser utilizadas, porém para a finalidade que precisamos neste trabalho foi suficiente.

3.2.2 Linguagem XACML

O XACML já foi explicado na seção 2.4 porém será citado novamente para esclarecer como foi usado nesse projeto.

XACML é um padrão OASIS que descreve uma linguagem de política e uma linguagem de solicitação / resposta de decisão de controle de acesso (ambos escritos em XML) (SPRING, 2017)**.** A especificação do OASIS XACML (2017) define uma linguagem de política de controle de acesso com base em atributos, uma arquitetura e um modelo de processamento descrevendo como avaliar requisições de acesso de acordo com as regras definidas.

XACML também pode ser usada para montar formatos de requisições e as respostas geradas que serão o resultado da decisão dos pontos gerados pelo PDP (Ponto de decisão). Neste trabalho está sendo usada a ferramenta OpenAZ implementa XACML apenas para montar as politicas de privacidade. As requisições e respostas são geradas em JSON, pois no OpenAZ não é possível usar JSON.

3.2.3 Spring Framework

O *Spring Framework* é uma estrutura de aplicação e inversão de contêiner de controle para a plataforma Java. Os recursos principais da estrutura podem ser usados por qualquer aplicativo Java, mas existem extensões para a construção de aplicativos da *Web* com a plataforma Java EE (SPRING, 2017).

O uso dos recursos do *Spring* são muito uteis e tornam as coisas muito mais práticas na hora de desenvolver, porém a quantidade de arquivos de configurações e instalações de ferramentas em conjunto pode ser bastante incômoda. Por isso foi considerado o uso do *Spring Boot*.

O *Spring Boot* é um facilitador que visa à criação de aplicações autônomas, baseadas em *Spring* (SPRING BOOT, 2017). Foi escolhida, pois facilita a implementação de um sistema e oferece uma maneira rápida de criar aplicativos. O *Spring Boot* verifica o *classpath* e configurado, faz suposições razoáveis sobre o que está faltando e adiciona. Além disso, é compatível com a ferramenta que simula o controle de acesso, o OpenAZ, que também utiliza essa tecnologia em sua concepção. A maioria das aplicações *Spring Boot* necessita de pouca configuração *Spring* (SPRING BOOT, 2017). Através do Maven no Eclipse e de configurações realizadas no arquivo pom.xml, onde ficam as dependências tem uma série de ferramentas para banco de dados, um Apache Tomcat embutido, entre outros.

Pode-se, além disso, criar um projeto com as ferramentas *Web* que precisar já configurada no pom.xml e a estrutura formada usando um site de inicialização de projeto do Spring (SPRING INITIALIZR, 2017).

3.2.4 HTTP Requester

Esta é uma ferramenta instalada no navegador *Chrome* ou Firefox como *plugin*. Usada para quem desenvolve na *Web* ou *REST*, ou quando precisa-se fazer solicitações HTTP que não são feitas facilmente através do navegador (*PUT* / *POST* / *DELETE*) (MOZILLA, 2017). O HTTP Requester foi usado com o PDP do OpenAZ, para não ter necessidade de criar uma aplicação para enviar uma requisição para o *REST*. O histórico gerado pelas requisições facilita a reexecução das mesmas. Aceita vários formatos de requisição como XML, JSON, podendo ser criado na própria interface ou podendo usar um arquivo próprio.

3.2.5 OpenAZ

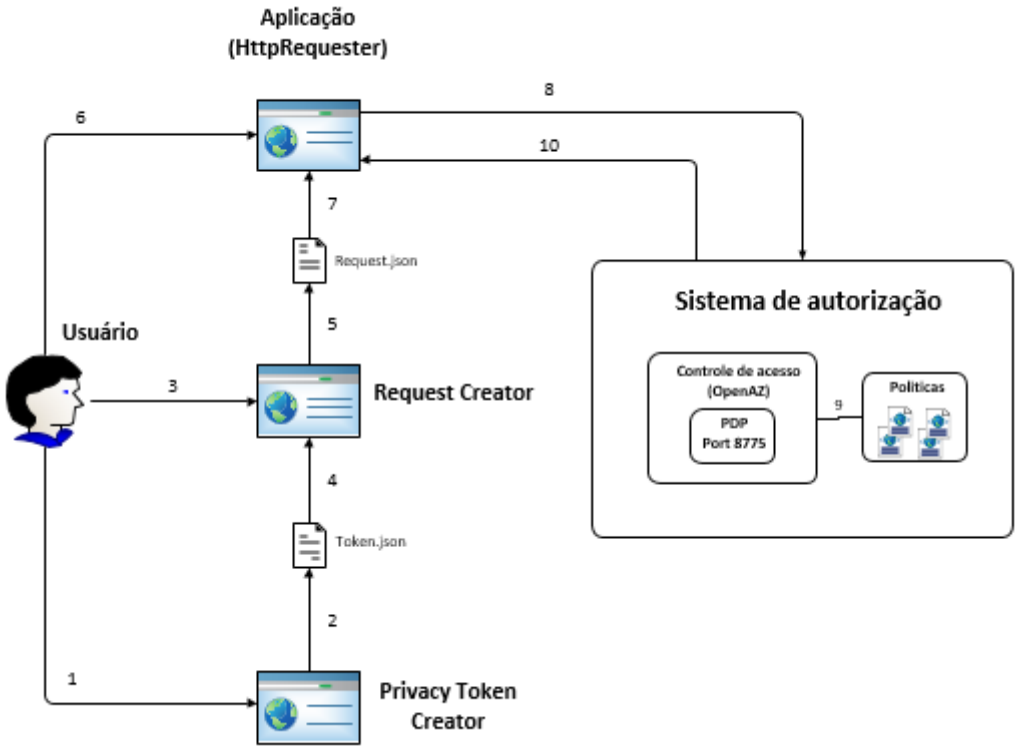
OpenAZ é um projeto que permite o desenvolvimento de sistemas de controle de acesso baseados em atributos ABAC em uma variedade de linguagens. Gera os principais pontos de processamento do XACML para fazer os testes do controle de acesso (APACHE, 2017).

Foi desenvolvida pela AT&T e incubado pela Apache até 2016. Foi escolhida por ter o código aberto e em Java e por ter sido usado por um dos membros do laboratório, e dessa forma seria mais acessível um suporte .

Neste projeto foi usada apenas a parte que faz a decisão, o PDP, criando um *endpoint REST*. Existem módulos para lidar com o PEP e PEP, mas não foram usados neste projeto.

3.3 INSTALAÇÕES E IMPLEMENTAÇÕES PRÁTICAS

Figura 8 - Implementação do modelo da proposta.



A Figura 8 - Representa as implementações desenvolvidas.

Inicialmente no passo 1, é utilizado um sistema para criar o *token*, o *Privacy Token Creator*, que num modelo real deveria estar em um IdP ou ser um sistema integrado ao gerenciador de identidade e por fim, ser cadastrado junto aos dados do usuário. O passo 1 é o preenchimento do *token* com suas preferências. No passo 2, sistema gera um *token* em formato JSON. Cada usuário possui seu próprio *token*.

O usuário no passo 3 gera uma requisição através de um sistema. Este sistema lê o *token* gerado anteriormente e gera uma requisição em arquivo JSON para ser enviado posteriormente, como ilustrado a Figura 12, o passo 4. Esse sistema é o *RequestCreator*, que

possui um método responsável por coletar as preferências ativas, ou seja, as preferências tenham valor 1. Em um modelo real essa função seria necessária para demonstrar como seria feito para coletar esses dados do *token* de privacidade.

Depois de gerar a requisição em formato JSON, no passo 6 o usuário solicita o envio desta requisição, onde é coletada a requisição gerada no passo 7, através de um *plugin* que existe no Firefox, o HTTP Requester (HTTPREQUESTER, 2017) responsável simplesmente por enviar a requisição ao sistema de controle de acesso.

No passo 8, o sistema de controle de acesso no projeto será apenas tratado pelo PDP da ferramenta OpenAZ instalada na maquina e configurada para uma porta com ajuda de um servidor, neste caso o Jetty(JETTY, 2017). O Jetty é um simulador de um sistema de controle de acesso, e possui vários módulos que podem ser usados caso o projeto tenha uma dimensão maior.

No passo 9 o PDP lê a requisição e tenta encontrar uma política que seja responsável por este controle de acesso ao recurso solicitado. Essas políticas, são desenvolvidas manualmente utilizando a linguagem XACML, porém agora adotando um modelo de controle de privacidade desenvolvido, utilizando para comparação os *tokens* de privacidade vindo no *request*. Em um modelo real, as informações referentes ao *token* de privacidade poderiam ser coletadas pelo PIP, para não precisar ter os dados na requisição.

No passo 10, a resposta de permissão é retornada no navegador para o usuário.

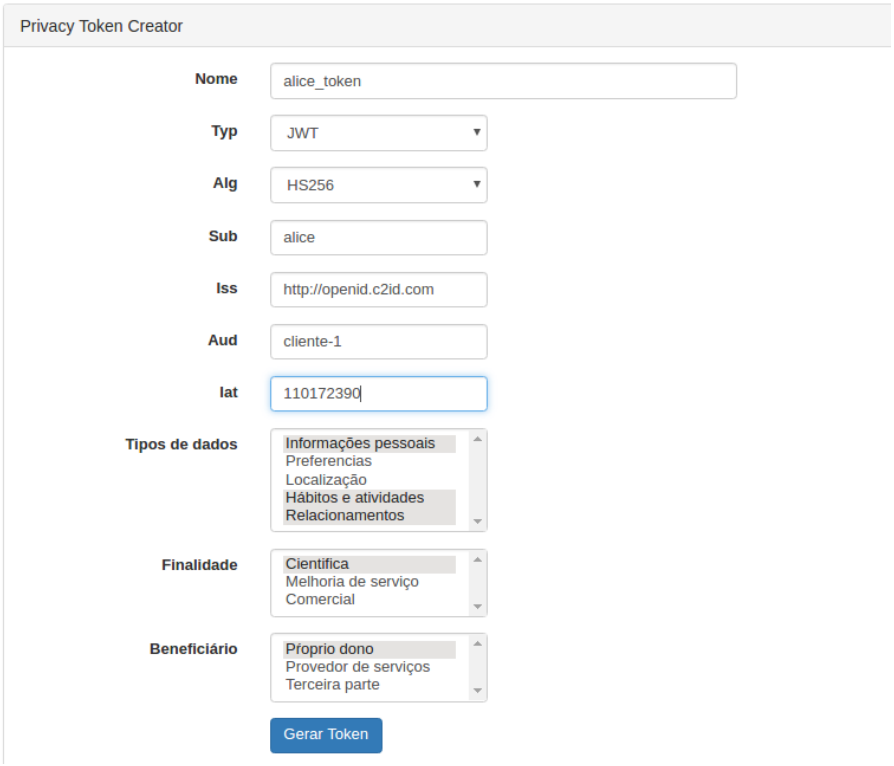
3.3.1 Desenvolvendo o Privacy Token

O *Privacy Token* concede ao usuário uma maneira de definir como deve ser o uso das suas informações e declarar suas preferências para um determinado sistema. O *Privacy Token* é um arquivo JSON com informações pré-definidas usado por políticas realizar controle de acesso.

O *Privacy Token Creator* é uma ferramenta que gera as preferências de privacidade se determinado usuário. Foi implementada em Java utilizando a ferramenta Eclipse Neon 3 e uma biblioteca para criar um objeto JSON chamada Json-Simple (JSON SIMPLE, 2017).

Para compreender melhor o código é necessário entender o que é e como é formado o *Privacy Token* e o que o compõe. A figura 9 representa a interface do Privacy Token Creator.

Figura 9 - Interface do *Privacy Token Creator*.



Como é possível observar na Figura 9, existe uma série de campos para preencher na configuração do *token* de privacidade. Primeiramente o campo nome, que é o nome do arquivo a ser gerado, a seguir o cabeçalho que possui os dados Typ, que é o tipo de estrutura que a princípio está fixo no padrão JWT. JWT é um padrão aberto que define como transmitir JSON *objects* de forma segura entre aplicações, o Alg, que é o algoritmo usado, no caso seria fixo no HS256 e essas informações concluem o cabeçalho (RFC 7519, 2017).

Os atributos, também conhecidos como *claims*, são o Sub (*subject*) que identifica o sujeito, o Iss (*issuer*) que representa quem emite, o Aud (*audience*) que identifica os destinatários para quem se destina, Iat (*issued* at) identifica o momento em que foi emitido. Esses dados foram mantidos com este nome pois além de ser o nome usado pelo sugerido no trabalho de VILLAREAL (s.d), também são termos padrões para JWT. Por fim, as preferências, que são atributos de privacidade criados para agrupar as preferências de privacidade e oferecer dados e que serão usados juntos com as politicas para tomar a decisão de acesso.

O campo Tipos de dados é um grupo que representa um determinado tipo de dados. Os tipos de dados usados são: Dados pessoal (PI) que agrupa dados que identificam o usuário, Características pessoais e preferência (PCP) agrupa dados que o usuário escolhe ou um dado que não necessariamente o identifica, Localização (LO) dados de localização que podem ser

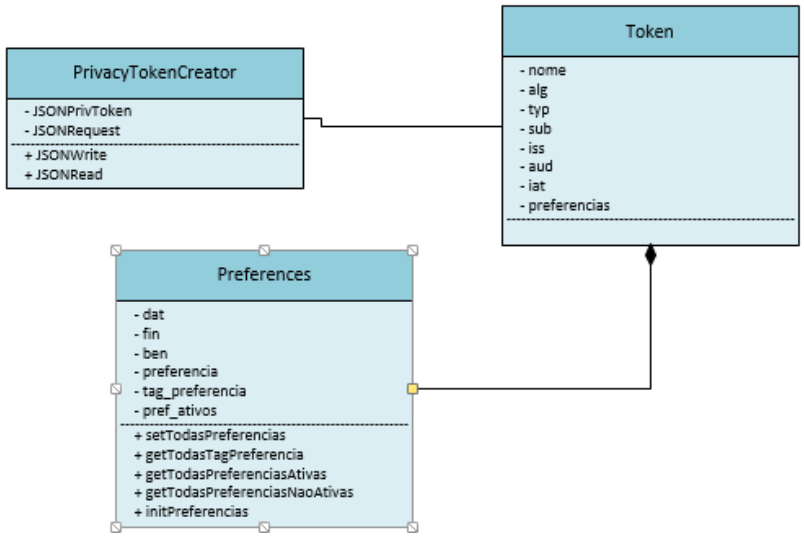
coletadas por dispositivos, Atividades e Hábitos (AH) dados relacionados a ações ou históricos e por fim Relacionamentos (RS) dados que contem as amizades, relacionamentos ligado a outras partes.

O campo Finalidade permite selecionar para quais finalidades os dados podem ser usados. Os tipos de finalidades definidos são: Cientifica (SC), Melhoria de Serviço (SI) e Comercial (CO).

O campo Beneficiário é quem fará uso dos dados e pode ser: Próprio dono (PP), terceira parte (TP) e provedor de serviços (SP). Os campos de preferências representados na Figura 9 pelos campos Tipos de dado, Finalidade e Beneficiário, são de múltipla seleção e precisa que ao menos 1 seja selecionado. Após preencher os dados, pode-se gerar um arquivo JSON que será usado para definir os dados a serem usados.

O *Privacy Token Creator* foi criado utilizando 3 classes, JSONCreator, *Token*, *Preferences*, como ilustra a Figur

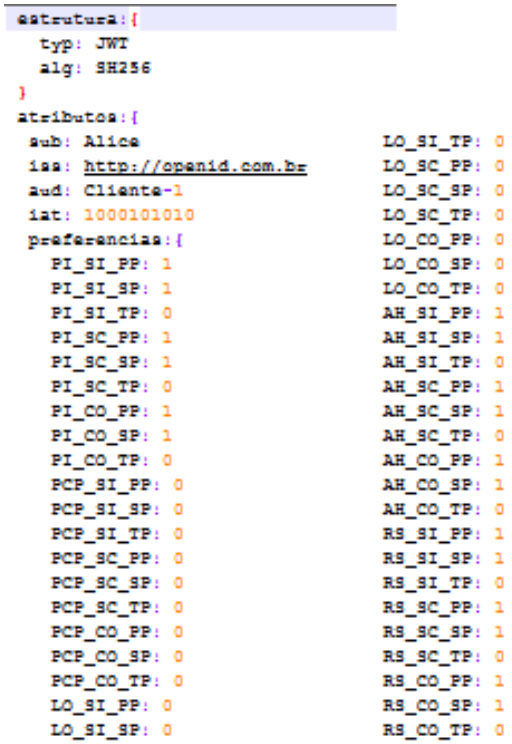
Figura 10 - Classes do *Privacy Token Creator*



A classe *JSONCreator*, tem como responsabilidade ler um arquivo JSON, em um diretório configurado, no formato do token através do método *JSONRead* e escrever da mesma maneira usando *JSONWrite*. Essas escritas e leituras dos documentos são feitas com uso do objeto *Token* da classe *Token*, o qual possui as informações que pertencem ao *Privacy Token*. Uma dessas informações são as preferências, os dados mais relevantes do *token* para este trabalho. As preferências estão em *arrays*: ben, dat e fin, cada um deles com string relacionados a beneficiário, tipos de dados e finalidades citadas na explicação do *Privacy*

Token. Essas *strings* foram usadas para montar a tripla, no caso um *tag* através da concatenação separada pelo caractere "_". Cada tripla criada na combinação dos dados dos 3 *arrays* é armazenada no *array tag_preferencia* e o valor atribuído a ela estará em outro *array* chamado preferência. O valor de uma *tag* é representado no mesmo índice do outro *array*. Assim, de acordo com os dados vindos da escolha do usuário nos *arrays* finalidade, beneficiário e tipo de dados foi possível ver se um dado estava ou não contido nos *arrays* correspondentes da classe preferências, assim gerando a lista de preferências como ilustra a Figura 11.

Figura 11: Resultado gerado do *privacy token.*



O documento gerado é um pouco diferente da forma como é mostrada na interface na Figura 9, já que está abreviado. Suas siglas são combinadas a fim de formar uma tripla com um elemento de cada campo de preferência (Tipos de dados, finalidade, beneficiário) para armazenar 1 para caso todos os elementos da tripla das siglas tenham sido selecionadas nos campos de preferências da Figura 10 ou 0 no caso de algum elemento da tripla não tenha sido selecionada. Na Figura 11, as preferências são apenas para PI_SC_PP, ou seja, acesso a apenas dados pessoais, com a finalidade científica, para uso do próprio dono do dado, qualquer outra configuração não poderia funcionar.

Com o *token* gerado, é possível usá-lo para gerar a requisição, assim.

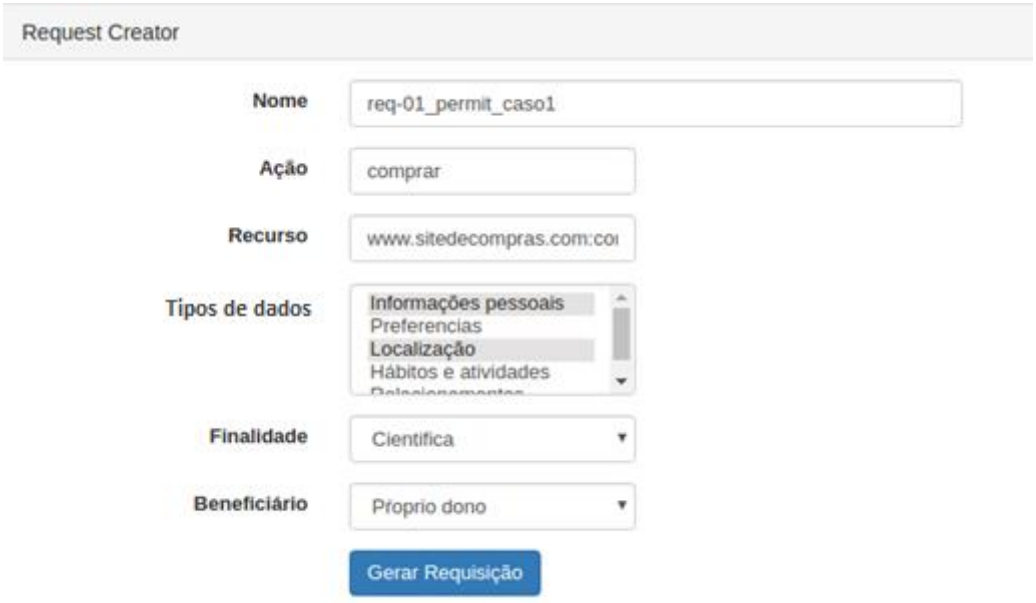
3.3.2 Desenvolvimento do Request Creator

Apesar da facilidade da leitura e da escrita do JSON, quando se precisa fazer testes com muitos itens na requisição ou quando precisa-se tratar algum dado antes de colocá-lo no documento, uma ferramenta se faz necessária para facilitar esse trabalho para um usuário. O *Request Creator* foi criado para fins de teste e para facilitar a montagem da requisição para este modelo de politica de privacidade. As requisições em aplicações *Web* geralmente são geradas de acordo com as ações feitas nessas aplicações e já são pré-estabelecidas.

Como foi criado para testar politicas criadas nesse trabalho, o *Request Creator* tem uma estrutura definida e utiliza o *token* gerado pelo *Privacy Token Creator* que deverá estar no diretório onde está o código a ser lido.

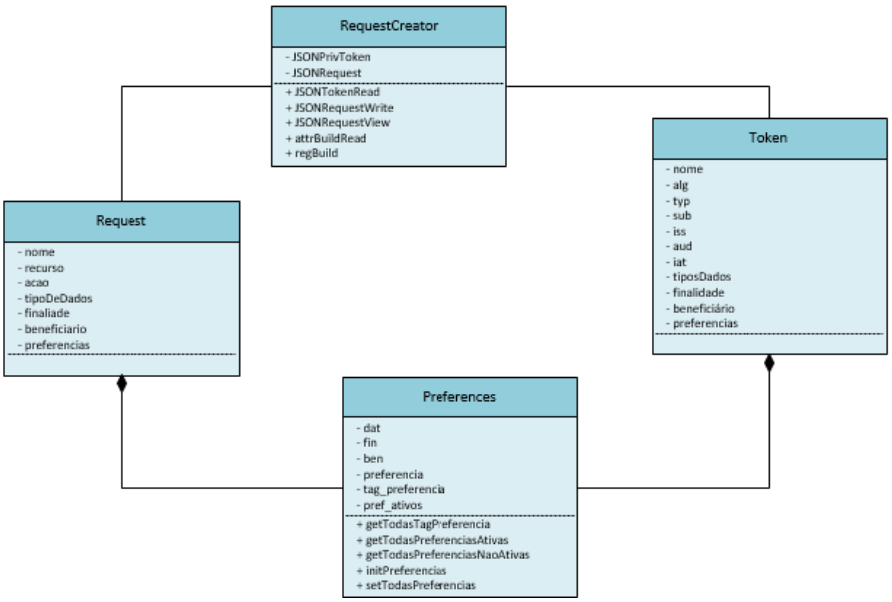
Para poder gerar essa requisição foi criada uma interface como mostra a Figura 12.

Figura 12 - Interface do *Request Creator*



Todos os dados com exceção do nome do arquivo são fundamentais para a definição da requisição. É preciso definir a ação e o recurso através de seus respectivos campos. Esses campos futuramente serão os responsáveis por encontrar as políticas comparando-se o *target* existente. Os outros 3 dados são semelhantes ao do *Privacy Token Creator*

Figura 13 - Classes do *Request Creator*



A classe *Token* possui os atributos do *token* de privacidade, preferências com cada um de seus *getters* e *setters* e atributos. Nesta aplicação a classe *Request* é o objeto que será gerado em um arquivo JSON pelo *Request Creator*. O *Request* possui os campos que cadastraremos na interface da Figura 16 e um de seus atributos também é a lista de preferências, que estão no *Token*. As preferências dessa vez usarão o método que coleta os dados ativos ou não ativos da lista. Esses dados ativos coletados pelo método *getTodasPreferenciasAtivas*, é uma comparação dos valores que o usuário selecionou com o valor "1" no *array* de valores da classe de Preferências e para cada “um” encontrado, a *tag* relacionada ao valor encontrado é armazenada no *array* pref_ativos do *Request*, ou seja, possuirá todos os valores a serem gerados para o *Request*. Por fim a classe *RequestCreator*, ela é a responsável por ler um objeto *Token* através do *JSONTokenRead* e por gerar com o *JSONRequestWrite* e apresentar a nova requisição com o *JSONRequestView*. Esses são os principais métodos criados nesta classe. O resultado gerado é a requisição em *JSON* conforme mostrado na Figura 14.

Figura 14 - Resultado gerado pelo *Request Creator*


```
"Request" : {
  "AccessSubject" : {
          "Attribute" : [
              {
                  "AttributeId" : "subject:subject-id",
                  "DataType" : "string",
                  "Value" : "PP"
              },
              {
                  "AttributeId" : "subject:preferences",
                  "DataType" : "string",
                  "Value" : "pi_sc_pp"
              },
              {
                  "AttributeId" : "subject:preferences",
                  "DataType" : "string",
                  "Value" : "ah_sc_pp"
              }  ]
   },"Action" : {
          "Attribute" : {
              "AttributeId" : "action:action-id",
              "DataType" : "string",
              "Value" : "comprar"
          }
   },"Resource" : {
          "Attribute" : [{

              "AttributeId" : "resource:resource-id",
              "DataType" : "string",
              "Value" : "www.sitedecompras.com:compras:produtoid"
          },
          {
              "AttributeId" : "resource:finalidade",
              "DataType" : "string",
              "Value" : "SC"
          },
          {
              "AttributeId" : "resource:tipo-dado",
              "DataType" : "string",
              "Value" : "PI"
          }]
      }
}
```


Na parte da requisição que lida com dados do sujeito foi definido o beneficiário representado por *subject:subject-id*, e uma forma encontrada para gerar uma *bag*, conjunto de dados de mesmo tipo, no JSON que é utilizar o mesmo id para todos os valores, no caso o *subject:preferences*. No documento de especificação do JSON para requisições oferecido pela OASIS (OASIS JSON, 2011) e nos exemplos existentes nos exemplos do OpenAZ (INCUBATOR OPENAZ, 2017) não foram encontrados exemplos de requisição em JSON e nenhuma maneira de enviar uma estrutura *bag* por ele, pois o tipo *bag* não foi definido neste padrão, portanto a princípio, foi uma dificuldade.

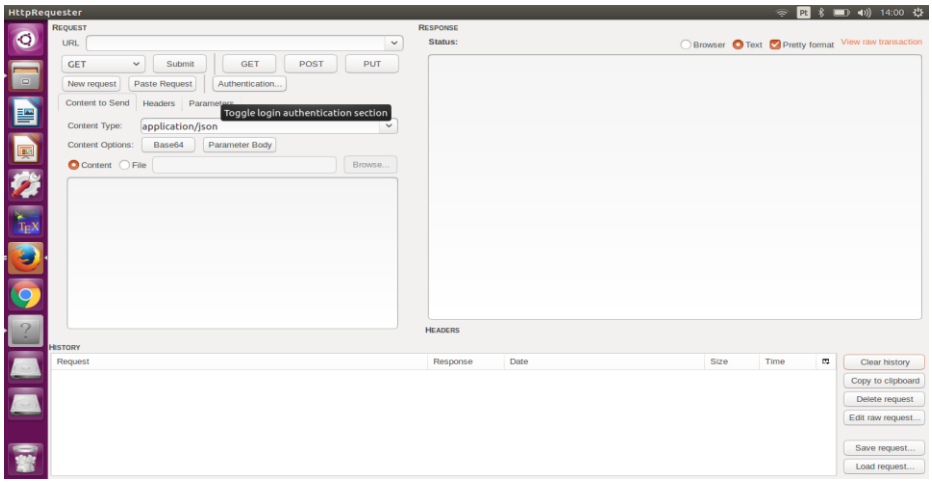
Na parte que lida com a *action* é enviada a ação realizada pelo usuário, uma compra, coleta, visualização, entre outros, representada na estrutura por *action:action-id*. E por fim na parte que lida com os recursos, são definidos o recurso a ser acessado, a finalidade de uso e os tipos de dados representados respectivamente por *resource:resource-id*, *resource:finalidade* e *resource:tipo-dado*. Os tipos de dados também podem ser representados como uma *bag*, da

mesma maneira realizada para as preferências do *token*. Com a requisição formada, um arquivo é gerado com o nome dado pelo usuário e este arquivo deve ser enviado agora para obter o acesso desejado.

3.3.3 Instalação e Configurações do HTTP Request

Esta ferramenta é na realidade um *plugin* utilizado no navegador Mozzila Firefox. É possível gerar uma requisição em um determinado formato, para fazer alguma função HTTP como *GET*, *PUT*, *POST* etc. A Figura 15 permite acompanhar seu uso.

Figura 15 - Interface do *Http Requester*.



Neste trabalho utilizamos o *POST*, pois é mais seguro e porque foi o que melhor funcionou com a ferramenta OpenAZ. Outro elemento importante é o endereço para onde o *request* é enviado. No caso teste a URL é *http://localhost:8775/pdp/* onde o PDP está executando e que será o responsável por trazer o resultado da requisição enviada de acordo com as regras definidas nas políticas. O formato está em *Content Type*: o formato JSON foi usado. A requisição pode ser escrita na tela ou capturada na máquina selecionando a opção *file*, desta maneira foi possível utilizar a requisição gerada pelo *Request Creator*.

.

3.3.4 Criando Políticas de Privacidade com XACML

Para entendermos esta etapa é preciso entender que a privacidade é tratada com a ajuda do *token* de privacidade. Este *token* define o uso dos dados do usuário como "***Quem terá o***

benefício *do recurso para realizar uma ação com* ***qual finalidade*** *utilizando que* ***tipos de dados***?". E essa pergunta define a privacidade o numa política de controle de acesso. Assim, foi definida a política de privacidade, compostas por 4 regras principais:

1. **Regra do PrivacyToken:** Trata os valores vindos da requisição extraídas do *token* de privacidade em comparação com as regras de privacidade. O valor retornado é o *tag* da tripla Ex: *pi_co_pp*. No *token* precisa estar habilitado o consentimento do usuário para poder ser comparado dentro da política com a regra definida, como é o caso de *pi_co_pp=1*.
2. **Regra do Beneficiário:** Trata aquele que é beneficiado pelo uso do recurso, podendo nesse caso ser SP, PP ou TP, que representam o provedor de serviços (IdP), o próprio usuário e uma terceira pessoa respectivamente. Essas informações são definidas pela aplicação para realizar a comparação e não pode usar a função *AND* na definição da política, pois o usuário TP é a terceira parte, qualquer um que não seja o provedor de serviços ou o dono do dado.
3. **Regra da Finalidade:** Para a finalidade de uso do recurso, se pode ter os valores CO, SC e SI. Essas informações também devem ser definidas pela aplicação. O CO representa as finalidades comerciais, ou seja, comércio, estatísticas de mercado para sua empresa etc. SC é para fim científico, educacional entre outros. E por fim, melhoria de serviços representada por SI, que pode abranger coletas de dados para estatística de melhoria do serviço utilizado.
4. **Regra do Tipo de dados:** Para o tipo de dado a ser usado é um agrupamento de dados como PI, PCP, LO, AH e RS. Esse grupo deve ser definido pela aplicação ou por um IdP, portanto neste caso será definido na requisição. Esse pode utilizar funções *AND* e *OR* de acordo com os requisitos a serem utilizados. PI são os dados pessoais, como nome, identificadores nacionais, nomes dos pais, endereço residencial, número do cartão de crédito, qualquer dado que identifique o sujeito. PCP são características pessoais e preferências do sujeito. Os dados vinculados a ele poderiam ser algo como peso, crenças religiosas, orientação sexual algo opcional para o sujeito ou que seja uma característica que não seja algo que o represente. LO são informações de localização oriundas de trajetórias ou localidade; dado de forma mais precisa por um dispositivo como GPS, IP, WIFI, etc. AH são atividades e hábitos de um sujeito, como sites visitados, compras, histórico de busca entre outros. E por fim RS que são relações ou relacionamentos, como cônjuge, amigos em lista de e-mail, redes sociais, compradores de um site e por ai vai.

Figura 16 - Condição de uma regra de permissão da politica de privacidade.

```
<Target/>
<Condition >
    <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:or">
        <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:and">
            <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:any-of-any"> <!-- pi_co_pp -->

            <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:any-of-any">  <!-- ah_co_pp -->

            <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:any-of-any">  <!-- rs_co_pp -->
                <Function FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-equal"/>

                <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-bag">
                    <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">rs_co_pp</AttributeValue>
                </Apply>
                <AttributeDesignator DataType="http://www.w3.org/2001/XMLSchema#string"
                                     MustBePresent="false"
                                     Category="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject"
                                     AttributeId="subject:preferences"/>
            </Apply>
            <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:any-of-any"><!-- PP -->
        </Apply>
        <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:and">
            <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:any-of-any">  <!-- pi_co_sp -->

            <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:any-of-any">   <!-- ah_co_sp -->

            <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:any-of-any">  <!-- rs_co_sp -->
                <Function FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-equal"/>
                <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-bag">
                    <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">rs_co_sp</AttributeValue>
                </Apply>
                <AttributeDesignator DataType="http://www.w3.org/2001/XMLSchema#string"
                                     MustBePresent="false"
                                     Category="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject"
                                     AttributeId="subject:preferences"/>
            </Apply>
            <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:any-of-any"><!-- SP -->
        </Apply>
    </Apply>
```

Para obter uma resposta é feita uma comparação simples de igualdade. Podem existir mais de um dado a ser comparado e podem ser agrupados usando função *AND* ou *OR* de acordo os privilégios para a política. Neste caso da Figura 16 consegue-se visualizar que a função *any-of-any* é usada para comparar a igualdade da *string* PI com cada elemento que poderá observar os dados vindos do *resources:tipo-dado*, oriundos da requisição enviada. Se ao menos um elemento desta *bag* for este dado ele retorna o valor *True*.

As políticas de privacidade foram criadas manualmente, utilizando especificações da OASIS e exemplos de política de controle de acesso para entendimento dos funcionamentos das funções. Uma grande dificuldade encontrada foi a existência de maneiras diferentes de se escrever a política, e pela pouca documentação encontrada sobre o OpenAZ, foi necessário um estudo sobre escritas de políticas no OpenAZ. Na Figura 17 há um exemplo de como é realizada a montagem de cada elemento do alvo nas políticas usadas nesse projeto.

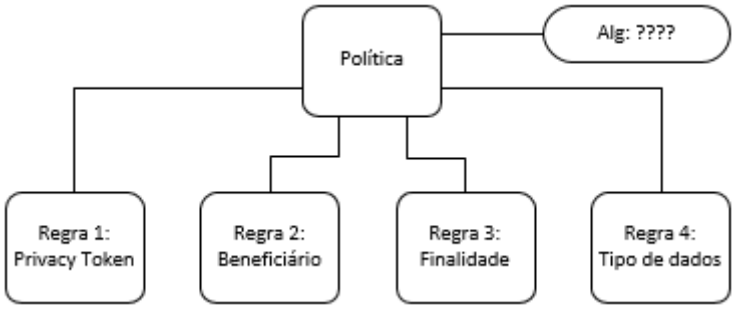
Figura 17 - Formato de um elemento usado no *target* das políticas

```
<AnyOf>
    <AllOf>
        <Match MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
            <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">comprar</AttributeValue>
            <AttributeDesignator MustBePresent="true"
                                 Category="urn:oasis:names:tc:xacml:3.0:attribute-category:action"
                                 AttributeId="action:action-id"
                                 DataType="http://www.w3.org/2001/XMLSchema#string"/>
        </Match>
    </AllOf>
</AnyOf>
```

O *Target* da política é a forma como o OpenAZ entende, pois o símbolos *Subjects*, *Actions*, *Resources* do XACML demonstrados no capítulo sobre controle de acesso não existem na implementação do OpenAZ. O formato do *target* foi adaptado para um formato que o OpenAZ pudesse entender, já que o modelo de *target* tradicional não pode ser usado por não possuir os símbolos equivalentes para sua compilação e interpretação sintática. Apenas observando o código do OpenAZ foi possível perceber este problema. Portanto foi declarada a categoria para *action* fazendo uma combinação usando a comparação de igualdade na *tag* <*Match*>.

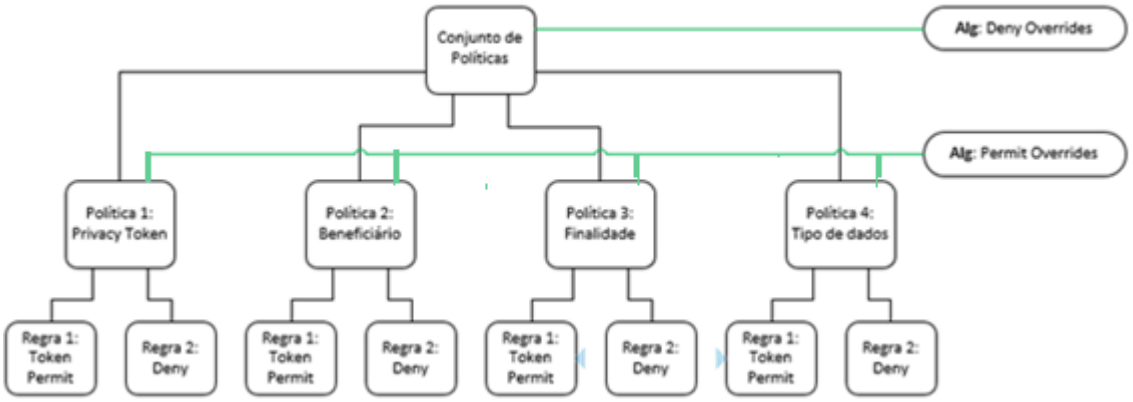
Para criar uma política que lide com a privacidade precisa-se de uma estrutura bem definida para tratá-las. Uma dificuldade encontrada foi em relação aos algoritmos: precisamos que todas as regras sejam examinadas para que a permissão seja concedida mas não existe nenhum algoritmo existente que realiza isso. Portanto, a política de privacidade deste modelo não pôde ser criada usando uma estrutura de política e regras como mostra a Figura 18 sem a criação de um novo algoritmo.

Figura 18 - Primeiro formato da politica de privacidade



A primeira idéia foi criar um algoritmo, mas alterar o código exigiria mais tempo além de fazer um estudo para a validação deste algoritmo. Um algoritmo foi criado modificando um algoritmo existente, mas, ainda assim, foi considerado mais adequado tentar encontrar um outro modo. O modo encontrado foi adaptar a ideia de uma regra para cada caso a ser analisado para uma política com duas regras para cada caso. Será explicada como foi tratada a política caso observando a Figura 19.

Figura 19 - Formato das políticas de privacidade



Pode-se observar que a politica de privacidade se tornou um conjunto de politicas, e cada política tem um caso a ser tratado. São no total quatro políticas, cada uma para tratar um caso: as regras do *token* de privacidade, as regras do beneficiário, as regras da finalidade e por fim as regras de tipos de dados. Cada política precisa ter duas regras, a primeira é a regra que define se permite ou não o acesso, como definido na Figura 19, enquanto a segunda regra será uma regra de negação por definição, pois não possui nenhuma condição, veja na Figura 20.

Figura 20 - Formato de regras usada nas políticas de privacidade.

```
<Rule RuleId="rule-1:privacytoken-01" Effect="Permit">

    <Condition >
        <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:any-of-any">
            <Function FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-equal"/>

            <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-bag">
                <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">pi_sc_pp</AttributeValue>
            </Apply>
            <AttributeDesignator DataType="http://www.w3.org/2001/XMLSchema#string"
                                 MustBePresent="false"
                                 Category="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject"
                                 AttributeId="subject:preferences"/>


        </Apply>
    </Condition>
</Rule>
<Rule RuleId="rule-2:deny" Effect="Deny"></Rule>
```

Pode-se observar na Figura 19, cada uma das 4 políticas possui um algoritmo de combinação de regras *permit-overrides*, que fornece permissão se ao menos uma das regras retornar *permit*. Caso exista qualquer outro resultado possível como *not applicable* ou *indeterminate* então a primeira regra é ignorada e segunda regra resulta em negação automaticamente e portanto a política retorna *deny*.

O algoritmo de combinação de políticas utilizado é o *deny-overrides*, assim, caso uma das políticas seja de negação, ele retorna *Deny* e o PDP retorna essa mensagem para o PEP ou no caso a aplicação *HTTPRequester*, com o *Status* e a *Decision Deny* caso uma das políticas seja *Deny*. Caso todas as politicas sejam *Permit*, então a permissão é concedida.

3.3.5 Configurações do OpenAZ

Com as políticas já configuradas, e as requisições formadas é possível enviar as políticas para o servidor que fará a decisão de acesso ao recurso. Para isso é necessário levar em consideração algumas configurações e entender algumas limitações. Primeiramente é preciso baixar o pacote com os códigos que o OpenAZ oferece no site (INCUBATOR OPENAZ, 2017). Neste pacote o único código que precisa-se utilizar neste trabalho é o *openaz-xacml-pdp* que possui os códigos responsáveis para o funcionamento e *openaz-xacml-pdp-rest*.

No Windows houve dificuldades para executar essa ferramenta por alguns motivos: muitas dependências não conseguem ser carregadas e isso causa a necessidade de abrir o arquivo de dependências do Maven, o pom.xml para incluir dependências mais atualizadas. Cada vez que for necessário instalar alguma dependência pode-se instalar diretamente no *Help > Install New Software*, indicando o caminho para instalar o que faltar.

Para executar o projeto *REST* do PDP, foi utilizado o Jetty, que foi baixado através da instalação de *plugins* do eclipse na aba Help (JETTY, 2017).

Após a instalação se concretizar, antes de configurar o Jetty e o executar é necessário fazer algumas configurações para o OpenAZ. A Figura 21 representa o arquivo *xacml.pdp.config*, localizado no diretório *openaz-xacml-pdp-rest*.

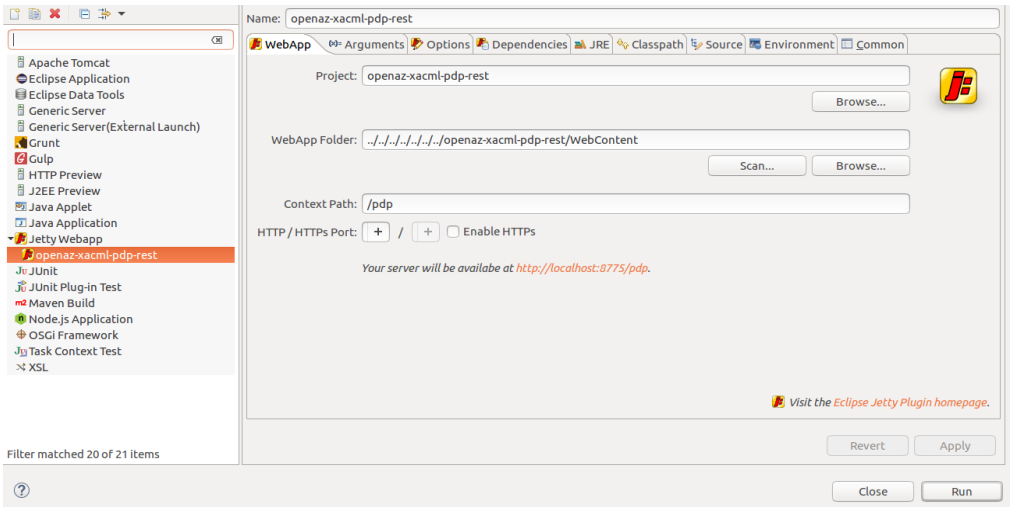
Figura 21 - Configurações do PDP com suas políticas

```
xacml.rest.pdp.id=http://localhost:8775/pdp/
xacml.rest.pdp.config=/opt/app/xacml/config
xacml.rest.pdp.id.description=PDP TCC
xacml.rest.pdp.id.name=PDPdoTCC

# ==================      POLÍTICAS   ======================
xacml.rootPolicies= politica_caso_1.xml, politica_caso_2.xml, politica_caso_3.xml
xacml.referencedPolicies=  politica_caso_1.xml, politica_caso_2.xml, politica_caso_3.xml

politica_caso_1.xml.name=politica_caso_1.xml
politica_caso_1.xml.file=/opt/app/xacml/policies/default/politica_caso_1.xml

politica_caso_2.xml.name=politica_caso_2.xml
politica_caso_2.xml.file=/opt/app/xacml/policies/default/politica_caso_2.xml

politica_caso_3.xml.name=politica_caso_3.xml
politica_caso_3.xml.file=/opt/app/xacml/policies/default/politica_caso_3.xml
```

Neste arquivo configura-se na linha *xacml.rest.pdp*.id é o local onde o PDP vai executar, ou seja, ele precisa ligar o código executado ao servidor. Depois é necessário definir o caminho das políticas. Esses caminhos e configurações de localização e nome das políticas são apresentados nesse documento e em outros 2 documentos em diretórios *config*, um localizado no diretório *config* do *openaz-xacml-pdp-rest* e outro no diretório que você descreveu no *xacml.rest.pdp.config* dado na linha 2 da Figura 21. Nesse diretório costuma ficar essa configuração. Além disso, também é possível mudar a descrição e o nome do projeto, mas é opcional. Por fim, precisa-se passar o nome das políticas no *xacml.rootPolicies*.

Com o OpenAZ configurado e o Jetty instalado é possível configurá-lo clicando em *Run configuration* e abre uma janela na qual pode-se selecionar o Jetty. A Figura 22 mostra os campos que precisam ser preenchidos.

Figura 22 - Configuração do Jetty.



É necessário escrever a porta que será usada para executar a aplicação, no caso o *REST* do PDP, no nosso caso vai executar na porta 8775. Também se precisa colocar do caminho do *Web content* do projeto no *WebApp* Folder, o projeto que será executado e por fim o que estaremos executando no campo *Context Path*. Após todas essas configurações é possível executar o Jetty para subir um servidor que receberá as requisições do HTTP e depois disso já estará pronto para realizar o controle de autorização para testar as políticas.

Após seguir todos os passos e ter todas as ferramentas devidamente configuradas e instaladas, podem-se seguir os passos descritos no começo deste capítulo para a execução dos testes. Esses testes serão demonstrados a seguir no capítulo de resultados com a ajuda dos casos de usos criados para cada política desenvolvida.

4. RESULTADOS EXPERIMENTAIS

Existem para fins de teste 3 políticas, uma para cada caso de uso. O *token* de privacidade utilizado será apenas um, considerando que o alvo sempre seja o mesmo. Como sabemos, as preferências são para cada usuário e não necessariamente para todos os sistemas, pois para cada sistema diferente, podemos ter privilégios e usos diferentes dos dados.

Para os casos de uso, vale ressaltar que algumas configurações já estarão estruturadas no ambiente:

- Para os testes dos casos de uso o OpenAZ será executado na porta 8775 para decisão do PDP.
- As 3 políticas foram todas criadas e estão em um diretório local para serem usadas para o controle de acesso.
- Os *tokens* de privacidade do usuário já deverá ter sido criado para ser usado pelo *RequestCreator*.

O recurso é baseado em um cenário do mundo real apenas para melhor compreensão. Os recursos seriam acessados através de um site, neste caso fictício *www.sitedecompras.com* e as ações que geram requisições no site serão representadas pelo *HTTP Requester*.

As três políticas criadas anteriormente serão explicadas ao longo deste capítulo e são aplicadas de acordo com as escolhas do usuário na aplicação em conjunto com a requisição gerada pelo *RequestCreator* juntamente ao *Privacy Token* selecionado.

Pode-se observar cada requisito das três politicas para cada caso de forma resumida na Tabela 2 - Essa tabela será melhor explicada ao longo do capítulo.

Tabela 2 - Dados de permissão nas politicas de privacidade.

DADOS DE PERMISSÃO NAS POLÍTICAS						
	Ação	**Recurso**	**Tipos de Dados**	**Finalidade**	**Beneficiário**	**Operação**
Caso de uso 1	comprar	produtoid	PI	SC	PP	**Tudo AND**
Caso de uso 2	visualizar	histórico	PI, AH	SI	SP	**Tudo AND**
Caso de uso 3	coletar	qtde_compradores	PI, AH, RS	CO	PP,SP	**Bem: OR**

Os passos realizados são os seguintes para todos os casos:

Passo 1: Precisa-se criar uma requisição para tentar obter acesso ao sistema, utilizando o *Request Creator*. A configuração da requisição criada pelo *RequestCreator* é exatamente como descrito nos casos de uso das seguintes tabelas: Tabela 3, Tabela 4 e Tabela 5.

Passo 2: A requisição é enviada para o serviço do PDP na porta 8775 através do HTTP *Requester*.

Passo 3: O PDP localizará uma política aplicável com uma comparação do *target* e o *request* enviado tomará a decisão de acordo com as regras das políticas.

Passo 4: PDP retorna resposta para o usuário no *HTTP Requester*.

Para que seja concedida a permissão, todas as políticas precisam ter a decisão *permit* para que o conjunto de políticas de privacidade conceda o controle de acesso. Caso contrário a autorização será negada (*deny*).

Na Figura 23 pode-se ver parte de dois *tokens*, o *token* 1 e o *token* 2, cada um com suas preferências que serão usadas nos testes.

Figura 23 - *Tokens* com preferências de privacidade distintas

Token 1

```
preferencias:{
  PI_SI_PP: 0
  PI_SI_SP: 0
  PI_SI_TP: 0
  PI_SC_PP: 0
  PI_SC_SP: 0
  PI_SC_TP: 0
  PI_CO_PP: 0
  PI_CO_SP: 0
  PI_CO_TP: 0
  PCP_SI_PP: 0
  PCP_SI_SP: 0
  PCP_SI_TP: 0
  PCP_SC_PP: 0
  PCP_SC_SP: 0
  PCP_SC_TP: 0
  PCP_CO_PP: 0
  PCP_CO_SP: 0
  PCP_CO_TP: 0
  LO_SI_PP: 0
  LO_SI_SP: 0
  LO_SI_TP: 0
  LO_SC_PP: 0
  LO_SC_SP: 0
  LO_SC_TP: 0
  LO_CO_PP: 0
  LO_CO_SP: 0
  LO_CO_TP: 0
  AH_SI_PP: 0
  AH_SI_SP: 1
  AH_SI_TP: 1
  AH_SC_PP: 0
  AH_SC_SP: 0
  AH_SC_TP: 0
  AH_CO_PP: 0
  AH_CO_SP: 1
  AH_CO_TP: 1
  RS_SI_PP: 0
  RS_SI_SP: 1
  RS_SI_TP: 1
  RS_SC_PP: 0
  RS_SC_SP: 0
  RS_SC_TP: 0
  RS_CO_PP: 0
  RS_CO_SP: 1
  RS_CO_TP: 1
```

Token 2

```
preferencias:{
  PI_SI_PP: 1
  PI_SI_SP: 1
  PI_SI_TP: 1
  PI_SC_PP: 1
  PI_SC_SP: 1
  PI_SC_TP: 1
  PI_CO_PP: 1
  PI_CO_SP: 1
  PI_CO_TP: 1
  PCP_SI_PP: 0
  PCP_SI_SP: 0
  PCP_SI_TP: 0
  PCP_SC_PP: 0
  PCP_SC_SP: 0
  PCP_SC_TP: 0
  PCP_CO_PP: 0
  PCP_CO_SP: 0
  PCP_CO_TP: 0
  LO_SI_PP: 0
  LO_SI_SP: 0
  LO_SI_TP: 0
  LO_SC_PP: 0
  LO_SC_SP: 0
  LO_SC_TP: 0
  LO_CO_PP: 0
  LO_CO_SP: 0
  LO_CO_TP: 0
  AH_SI_PP: 1
  AH_SI_SP: 1
  AH_SI_TP: 1
  AH_SC_PP: 1
  AH_SC_SP: 1
  AH_SC_TP: 1
  AH_CO_PP: 1
  AH_CO_SP: 1
  AH_CO_TP: 1
  RS_SI_PP: 1
  RS_SI_SP: 1
  RS_SI_TP: 1
  RS_SC_PP: 1
  RS_SC_SP: 1
  RS_SC_TP: 1
  RS_CO_PP: 1
  RS_CO_SP: 1
  RS_CO_TP: 1
```

O *Token* 1 é um subconjunto do *Token* 2 e portanto fornece menos privilégios a quem for tentar algum acesso. O *Token* 1 será usado para os casos de negação e o *Token* 2 para os casos de permissão. A estrutura das requisições foi fixada para facilitar os testes.

Os detalhes cada caso de uso com as determinadas requisições e a aplicação dos privilégios do usuário para cada ação desejada a cada recurso, são explicados nas próximas seções.

4.1 CASO DE USO 1

Para este caso de uso existe uma política criada manualmente para este trabalho, representada de forma mais clara pela política 1 da Tabela 2, e duas requisições que foram geradas pelo algoritmo *RequestCreator* e representadas na tabela 3 para melhor compreensão. Uma das requisições possui um caso de acesso permitido e outra de acesso negado.

A ação pretendida é a tentativa de um determinado usuário realizar uma compra de um software em um site de compras *www.sitedecompras.com/compras/produtoid.*

Pode-se observar no caso 1 da Tabela 2 que representa a política *politica_caso_1.xml* a qual as requisições deste caso de uso buscam acessar através do *target* vinculado a política para localizá-la. O PDP usa a requisição e o *target* da política contendo recurso que se deseja acessar e a ação que se deseja fazer com este recurso. Neste caso, comprar é a ação e o recurso é o produto em questão.

Tabela 3 - Requisições do caso de uso 1.

REQUISIÇÕES DO CASO DE USO 1						
Requisição	**Tipos de dados**	**Finalidade**	**Beneficiário**	**Token Válido?**	**Target**	**Status**
1	PI	SC	PP	SIM	OK	**PERMIT**
2	PI	SC	PP	NÃO	OK	**DENY**

Também é possível observar na Figura 24 um trecho da política que trata da comparação dos valores das requisições relativas ao *token* de privacidade.

Figura 24 - Trecho da política do caso de uso 1.

```
<Rule RuleId="rule-1:privacytoken-01" Effect="Permit">

    <Condition >
        <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:any-of-any">
            <Function FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-equal"/>

            <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-bag">
                <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">pi_sc_pp</AttributeValue>
            </Apply>
            <AttributeDesignator DataType="http://www.w3.org/2001/XMLSchema#string"
                                 MustBePresent="false"
                                 Category="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject"
                                 AttributeId="subject:preferences"/>

        </Apply>
    </Condition>
```

Para a regra receber o valor *permit* da política que trata do *token*, o dado contido na requisição vindo do *subject:preferences*, precisa estar presente na política. Vamos supor que o dado na requisição contém com a tripla *pi_sc_pp*, então na política esta mesma tripla precisa estar presente .Caso contrário a regra 2 que é uma regra de negação direta retornará *deny* e não dará a permissão de acesso.

Agora que já conseguimos entender como a politica_caso_1.xml se comporta podemos analisar os casos de permissão e negação de acesso relacionados a privacidade de acordo com cada requisição enviada e cada *token* especificado.

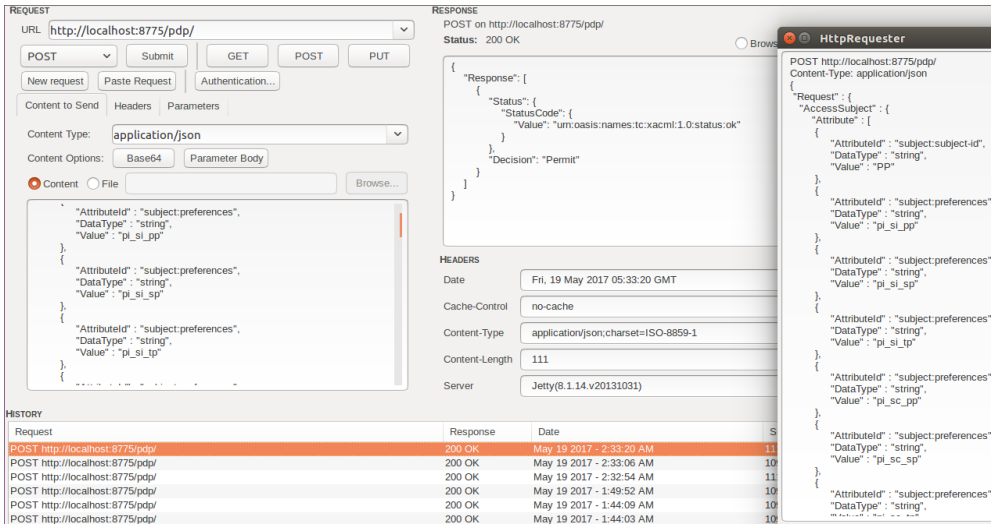
4.1.1 Caso de Permissão Concedida

A requisição usada neste exemplo é a requisição 1 da Tabela 3. O usuário dono do recurso (PP) tenta realizar uma compra de um software (produto) com a finalidade educacional (SC). Os dados utilizados são de informação PI (cpf, n_da conta, endereço ect). O *token* 2 da Figura 23 está anexado na requisição.

Pode-se observar que todos os requisitos necessários para o acesso estão coerentes com os dados da política (beneficiário, finalidade e tipos de dados) inclusive o *token* de privacidade está com as preferências coerentes com as da política. Na comparação da regra do *token* contida na política representada pela Figura 24 com a regra contida no *token* de privacidade do usuário representada pela Figura 23 é possível observar que todos os dados exigidos na política estão presentes na requisição, considerando as preferências do Token 2, que é o exemplo de t*oken* com mais privilégios.

Através na tabela 2 e tabela 3 é possível ver de forma mais clara a inconsistência dos dados da política 1 com a requisição 1.

Figura 25 - Resposta de permissão após o envio da requisição.



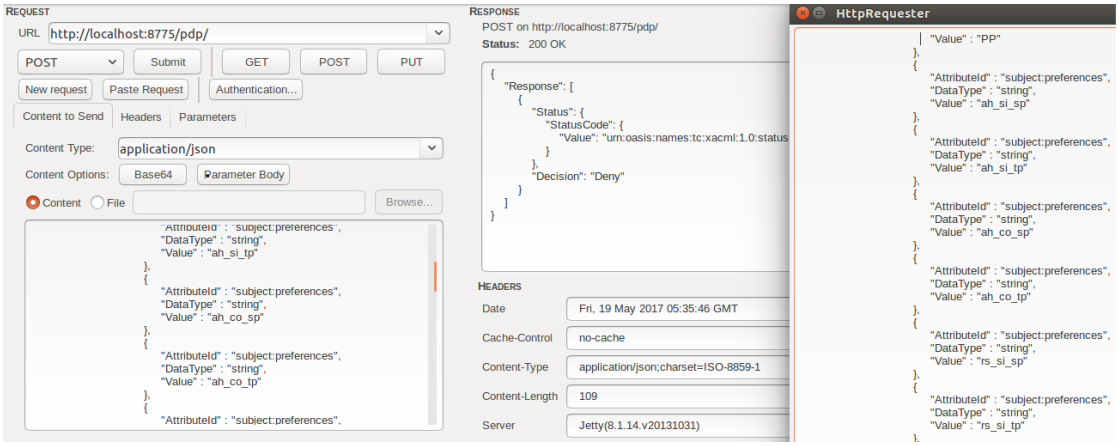
Por fim, a figura 25 ilustra a resposta padrão para o acesso permitido é *Status*: Ok e a *Decision: Permit* após a análise realizada pelo PDP.

4.1.2 Caso de Permissão Negada

A requisição usada neste exemplo é a requisição 2 da Tabela 3. O usuário tenta realizar uma compra de um software (produto) com a finalidade educacional. Os dados utilizados são de informações pessoal (PI) (cpf, n_da conta, endereço ect). O *token* 1 da Figura27 está anexado na requisição.

Podemos observar que todos os requisitos necessários para o acesso estão coerentes com os dados que a política. Porém existe um caso no qual não passa o acesso. O *token* de privacidade não foi habilitado para permitir o acesso daquele beneficiário para realizar aquela ação e, portanto sua permissão é negada a fim de proteger a privacidade do proprietário do recurso. Na requisição a tripla solicitada pela politica representada pela figura 26 não existe no *token 1* ilustrada na figura 23. Logo não está presente na requisição enviada pelo usuário. Através na tabela 2 e tabela 3 é possível ver de forma mais clara a inconsistência dos dados da política 1 com a requisição 2.

Figura 26 - Resposta de permissão após o envio da requisição.



A Figura 26 mostra a resposta padrão quando o acesso ao recurso é negado. O *Status*: *Ok* e a *Decision*: *Deny*. As demais regras das outras políticas nem precisam ser comparados quando este *token* é negado.

4.2 CASO DE USO 2

Para este caso de uso existe uma política criada manualmente para este trabalho, representada de forma mais clara pela política 2 da Tabela 2, e duas requisições que foram geradas pelo algoritmo *RequestCreator* e representadas na Tabela 4 para melhor compreensão. Uma das requisições possui um caso de acesso permitido e outra de acesso negado.

A ação pretendida é tentativa de um usuário de visualizar o histórico de um usuário de um site de compras *www.sitedecompras.com/userid/historico* para melhoria de serviços.

Pode-se observar na Tabela 4, o caso 2 representa a política politica_caso_2.xml na qual as requisições deste caso de uso buscam acessar através do alvo para localizá-la. O PDP usa a requisição e o *target* da política contendo o recurso que deseja-se acessar e a ação que deseja-se fazer com este recurso, neste caso, visualizar é a ação e o recurso é o histórico do usuário em questão.

Tabela 4 - Requisições do caso de uso 2.

REQUISIÇÕES DO CASO DE USO 2						
Requisição	**Tipos de dados**	**Finalidade**	**Beneficiário**	**Token Válido?**	**Target**	**Status**
1	PI, AH	SI	PP	SIM	OK	**PERMIT**
2	PI, AH	SI	PP	NÃO	OK	**DENY**

Também é possível observar na Figura 27 um trecho da política que trata da comparação dos valores das requisições relativas ao *token* de privacidade.

Figura 27 - Trecho da política do caso de uso 1.

```
<Condition >
    <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:and">
        <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:any-of-any">
            <Function FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-equal"/>

            <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-bag">
                <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">pi_si_sp</AttributeValue>
            </Apply>
            <AttributeDesignator DataType="http://www.w3.org/2001/XMLSchema#string"
                                 MustBePresent="false"
                                 Category="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject"
                                 AttributeId="subject:preferences"/>
        </Apply>

        <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:any-of-any">
            <Function FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-equal"/>

            <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-bag">
                <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">ah_si_sp</AttributeValue>
            </Apply>
            <AttributeDesignator DataType="http://www.w3.org/2001/XMLSchema#string"
                                 MustBePresent="false"
                                 Category="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject"
                                 AttributeId="subject:preferences"/>
        </Apply>
    </Apply>
```

Para a regra receber *Permit* dentro da política que trata do *token*, o dado da requisição vindo do *subject:preferences*, precisa conter o dado presente na política, para cada elemento relacionado a preferência. Caso contrário a regra 2 que é uma regra de negação direta retornará *deny*.

Agora que já conseguimos entender como a politica_caso_1.xml se comporta podemos analisar os casos de permissão e negação de acesso relacionados a privacidade de acordo com cada requisição enviada e cada *token* especificado.

4.2.1 Caso de Permissão Concedida

A requisição usada neste exemplo é a requisição 1 da Tabela 4. O provedor de serviços deseja visualizar o histórico de um usuário de um site de compras *www.sitedecompras.com:userid:historico* para melhoria de serviços. O *token* 2 da Figura1 está anexado na requisição.

Pode-se observar que todos os requisitos necessários para o acesso está coerente com os dados da política. Também tendo em vista a Figura 23 que representa em o token de privacidade do usuário, o *token* 2 tem as preferências habilitadas para a ação desse recurso com o consentimento do usuário em concordância com a política de privacidade. Logo todos os dados necessários para o acesso do usuário solicitados pela política da figura 27 estão presentes na requisição, permitindo assim o acesso ao recursos.

4.2.2 Caso de Permissão Negada

A requisição usada neste exemplo é a requisição 2 da Tabela 4.

O provedor de serviços deseja visualizar o histórico de um usuário de um site de compras *www.sitedecompras.com:userid:historico* para melhoria de serviços. O *token* 1 da Figura 27 está anexado na requisição.

Pode-se observar que todos os requisitos necessários para o acesso estão coerentes com os dados que a política. Porém existe um caso no qual o acesso não é concedido, no caso o *token* de privacidade não foi habilitado para permitir o acesso do beneficiário para realizar aquela ação e portanto sua permissão é negada a fim de proteger a privacidade do proprietário do recurso.

4.3 CASO DE USO 3

Para este caso de uso existe uma política criada manualmente para este trabalho, representada de forma mais clara pela política 3 da Tabela 2, e três requisições que foram geradas pelo algoritmo *RequestCreator* e representadas na Tabela 5 para melhor compreensão. Duas das requisições possuem um caso de acesso permitido e outra de acesso negado.

A ação pretendida é a tentativa de um usuário realizar uma coleta de informações sobre o número de compradores no histórico de compras de um determinado usuário com finalidade comercial. O recurso se encontra em *www.sitedecompras.com:userid:historico:qtde_compradores.*

Pode-se observar na Tabela 2, o caso 1 representa a política politica_caso_3.xml na qual as requisições deste caso de uso buscam acessar através do alvo para localizá-la. O PDP usa a requisição e o *target* da política contendo recurso que deseja-se acessar e a ação que deseja-se fazer com este recurso, neste caso, coletar é a ação e o recurso é o número de compradores no histórico de compras de um usuário em questão.

Tabela 5- Requisições do caso de uso 3

REQUISIÇÕES DO CASO DE USO 3						
Requisição	**Tipos de dados**	**Finalidade**	**Beneficiário**	**Token Válido?**	**Target**	**Status**
1.a	PI, AH,RS	CO	PP	SIM	OK	**PERMIT**
1.b	PI, AH,RS	CO	SP	SIM	OK	**PERMIT**
2	PI, AH,RS	CO	PP	NÃO	OK	**DENY**

Também é possível observar na Figura 28 um trecho da política que trata da comparação dos valores da requisição relativa ao *token* de privacidade.

Figura 28 - Trecho da política do caso de uso 3.

```
<Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:or">
    <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:and">
        <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:any-of-any">
            <Function FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-equal"/>

            <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-bag"> <!-- pi_co_pp -->
                <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">pi_co_pp</AttributeValue>
            </Apply>
            <AttributeDesignator DataType="http://www.w3.org/2001/XMLSchema#string"
                                 MustBePresent="false"
                                 Category="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject"
                                 AttributeId="subject:preferences"/>
        </Apply>
        <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:any-of-any">  <!-- ah_co_pp -->
        <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:any-of-any">  <!-- rs_co_pp -->
        <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:any-of-any">  <!-- regra só se aplica se for para PP -->
            <Function FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-equal"/>

            <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-bag">
                <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">PP</AttributeValue>
            </Apply>
            <AttributeDesignator DataType="http://www.w3.org/2001/XMLSchema#string"
                        MustBePresent="false"
                        Category="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject"
                         AttributeId="subject:subject-id"/>

        </Apply>

</Apply>
    <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:and">
        <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:any-of-any">  <!-- pi_co_sp -->
        <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:any-of-any">  <!-- ah_co_sp -->
            <Function FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-equal"/>

            <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-bag">
                <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">ah_co_sp</AttributeValue>
            </Apply>
            <AttributeDesignator DataType="http://www.w3.org/2001/XMLSchema#string"
                         MustBePresent="false"
                         Category="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject"
```

Para a regra receber *permit* dentro da política que trata do *token*, o dado da requisição vindo do *subject:preferences*, precisa conter os dados presentes na política para cada tipo de dado solicitado. Além disso, o usuário beneficiário precisa ser avaliado junto ao *token* além da política que trata do beneficiário, para que o acesso seja coerente, ou seja, o *subject:subject-id* vinda da requisição precisa ser o usuário estipulado na regra. Caso isso não seja feito, mesmo que os dados não tenham permissão para um determinado usuário, ele pode vir a ter acesso permitido. Essa exceção acontece, pois agora o privilégio foi dado a mais de um tipo de usuário.

Caso a regra referente ao *token* não seja obedecida, a regra 2 retornará *deny* para a política do *token*. Analisaremos agora os casos de permissão e o caso de negação para a política do caso de uso 3.

4.3.1 Caso de Permissão Concedida

A requisição usada neste exemplo é a requisição 1.a e 1.b da Tabela 5. O usuário tenta realizar a coleta do número de compradores para fim comercial. No caso o usuário é próprio dono do recurso (PP) na requisição 1.a e o provedor de serviço (SP) na 1.b. O *token* 2 da Figura 23 está anexado a requisição.

Podemos observar que todos os requisitos necessários para o acesso estão coerentes com os dados que a política. Não difere muito dos casos anteriores, a única diferença é que para este caso dois tipos de beneficiário provedor de serviço(SP) ou o próprio usuário (PP) tem o privilégio da ação de coleta para fim comercial do recurso desejado, pois o *token* está habilitado para este fim. O resultado de *Permit* é apresentado da mesma maneira como ilustrado na Figura 25.

4.3.2 Caso de Permissão Negada

A requisição usada neste exemplo é a requisição 2 da Tabela 5. O usuário tenta realizar uma coleta de dados do histórico de compras de um usuário com a finalidade comercial. O *token* 1 da Figura 23 está anexado a requisição.

Pode-se observar que todos os requisitos necessários para o acesso estão coerentes com os dados que a política. Porém existe um caso o qual não passa o acesso, no caso o *token* de privacidade não foi habilitado para permitir o acesso ao beneficiário realizar uma ação e, portanto sua permissão é negada a fim de proteger a privacidade do proprietário do recurso.

Como citado anteriormente, para este caso além das regras de comparação do *token* na política do *token*, é necessário também comparar o beneficiário na mesma política do *token* além da comparação já realizada na política do beneficiário.

4.4 CONSIDERAÇÕES

Foi descrita neste capítulo a aplicação da privacidade com a ajuda do *token* gerado pelo *Privacy Token*. A ideia é que este *token* seja gerado por um gerenciador de identidade como OpenID Connect, através do IdP. Mas foi possível ver que se podem utilizar esses métodos para proteger os dados sensíveis do usuário proprietário do recurso.

5 CONCLUSÕES

Neste trabalho foi realizada a criação de políticas de privacidade utilizando a linguagem de controle de acesso XACML além da criação de algumas ferramentas que ajudaram a tornar o ambiente de testes mais simples de se utilizar como o *Privacy Token Creator* e *Request Creator*. Os estudos das bibliografias coletadas deu base aos resultados positivos dos experimentos realizados neste trabalho.

Foi possível entender durante a concepção desse trabalho como funciona um sistema de controle de acesso e a dificuldade que se pode ter para montar politicas manualmente. Além disso, definir o papel de cada entidade participante do modelo proposto pode ser complicado. Um exemplo poderia ser quem é o responsável pela coleta de informações? Aonde são armazenados os dados de preferência no modelo mantendo a estrutura do sistema de controle de acesso existente? Portanto muitas vezes é possível apenas definir sugestões para o modelo que queremos realizar.

Foi necessário um estudo de um sistema de controle de acesso assim como o entendimento da construção de uma política de controle de acesso para aderir o conceito de privacidade a ela. O estudo envolveu outros trabalhos relacionados onde foi utilizado conceitos de um deles para que fosse possível concluir o trabalho e chegar no resultado.

Também foi necessário desenvolver uma arquitetura que pudesse realizar todos os passos necessários para gerar o resultado, como sistemas que gerassem requisições e o *privacy token* .

5.1 PRINCIPAIS CONTRIBUIÇÕES

As principais contribuições deste trabalho foram:

- Uma implementação de um sistema de requisição que pode ser útil em testes de políticas voltadas a privacidade usando este modelo;
- Modelo de política de privacidade que pode ser usada junto a políticas de controle de acesso convencionais.
- A implementação de um sistema que cria um *Token* de privacidade baseado no modelo de *Privacy Token* de trabalhos relacionados.

5.2 TRABALHOS FUTUROS

Para trabalhos futuros poderiam ser sugeridos uma forma de integração das políticas com um sistema de gerenciamento de identidades, mas especificamente o OpenID Connect, para inserir a essa proposta em um modelo real um IdM.

Um editor para implementação dessas políticas poderia tornar mais simples a criação dessas politicas,

Considerando a ideia de criação de politicas, poderia ser considerado um modelo de processamento de políticas de privacidade e controle de acesso em formato JSON, já que não existe ainda.

Outro estudo que poderia ser interessante é um estudo e o desenvolvimento de algoritmos de combinação de regras que considere todas as permissões ou negações dentro de uma política e assim não seria necessário utilizar uma estrutura maior como um conjunto de políticas. Seria muito mais simples e poderia vir a ter alguma vantagem sobre o modelo de politica de privacidade criado nesse documento.

REFERÊNCIAS

APACHE SOFTWARE FOUNDATION, **OpenAZ 2009-2017**, Disponível em: <http://incubator.apache.org/projects/openaz.html.> Acesso em 28 de fevereiro de 2017.

BERTINO, E. e TAKAHASHI, K. (2011). **Identity Management: Concepts, Technologies, and Systems**. Artech House

BIRREL, E. and SCHINEIDER, F. ,**Federated Identity Management Systems: A Privacy-based Characterization.** IEEE Security and Privacy, 2013. [30]

BODNAR L. M.; WESTPHALL C. M.; WERNER J.; WESTPHALL, C. B.; **Towards Privacy in Identity Management Dynamic Federations**, Universidade Federal de Santa Catarina, Disponível em: <https://www.researchgate.net/publication/299483504_Towards_Privacy_in_Identity_Management_Dynamic_Federations > Acesso em: 05 de maio de 2017.

BRAGANÇA, C.; LUCIANO E.; TESTA , M. **Segurança da Informação e privacidade de informações de pacientes de instituições de saúde: uma análise exploratória da privacidade percebida pelos profissionais**. EnANPAD, 2010.

BRANCO, E. C.; JAVAM, Jr.; MACHADO, C. ; MONTEIRO, J. M., **Estratégias para Proteção da Privacidade de Dados Armazenados na Nuvem, Tópicos em Gerenciamento de Dados e Informações**, Cap 2, SBC, 2014

CAMILLO, G. L. ; WESTPHALL, C. M. ; WERNER, J. ; WESTPHALL C. B.; **Preserving Privacy with Fine-grained Authorization in an Identity Management System**. 2017.

CHADWICK, D. W.; FATEMA, K. **A privacy preserving authorisation system for the cloud. Journal of Computer and System Sciences**, Elsevier, 2012.

ECLIPSE**, Eclipse Neon 3**, Disponível em: <http://www.eclipse.org/downloads/packages/release/Neon/3.RC3> Acesso em 2 de janeiro de 2017.

ECMA-404 **The JSON Data Interchange Standard**, Disponível em:< http://www.json.org> Acesso em 28 de fevereiro de 2017

EDUCAUSE review, **Information Privacy Revealed**, 2013 Disponível em: <http://er.educause.edu/articles/2013/1/information-privacy-revealed> Acesso em 2 de Julho de 2017.

GOLDBERG I.; WAGNER D.; BREWER E. **Privacy-enhancing technologies for the Internet**,1997.

HU, V. C.; KUHN, R.; FERRAIOLO, D. F., **Attribute-Based Access Control National, Institute of Standards and Technology**, Gaithersburg, MD, USA Attribute-Based Access Control Disponível em: <https://www.researchgate.net/publication/273393378_Attribute-Based_Access_Control. > Acessado em : 10 de março 2017.

HU, Vincent C. et al. **Conformance checking of access control policies specified in XACML. In: COMPUTER SOFTWARE AND APPLICATIONS CONFERENCE (COMPSAC)**, 2007, Pequim/China. Anais. Carolina do Norte EUA: IEEE Computer Society, 2007.

IBM, **Enterprise privacy authorization language**, 2003, Acessado em 03 de maio de 2017. Disponível em: <https://www.w3.org/Submission/2003/SUBM-EPAL-20031110/>

INCUBATOR-OPENAZ**,Incubator-openaz**, Disponível em: <https://github.com/apache/incubator-openaz> Acesso em 28 de fevereiro de 2017

ISHITANI , L., **Uma Arquitetura para controle de privacidade na Web**. Tese de doutorado , Universidade federal de Minas Gerais, 2003.

ISO/IEC 17799 , **Information technology - Security technical - Code of pratice for information security management**, 2005

ISO/IEC 24760-1, **Information technology - Security techniques - A framework for identity management** , 2011

ISO/IEC 29100. **international standard - information technology- security techniques - pri vacy framework**, 2011.

IYILADE, J. e VASSILEVA, J. , **P2U: A privacy policy specification language for secondary data sharing and usage**, 2014.

JETTY, **Jetty Eclipse Integration**, Disponível em: <http://eclipse-jetty.github.io/> Acesso em 28 de fevereiro de 2017.

JOSANG, A.; FABRE, J.; HAY, B.; DALZIEL J.; POPE S. **Trust Requirements in Identity Management**; Distributed Systems Technology Centre, Telstra Research Laboratories and Macquarie University.

JSON SIMPLE, **JSON Simple Code**, Disponível em: < https://code.google.com/archive/p/json-simple/> Acesso em 8 de abril de 2017.

KALEMPA, V. **Especificando privacidade em ambientes de computação ubíqua**. Dissertação (Mestrado), Universidade Federal de Santa Catarina, 2009.

LANDWEHR, C. et al. **Privacy and cybersecurity: The next 100 years**. Proceedings of the IEEE, IEEE, Special Centennial Issue, 2012.

LANGHEINRICH, M. **A privacy awareness system for ubiquitous computing environments**. In: Ubicomp, Lecture Notes in Computer Science. [S.l.]: Springer-Verlag, 2002.

LAUREANO. Marcos A. P. **Gestão de segurança da informação.** [S.l.], 2005. Disponível em: <http://pt.scribd.com/doc/100687499/18/Autenticacao-e-autorizacao> . Acesso em: 16 maio 2017.

LEANDRO, M. A. P., **Federação de identidades e computação em nuvem: Estudo de caso usando o Shibboleth**, Dissertação(Mestrado), Universidade Federal de Santa Catarina, 2012.

MACEDO, R**; Reforço da Privacidade Através do Controlo da Pegada Digital Privacidade**, Universidade do Minho Escola de Engenharia, Junho 2013.

MARCON Jr., Arlindo et al. **Aspectos de segurança e privacidade em ambientes de computação em nuvem**. In: Livro-texto de minicursos do SBSeg 2010. Porto Alegre, RS: Sociedade Brasileira de Computação, 2010.

MONT, M.C , **Privacy-aware Access Control for Personal Data** , Hewlett Packard Development Company, Cloud & Security Lab Bristol, UK 2011 Disponível em: <http://shiftleft.com/mirrors/www.hpl.hp.com/personal/Marco_Casassa_Mont/Projects/Pr

ivacyAwareAccessControl/PrivacyAwareAccessControl.htm> Acesso em 15 de maio de 2017.

MOZILLA, **HTTP Requester**, Extensões, Disponível em: <https://addons.mozilla.org/pt-BR/firefox/addon/httprequester/> Acesso em 13 de maio de 2017.]

NIST Special Publication 800-162. ,**Guide to Attribute Based Access Control (ABAC) Definition and Considerations**, 2013. Disponível em: < http://csrc.nist.gov/projects/abac/> Acesso em: 10 de maio de 2017.

OASIS STANDARD 2013, xacml-3.0-core-spec-os-en , eXtensible Access Control Markup Language (XACML) Version 3.0 Disponível em: <https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xacml> Acesso em 28 de fevereiro de 2017.

OASIS JSON **Profile of XACML 3.0 Version 1.0** (2011) < http://docs.oasis-open.org/xacml/xacml-json-http/v1.0/xacml-json-http-v1.0.html> Acesso em 10 de abril 2017.

OECD , **OECD Guidelines on the Protection of Privacy and Transborder Flows of Personal Data**, Cap 1, 2013.

OPENID**, OpenID Connect FAQ and Q&As** Disponvel em: <http://openid.net/connect/faq/> Acesso em: 11 de maio de 2017 .

PEARSON, S. e CHARLESWORTH, A., **Accountability as a Way Forward for Privacy Protection in the Cloud**, HP Labs, Long Down Avenue, Stoke Gifford, Bristol, UK. 2009.

POWER, R.,**Tangled Web: Tales of Digital Crime from the Shadows of Cyberspace**, 2000.

RAO , J. R. e ROHATGI, P., **Can Pseudonymity Relly Guarantee Privacy?**, USENIX Security Symposium, 2000.

RAULINO, A. F.; **Uma extensão do XACML para federação de identidades em nuvem computacional**, 2013.

RFC 7519, **JSON Web Token** , Disponível em: <https://tools.ietf.org/html/rfc7519/> Acesso em 8 de abril de 2017

ROSSET, V. **Um modelo de autorização e distribuição de direitos de acesso sobre conteúdos digitais**. 2004. Dissertação (Mestrado) – Centro de Ciências Exatas Universidade Federal de Santa Catarina, Florianópolis.

SANTOS, D. R., **Gerenciamento de identidades e privacidade em ambientes de computação em nuvem**, Universidade Federal de Santa Catarina, Julho de 2011 .

SILVA, E. F.; MUCHALUAT-SAADEL, D. ; FERNANDES, N. C. **Controle de Acesso Baseado em Políticas e Atributos para Federações de Recursos**, Universidade Federal Fluminense, Laboratório MídiaCom, SBSeg 2014.

SINNEMA, R.,**XACML- XML** Amsterdam 2011, Disponível em: <https://www.slideshare.net/canhnt/pst2013-xacml-engine128> Acesso em: 30 de maio de 2017.

STALLINGS, W**. Cryptography and Network Security - Principles and Practice**. 2. ed.Prentice-Hall, 1999.

SPRING **, Spring Framework**, Disponível em: <http://projects.spring.io/spring-framework/> Acesso em 20 de fevereiro de 2017.

SPRING BOOT, **Spring Boot**, Spring. Disponível em: <https://spring.io/guides/gs/spring-boot/> Acesso em 28 de fevereiro de 2017.

SPRING INITIALIZR ,**Spring Initializr and Pivotal Web Services**, Disponível em: https://start.spring.io /> Acesso em 8 de abril de 2017.

STUFFLEBEAM, W. ; ANTÓN , A. I.;HE O.; JAIN N. **Specifying Privacy Policies awith P3P and EPAL**: Lessons Learned, Department of Computer Science, North Carolina State University, s.d.

VILLAREAL, M. E.; WESTPHALL, C. M. **Privacy Token: A Mechanism for User's Privacy Specification in Identity Management Systems for the Cloud**, 2017.

W3C, **The platform for privacy preferences 1.1 specification**, 2006, acessado em 03 de maio de 2017. Disponível em:<https://www.w3.org/TR/P3P11/>

WEINGÄRTNER, R. **Controle de disseminação de dados sensíveis em ambientes federados**, Dissertação de mestrado, 2014

WERNER, J.; WESTPHALL, C., "**A Model for Identity Management with Privacy in the Cloud**" in 2016 IEEE Symposium on Computers and Communication (ISCC), Messina, Italy, June 2016.

WERNER J., WESTPHALL C.M, WESTPHALL C. B., **Cloud identity management: A survey on privacy strategies**. Computer Networks 122, Elsevier. 2017

WESPHALL, C. M.. **Um Esquema de Autorização para a Segurança de Sistemas Distribuídos de Larga Escala.** Tese de Doutorado, PGEELUFSC, Florianópolis, SC, dezembro de 2000.

ZANELATO, M. A. **Proposta de um modelo de biblioteca cliente para o Shibboleth: Facilitando a adaptação das aplicações**, Universidade Federal de Santa Catarina, 2014.

APÊNDICE A – REQUEST CREATOR

```
package jsonrequescreator;

import java.util.ArrayList;

public class Preferences {


    public String[] dat = { "PI", "PCP", "LO", "AH", "RS"};
    public String[] fin = {"SI", "SC", "CO"};
    public String[] ben = {"PP", "SP", "TP"};

    public ArrayList<String> preferencia;
    public ArrayList<String> tag_preferencia;
    ArrayList<String> pref_ativos = new ArrayList<>();

    public Preferences() {
        preferencia = new ArrayList<>();
        tag_preferencia = new ArrayList<>();
        initPreferencias();
    }

    public final void initPreferencias( ){

        int cont=0;
        for(int i=0 ; i<5; i++)
            for(int j=0 ; j<3; j++)
                for(int k=0 ; k<3; k++){
                    setTag_preferencia( cont, getDat(i)+"_"+getFin(j)+"_"+getBen(k));
                    setPreferencia( cont,"0");
                    cont++;
                }
    }
//Getter de todas as preferencias ativas
    public ArrayList<String> getTodasPreferenciasAtivas(){
        int i=0;
        for(String pref : preferencia){
                if("1".equals(pref))
                    pref_ativos.add(tag_preferencia.get(i).toLowerCase());

                i++;
        }
        return pref_ativos;
    }
//Getter de todas as preferencias inativas
    public ArrayList<String> getTodasPreferenciasNaoAtivas(){

        int i=0;
```

```
    for(String pref : preferencia){
            if("0".equals(pref))
               pref_ativos.add(this.getTag_preferencia(i));
            i++;
    }
    return pref_ativos;
}

public ArrayList<String> getTodasTagPreferencia() {

    for(int i=0;i<45;i++)
       tag_preferencia.get(i);
    return tag_preferencia;
}

public void setDat(String[] dat) {
    this.dat = dat;
}

public void setFin(String[] fin) {
    this.fin = fin;
}

public void setBen(String[] ben) {
    this.ben = ben;
}

public String getPreferencia(int i) {
    return preferencia.get(i);
}

public void setPreferencia( int i, String value ) {

    preferencia.add(i, value);

}

public String getTag_preferencia(int i) {
    return preferencia.get(i);
}

public void setTag_preferencia(int i, String value) {
    this.tag_preferencia.add(i, value);
}

public String getDat(int i) {
    return dat[i];
}

public String getFin(int i) {
```

```
        return fin[i];
    }

    public String getBen(int i) {
        return ben[i];
    }


}



package jsonrequescreator;

import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.util.ArrayList;
import org.json.simple.JSONArray;
import org.json.simple.JSONObject;
import org.json.simple.parser.JSONParser;
import org.json.simple.parser.ParseException;

public class JSONRequesCreator {

  // E AQUI QUE COLETAMOS OS DADOS DO REQUEST
    public static void main(String[] args) {
        Token JSONPrivToken = new Token();
        JSONPrivToken.setPreferencias(new Preferences());
        Request JSONReq = new Request();
       //Esses dados podem vir de uma tela
        JSONReq.setRecurso("www.sitedecompras.com:userid:historico");
        JSONReq.setAcao("comprar");

        JSONReq.setName("request_1_1");

        JSONReq.setTipoDeDados("PCP");
        JSONReq.setTipoDeDados("PI");

        JSONReq.setFinalidade("SC");
        JSONReq.setFinalidade("SI");

        JSONReq.setBeneficiario("PP");
        JSONReq.setBeneficiario("SP");
        JSONReq.setPreferencias(JSONTokenRead(JSONPrivToken));
        JSONRequestWrite(JSONReq,"windows");
        JSONRequestView(JSONReq);
    }
```

```
   public static Preferences JSONTokenRead(Token JSONPrivToken){

                    JSONObject jsonObject;
                    //Cria o parse de tratamento
                    JSONParser parser = new JSONParser();

               try {
                  jsonObject = (JSONObject) parser.parse(new
FileReader("model_token/token.json"));
                    //Salva nas variaveis os dados retirados do arquivo

                  JSONObject estrutura = (JSONObject) jsonObject.get("estrutura");
                  JSONPrivToken.setTyp((String) estrutura.get("typ"));
                  JSONPrivToken.setAlg((String) estrutura.get("alg"));
                  JSONObject  atributos = (JSONObject)jsonObject.get("atributos");
                  JSONObject  preferencias =(JSONObject) atributos.get("preferencias");
                  JSONPrivToken.setSub((String) atributos.get("sub"));
                  JSONPrivToken.setIss((String) atributos.get("iss"));
                  JSONPrivToken.setAud((String) atributos.get("aud"));
                  JSONPrivToken.setIat((String) atributos.get("iat"));

                  String prefTag;


                  for(int i=0; i<45; i++){

                     prefTag = JSONPrivToken.getPreferencias().getTodasTagPreferencia().get(i);

                     JSONPrivToken.getPreferencias().setPreferencia(i, (String)
preferencias.get(prefTag));
                  }

                  String pref="\n";
                  for(int i=0; i<45; i++){
                     pref= pref+"
"+JSONPrivToken.getPreferencias().getTodasTagPreferencia().get(i)+": "+
JSONPrivToken.getPreferencias().getPreferencia(i)+"\n";
                  }

//                  String tkn = "{\n estrutura:{\n   typ: "+JSONPrivToken.getTyp()+"\n"
//                        + "   alg: "+JSONPrivToken.getAlg()+"\n }\n"
//                        + "\n atributos:{"
//                        + "\n  sub: " +JSONPrivToken.getSub()
//                        + "\n  iss: "  +JSONPrivToken.getIss()
//                        + "\n  aud: "  +JSONPrivToken.getAud()
//                        + "\n  iat: "  +JSONPrivToken.getIat()
//                        + "\n\n  preferencias:{"+pref+"  }\n }\n}\n";
//                  System.out.printf(""+tkn);
               }
                    //Trata as exceptions que podem ser lançadas no decorrer do processo
```

```
                catch (FileNotFoundException e) {
                        e.printStackTrace();
                } catch (IOException e) {
                        e.printStackTrace();
                } catch (ParseException e) {
                        // TODO Auto-generated catch block
                        e.printStackTrace();
                }

        return JSONPrivToken.getPreferencias();
    }

  public static void JSONRequestWrite(Request JSONReq, String OS){

                //Cria um Objeto JSON e Array json
        ArrayList<JSONObject> jsonAttributeRsc = new ArrayList<>() ;
        ArrayList<JSONObject> jsonAttributeSub = new ArrayList<>() ;
        ArrayList<JSONObject> jsonAttributeAct = new ArrayList<>() ;

        JSONObject jsonObject = new JSONObject();
        JSONObject jsonRequest = new JSONObject();

        JSONObject jsonSubject = new JSONObject();
        JSONObject jsonAction = new JSONObject();
        JSONObject jsonResource = new JSONObject();

                FileWriter writeFile = null;
                    //Valorar cada atributo
        int i =0;

attrBuildWrite(jsonAttributeRsc,JSONReq.getRecurso(),"string" ,"resource:resource-id", i);
        i++;


        for(String fin: JSONReq.getFinalidade()){
           attrBuildWrite(jsonAttributeRsc, fin, "string", "resource:resource-id", i);

           i++;
        }

        for(String typ: JSONReq.getTipoDeDados()){
           attrBuildWrite(jsonAttributeRsc, typ, "string", "resource:tipo-dado", i);
           i++;
        }

        i =0;
        attrBuildWrite(jsonAttributeAct, JSONReq.getAcao(), "string", "action:action-id",
i);

        for(String ben: JSONReq.getBeneficiario()){
```

```
            attrBuildWrite(jsonAttributeSub, ben, "string", "subject:subject-id", i);
            i++;
        }

        for(String pre: JSONReq.getPreferencias().getTodasPreferenciasAtivas()){
            attrBuildWrite(jsonAttributeSub, pre, "string", "subject:preferences", i);
            i++;
        };

        jsonSubject.put("Attribute", jsonAttributeSub);
        jsonAction.put("Attribute", jsonAttributeAct);
        jsonResource.put("Attribute", jsonAttributeRsc);

        jsonRequest.put("AccessSubject", jsonSubject);
        jsonRequest.put("Action", jsonAction);
        jsonRequest.put("Resource", jsonResource);

        jsonObject.put("Request", jsonRequest);

            try{

//                      if( "windows".equals(OS))
//
        writeFile = new FileWriter(JSONReq.getName()+".json");
                    //Escreve no arquivo conteudo do Objeto JSON
                    writeFile.write(jsonObject.toJSONString());
                    writeFile.close();
            }
            catch(IOException e){
                    e.printStackTrace();
            }
            //Imprimne na Tela o Objeto JSON para vizualização
            //System.out.println(jsonObject);

    }

  public static void JSONRequestView(Request JSONReq){

    JSONObject jsonObject;
            //Cria o parse de tratamento
       JSONParser parser = new JSONParser();

    try {
    jsonObject = (JSONObject) parser.parse(new FileReader(JSONReq.getName()+".json"));
            //Salva nas variaveis os dados retirados do arquivo
    JSONObject request = (JSONObject) jsonObject.get("Request");
            JSONObject subject = (JSONObject)request.get("AccessSubject");
            JSONObject action =(JSONObject) request.get("Action");
            JSONObject resource =(JSONObject) request.get("Resource");
            JSONArray attributesSub =(JSONArray) subject.get("Attribute");
```

```
                JSONArray attributesAct =(JSONArray) action.get("Attribute");
                JSONArray attributesRsc =(JSONArray) resource.get("Attribute");

        String s = attrBuildRead(attributesSub);
        String a = attrBuildRead(attributesAct);
        String r = attrBuildRead(attributesRsc);
        reqBuild(s,a,r);

                //Trata as exceptions que podem ser lançadas no decorrer do processo
        }catch (FileNotFoundException e) {
                e.printStackTrace();
            } catch (IOException e) {
                e.printStackTrace();
            } catch (ParseException e) {
                        // TODO Auto-generated catch block
                e.printStackTrace();
            }

    }

    public static  void attrBuildWrite( ArrayList<JSONObject> jsonAttributeRsc,String
val,Object dTyp ,String att, int i){

            jsonAttributeRsc.add(new JSONObject());
            jsonAttributeRsc.get(i).put("Value", val);
            jsonAttributeRsc.get(i).put("DataType", dTyp);
            jsonAttributeRsc.get(i).put("AttributeId", att);

    }

    public static  String attrBuildRead(JSONArray arrJson){

        String s= "";
        String a= "";
        String r= "";
        String aid="";
        String val="";
        String dt="";
        for(Object obj : arrJson){
            aid=  (String) ((JSONObject) obj).get("AttributeId");
            dt=  (String) ((JSONObject) obj).get("DataType");
            val=  (String) ((JSONObject) obj).get("Value");
            s+="\n       {\n            \"AttributeId\" : \""+aid+"\","
             +"\n            \"DataType\" : \""+dt+"\","
             +"\n            \"Value\" : \""+val
             +"\"\n       }";
            if(arrJson.indexOf(obj)+1!=arrJson.size())
                s+=",";
        }
        return s;
```

```
    }


    public static void reqBuild(String sub, String act, String rsc){

        String req = "{\n \"Request\" : {   \n"
        + "   \"AccessSubject\" : { \n    "
        + "   \" Attribute\" : [ "+sub+"\n        ]\n\n   }\n"
        + "   \"Action\" : { \n "
        + "       \"Attribute\" : [{ "+act+"\n        }]\n\n   }\n"
        + "   \"Resource\" : { \n "
        + "       \"Attribute\" : [{ "+rsc+"\n        }]\n\n   }\n"
        + "} \n }";
        System.out.printf(""+req);
    }
}



package jsonrequescreator;

import java.util.ArrayList;

public class Token {

   private String typ;
   private String alg;
   private String sub;
   private String iss;
   private String aud;
   private String iat;

   private ArrayList<String> tipoDeDados = new ArrayList<>();
   private ArrayList<String> finalidade = new ArrayList<>();
   private final ArrayList<String> beneficiario;

   protected Preferences preferencias;

    public Token() {
       this.beneficiario = new ArrayList<>();
    }

   public void setTodasPreferencias(){

       preferencias = new Preferences();
       String s;

       for(String td : tipoDeDados)
          for (String fi : finalidade)
             for (String be  : beneficiario) {
```

```
                s = td+"_"+fi +"_"+be;
                if(preferencias.tag_preferencia.contains(s))
                    preferencias.setPreferencia(preferencias.tag_preferencia.indexOf(s),"1");
            }
}

public Preferences getPreferencias() {
    return preferencias;
}

public void setPreferencias(Preferences preferencias) {
    this.preferencias = preferencias;
}

public String getTyp() {
    return typ;
}

public void setTyp(String typ) {
    this.typ = typ;
}

public String getAlg() {
    return alg;
}
public void setAlg(String alg) {
    this.alg = alg;
}
public String getSub() {
    return sub;
}

public void setSub(String sub) {
    this.sub = sub;
}

public String getIss() {
    return iss;
}

public void setIss(String iss) {
    this.iss = iss;
}

public String getAud() {
    return aud;
}

public void setAud(String aud) {
    this.aud = aud;
```

```
    }

    public String getIat() {
        return iat;
    }

    public void setIat(String iat) {
        this.iat = iat;
    }

    public ArrayList<String> getTipoDeDados() {
        return tipoDeDados;
    }

    public void setTipoDeDadosArr(ArrayList<String> arr) {
        this.tipoDeDados= arr;
    }

    public void setTipoDeDados(String tipoDeDados) {
        this.tipoDeDados.add(tipoDeDados);
    }

    public ArrayList<String> getFinalidade() {
        return finalidade;
    }

    public void setFinalidadeArr(ArrayList<String> arr) {
        this.finalidade = arr;
    }

    public void setFinalidade(String finalidade) {
        this.finalidade.add(finalidade);
    }

    public void setBeneficiarioArr(ArrayList<String> arr) {
        this.finalidade = arr;
    }

    public ArrayList<String> getBeneficiario() {
        return beneficiario;
    }

    public void setBeneficiario(String beneficiario) {
        this.beneficiario.add(beneficiario);
    }

}

package jsonrequescreator;
```

```
import java.util.ArrayList;


public class Request {

   //Atributos necessarios para a requisição de privacidade.
   private String name;
   private String recurso;
   private String acao;

   private ArrayList<String> tipoDeDados = new ArrayList<>();
   private ArrayList<String> finalidade = new ArrayList<>();
   private final ArrayList<String> beneficiario;
   protected Preferences preferencias;

   public Request() {
      this.beneficiario = new ArrayList<>();
   }

   public void setTodasPreferencias(){

      preferencias = new Preferences();
      String s;

      for(String td : tipoDeDados)
         for (String fi : finalidade)
            for (String be  : beneficiario) {
               s = td+"_"+fi +"_"+be;
               if(preferencias.tag_preferencia.contains(s))
                  preferencias.setPreferencia(preferencias.tag_preferencia.indexOf(s),"1");
            }
   }

   public ArrayList<String> getTodasPreferenciasAtivas(){

      preferencias = new Preferences();
      return preferencias.getTodasPreferenciasAtivas();
   }

   public Preferences getPreferencias() {
      return preferencias;
   }

   public void setPreferencias(Preferences preferencias) {
      this.preferencias = preferencias;
   }

   public ArrayList<String> getTipoDeDados() {
      return tipoDeDados;
   }
```

```
public void setTipoDeDadosArr(ArrayList<String> arr) {
   this.tipoDeDados= arr;
}

public void setTipoDeDados(String tipoDeDados) {
   this.tipoDeDados.add(tipoDeDados);
}

public ArrayList<String> getFinalidade() {
   return finalidade;
}

public void setFinalidadeArr(ArrayList<String> arr) {
   this.finalidade = arr;
}

public void setFinalidade(String finalidade) {
   this.finalidade.add(finalidade);
}

public void setBeneficiarioArr(ArrayList<String> arr) {
   this.finalidade = arr;
}

public ArrayList<String> getBeneficiario() {
   return beneficiario;
}

public void setBeneficiario(String beneficiario) {
   this.beneficiario.add(beneficiario);
}

public String getRecurso() {
   return recurso;
}

public void setRecurso(String recurso) {
   this.recurso = recurso;
}

public String getAcao() {
   return acao;
}

public void setAcao(String acao) {
   this.acao = acao;
}

public void setName(String name) {
```

```
        this.name = name;
    }

    public String getName() {
      return name;
    }

}
```

APÊNDICE B – PRIVACY TOKEN CREATOR

```
package novotestejson;

import java.util.ArrayList;

public class JSONToken {

   private String nome;
   //Atributos pertencentes a estrutura
   private String typ;
   private String alg;
   //Atributos pertencentes a
   private String sub;
   private String iss;
   private String aud;
   private String iat;

   //Preferencias oferencidas pela interface
   private ArrayList<String> tipoDeDados = new ArrayList<>();
   private ArrayList<String> finalidade = new ArrayLiast<>();
   private final ArrayList<String> beneficiario;
    //Preferencias oferencidas pelo
   protected Preferences preferencias;

    public JSONToken() {
        this.beneficiario = new ArrayList<>();
    }

    public void setTodasPreferencias(){

        preferencias = new Preferences();
        String s;

        for(String td : tipoDeDados)
            for (String fi : finalidade)
                for (String be  : beneficiario) {
                    s = td+"_"+fi +"_"+be;
```

```
            if(preferencias.tag_preferencia.contains(s))
               preferencias.setPreferencia(preferencias.tag_preferencia.indexOf(s),"1");
         }
}

public void getTodasPreferencias(){

   preferencias = new Preferences();
   String s;

}

public Preferences getPreferencias() {
   return preferencias;
}

public void setPreferencias(Preferences preferencias) {
   this.preferencias = preferencias;
}

public String getNome() {
   return nome;
}

public void setNome(String nome) {
   this.nome = nome;
}

public String getTyp() {
   return typ;
}

public void setTyp(String typ) {
   this.typ = typ;
}

public String getAlg() {
   return alg;
}

public void setAlg(String alg) {
   this.alg = alg;
}

public String getSub() {
   return sub;
}

public void setSub(String sub) {
   this.sub = sub;
```

```
}

public String getIss() {
   return iss;
}

public void setIss(String iss) {
   this.iss = iss;
}

public String getAud() {
   return aud;
}

public void setAud(String aud) {
   this.aud = aud;
}

public String getIat() {
   return iat;
}

public void setIat(String iat) {
   this.iat = iat;
}

public ArrayList<String> getTipoDeDados() {
   return tipoDeDados;
}

public void setTipoDeDadosArr(ArrayList<String> arr) {
   this.tipoDeDados= arr;
}

public void setTipoDeDados(String tipoDeDados) {
   this.tipoDeDados.add(tipoDeDados);
}

public ArrayList<String> getFinalidade() {
   return finalidade;
}

public void setFinalidadeArr(ArrayList<String> arr) {
   this.finalidade = arr;
}

public void setFinalidade(String finalidade) {
   this.finalidade.add(finalidade);
}
```

```
    public void setBeneficiarioArr(ArrayList<String> arr) {
        this.finalidade = arr;
    }

    public ArrayList<String> getBeneficiario() {
        return beneficiario;
    }

    public void setBeneficiario(String beneficiario) {
        this.beneficiario.add(beneficiario);
    }

}


package novotestejson;

import java.util.ArrayList;

public class Preferences {

//Atributos usados apenas para referenciar o tag (Mudar para Enum)
    public String[] dat = { "PI", "PCP", "LO", "AH", "RS"};
    public String[] fin = {"SI", "SC", "CO"};
    public String[] ben = {"PP", "SP", "TP"};
//Array de valores das preferencias
    public ArrayList<String> preferencia;
//Array de tag das preferencias
    public ArrayList<String> tag_preferencia;

//Construtor das Preferencias
    public Preferences() {
        preferencia = new ArrayList<>();
        tag_preferencia = new ArrayList<>();
        initPreferencias();
    }
//Inicializa cada valor de preferencia com 0 para cada tag
    public final void initPreferencias( ){

        int cont=0;
        for(int i=0 ; i<5; i++)
            for(int j=0 ; j<3; j++)
                for(int k=0 ; k<3; k++){
                    setTag_preferencia( cont, getDat(i)+"_"+getFin(j)+"_"+getBen(k));
                    setPreferencia( cont,"0");
                    cont++;
                }
    }
//
    public ArrayList<String> getTodasTagPreferencia() {
```

```
        for(int i=0;i<45;i++)
            tag_preferencia.get(i);
        return tag_preferencia;
            }
//Setters
    public void setDat(String[] dat) {
        this.dat = dat;
    }

    public void setFin(String[] fin) {
        this.fin = fin;
    }

    public void setBen(String[] ben) {
        this.ben = ben;
    }

    public void setPreferencia( int i, String value ) {

        preferencia.add(i, value);

    }

    public void setTag_preferencia(int i, String value) {
        this.tag_preferencia.add(i, value);
    }
//Getters
    public String getDat(int i) {
        return dat[i];
    }

    public String getFin(int i) {
        return fin[i];
    }

    public String getBen(int i) {
        return ben[i];
    }

    public String getTag_preferencia(int i) {
        return preferencia.get(i);
    }

    public String getPreferencia(int i) {
        return preferencia.get(i);
    }

}
```

```
package novotestejson;

import java.util.ArrayList;

public class Preferences {

//Atributos usados apenas para referenciar o tag (Mudar para Enum)
   public String[] dat = { "PI", "PCP", "LO", "AH", "RS"};
   public String[] fin = {"SI", "SC", "CO"};
   public String[] ben = {"PP", "SP", "TP"};
//Array de valores das preferencias
   public ArrayList<String> preferencia;
//Array de tag das preferencias
   public ArrayList<String> tag_preferencia;

//Construtor das Preferencias
   public Preferences() {
      preferencia = new ArrayList<>();
      tag_preferencia = new ArrayList<>();
      initPreferencias();
   }
//Inicializa cada valor de preferencia com 0 para cada tag
   public final void initPreferencias( ){

      int cont=0;
      for(int i=0 ; i<5; i++)
         for(int j=0 ; j<3; j++)
            for(int k=0 ; k<3; k++){
               setTag_preferencia( cont, getDat(i)+"_"+getFin(j)+"_"+getBen(k));
               setPreferencia( cont,"0");
               cont++;
            }
   }
//
   public ArrayList<String> getTodasTagPreferencia() {
      for(int i=0;i<45;i++)
         tag_preferencia.get(i);
      return tag_preferencia;
         }
//Setters
   public void setDat(String[] dat) {
      this.dat = dat;
   }

   public void setFin(String[] fin) {
      this.fin = fin;
   }

   public void setBen(String[] ben) {
```

```
        this.ben = ben;
    }

    public void setPreferencia( int i, String value ) {

        preferencia.add(i, value);

    }

    public void setTag_preferencia(int i, String value) {
        this.tag_preferencia.add(i, value);
    }
//Getters
    public String getDat(int i) {
        return dat[i];
    }

    public String getFin(int i) {
        return fin[i];
    }

    public String getBen(int i) {
        return ben[i];
    }

    public String getTag_preferencia(int i) {
        return preferencia.get(i);
    }

    public String getPreferencia(int i) {
        return preferencia.get(i);
    }

}


/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package novotestejson;

import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import org.json.simple.JSONObject;
import org.json.simple.parser.JSONParser;
import org.json.simple.parser.ParseException;
```

```
public class PrivacyTokenCreator {

   public static void main(String[] args) {


      JSONToken JSONPrivToken = new JSONToken();

      JSONPrivToken.setTyp("JWT");
      JSONPrivToken.setAlg("SH256");

      JSONPrivToken.setSub("Alice");
      JSONPrivToken.setIss("http://openid.com.br");
      JSONPrivToken.setAud("Cliente-1");
      JSONPrivToken.setIat("1000101010");

      JSONPrivToken.setTipoDeDados("PI");
      JSONPrivToken.setTipoDeDados("PCP");
      JSONPrivToken.setTipoDeDados("LO");
      JSONPrivToken.setTipoDeDados("AH");
      JSONPrivToken.setTipoDeDados("RS");

      JSONPrivToken.setFinalidade("SI");
      JSONPrivToken.setFinalidade("SC");
      JSONPrivToken.setFinalidade("CO");

      JSONPrivToken.setBeneficiario("PP");
      JSONPrivToken.setBeneficiario("SP");
      JSONPrivToken.setBeneficiario("TP");

      JSONPrivToken.setTodasPreferencias();

      JSONWrite(JSONPrivToken);
      JSONRead(JSONPrivToken);
   }

   public static JSONToken JSONRead(JSONToken JSONPrivToken){

      JSONObject jsonObject;
                  //Cria o parse de tratamento
                  JSONParser parser = new JSONParser();

      try {
                        jsonObject = (JSONObject) parser.parse(new FileReader("token.json"));
         //Salva nas variaveis os dados retirados do arquivo

         JSONObject estrutura = (JSONObject) jsonObject.get("estrutura");
         JSONPrivToken.setTyp((String) estrutura.get("typ"));
         JSONPrivToken.setAlg((String) estrutura.get("alg"));
```

```
        JSONObject  atributos = (JSONObject)jsonObject.get("atributos");
        JSONObject  preferencias =(JSONObject) atributos.get("preferencias");
        JSONPrivToken.setSub((String) atributos.get("sub"));
        JSONPrivToken.setIss((String) atributos.get("iss"));
        JSONPrivToken.setAud((String) atributos.get("aud"));
        JSONPrivToken.setIat((String) atributos.get("iat"));

        String prefTag;

        for(int i=0; i<45; i++){
                                prefTag =
JSONPrivToken.getPreferencias().getTodasTagPreferencia().get(i);
           JSONPrivToken.getPreferencias().setPreferencia(i, (String)
preferencias.get(prefTag));
        }

        String pref="\n";
        for(int i=0; i<45; i++){
           pref= pref+"
"+JSONPrivToken.getPreferencias().getTodasTagPreferencia().get(i)+": "+
JSONPrivToken.getPreferencias().getPreferencia(i)+"\n";
        }

                        String tkn = "{\n estrutura:{\n   typ: "+JSONPrivToken.getTyp()+"\n"
           + "   alg: "+JSONPrivToken.getAlg()+"\n }\n"
           + "\n atributos:{"
           + "\n  sub: " +JSONPrivToken.getSub()
           + "\n  iss: "  +JSONPrivToken.getIss()
           + "\n  aud: "  +JSONPrivToken.getAud()
           + "\n  iat: "  +JSONPrivToken.getIat()
           + "\n\n  preferencias:{"+pref+"  }\n }\n}\n";
           System.out.printf(""+tkn);
     }
                //Trata as exceptions que podem ser lançadas no decorrer do processo
                catch (FileNotFoundException e) {
                        e.printStackTrace();
                } catch (IOException e) {
                        e.printStackTrace();
                } catch (ParseException e) {
                        // TODO Auto-generated catch block
                        e.printStackTrace();
                }

        return JSONPrivToken;
     }

   public static void JSONWrite(JSONToken JSONPrivToken){


                JSONObject jsonObject = new JSONObject();
```

```
		JSONObject jsonStruct = new JSONObject();
	JSONObject jsonAttr = new JSONObject();
	JSONObject jsonPref = new JSONObject();

		FileWriter writeFile = null;

	jsonObject.put("nome",JSONPrivToken.getNome());
	jsonStruct.put("typ",JSONPrivToken.getTyp() );
		jsonStruct.put("alg", JSONPrivToken.getAlg());
	jsonAttr.put("sub", JSONPrivToken.getSub());
		jsonAttr.put("iss", JSONPrivToken.getIss());
		jsonAttr.put("aud", JSONPrivToken.getAud());
		jsonAttr.put("iat", JSONPrivToken.getIat());

	for(int i=0; i<45; i++){

jsonPref.put(JSONPrivToken.getPreferencias().getTodasTagPreferencia().get(i),JSONPrivTo
ken.getPreferencias().getPreferencia(i) );
	}

	jsonAttr.put("preferencias", jsonPref);
	jsonObject.put("atributos", jsonAttr);
	jsonObject.put("estrutura",jsonStruct);

		try{
			writeFile = new FileWriter("token.json");
			//Escreve no arquivo conteudo do Objeto JSON
			writeFile.write(jsonObject.toJSONString());
			writeFile.close();
		}
		catch(IOException e){
			e.printStackTrace();
		}
		//Imprimne na Tela o Objeto JSON para vizualização
		//System.out.println(jsonObject);
	}
}
```

APÊNDICE C – EXEMPLOS DE REQUEST

REQUISIÇÃO 1.1


```
{
 "Request" : {
   "AccessSubject" : {
      "Attribute" : [
       {
          "AttributeId" : "subject:subject-id",
          "DataType" : "string",
          "Value" : "PP"
         },
         {
          "AttributeId" : "subject:preferences",
          "DataType" : "string",
          "Value" : "pi_si_pp"
       },
       {
          "AttributeId" : "subject:preferences",
          "DataType" : "string",
          "Value" : "pi_si_sp"
       },
       {
          "AttributeId" : "subject:preferences",
          "DataType" : "string",
          "Value" : "pi_si_tp"
       },
       {
          "AttributeId" : "subject:preferences",
          "DataType" : "string",
          "Value" : "pi_sc_pp"
       },
       {
          "AttributeId" : "subject:preferences",
          "DataType" : "string",
          "Value" : "pi_sc_sp"
       },
       {
          "AttributeId" : "subject:preferences",
          "DataType" : "string",
          "Value" : "pi_sc_tp"
       },
       {
          "AttributeId" : "subject:preferences",
          "DataType" : "string",
          "Value" : "pi_co_pp"
       },
       {
```

```
    "AttributeId" : "subject:preferences",
    "DataType" : "string",
    "Value" : "pi_co_sp"
},
{
    "AttributeId" : "subject:preferences",
    "DataType" : "string",
    "Value" : "pi_co_tp"
},
{
    "AttributeId" : "subject:preferences",
    "DataType" : "string",
    "Value" : "ah_si_pp"
},
{
    "AttributeId" : "subject:preferences",
    "DataType" : "string",
    "Value" : "ah_si_sp"
},
{
    "AttributeId" : "subject:preferences",
    "DataType" : "string",
    "Value" : "ah_si_tp"
},
{
    "AttributeId" : "subject:preferences",
    "DataType" : "string",
    "Value" : "ah_sc_pp"
},
{
    "AttributeId" : "subject:preferences",
    "DataType" : "string",
    "Value" : "ah_sc_sp"
},
{
    "AttributeId" : "subject:preferences",
    "DataType" : "string",
    "Value" : "ah_sc_tp"
},
{
    "AttributeId" : "subject:preferences",
    "DataType" : "string",
    "Value" : "ah_co_pp"
},
{
    "AttributeId" : "subject:preferences",
    "DataType" : "string",
    "Value" : "ah_co_sp"
},
{
```

```
    "AttributeId" : "subject:preferences",
    "DataType" : "string",
    "Value" : "ah_co_tp"
},
{
    "AttributeId" : "subject:preferences",
    "DataType" : "string",
    "Value" : "rs_si_pp"
},
{
    "AttributeId" : "subject:preferences",
    "DataType" : "string",
    "Value" : "rs_si_sp"
},
{
    "AttributeId" : "subject:preferences",
    "DataType" : "string",
    "Value" : "rs_si_tp"
},
{
    "AttributeId" : "subject:preferences",
    "DataType" : "string",
    "Value" : "rs_sc_pp"
},
{
    "AttributeId" : "subject:preferences",
    "DataType" : "string",
    "Value" : "rs_sc_sp"
},
{
    "AttributeId" : "subject:preferences",
    "DataType" : "string",
    "Value" : "rs_sc_tp"
},
{
    "AttributeId" : "subject:preferences",
    "DataType" : "string",
    "Value" : "rs_co_pp"
},
{
    "AttributeId" : "subject:preferences",
    "DataType" : "string",
    "Value" : "rs_co_sp"
},
{
    "AttributeId" : "subject:preferences",
    "DataType" : "string",
    "Value" : "rs_co_tp"
}
]
```

```
  },
  "Action" : {
     "Attribute" : {

        "AttributeId" : "action:action-id",
        "DataType" : "string",
        "Value" : "comprar"

     }

  },
  "Resource" : {
     "Attribute" : [{

        "AttributeId" : "resource:resource-id",
        "DataType" : "string",
        "Value" : "www.sitedecompras.com:compras:produtoid"
     },
     {
        "AttributeId" : "resource:finalidade",
        "DataType" : "string",
        "Value" : "SC"
     },
     {
        "AttributeId" : "resource:tipo-dado",
        "DataType" : "string",
        "Value" : "PI"
     }
     ]

  }
}
 }
```

REQUISIÇÃO 1.2


```
{
        "Request" : {
                "AccessSubject" : {
                        "Attribute" : [
                                {
                                   "AttributeId" : "subject:subject-id",
                                   "DataType" : "string",
                                   "Value" : "PP"
                                },
                                {
                                   "AttributeId" : "subject:preferences",
                                   "DataType" : "string",
                                   "Value" : "ah_si_sp"
                                },
                                {
                                   "AttributeId" : "subject:preferences",
                                   "DataType" : "string",
                                   "Value" : "ah_si_tp"
                                },
                                {
                                   "AttributeId" : "subject:preferences",
                                   "DataType" : "string",
                                   "Value" : "ah_co_sp"
                                },
                                {
                                   "AttributeId" : "subject:preferences",
                                   "DataType" : "string",
                                   "Value" : "ah_co_tp"
                                },
                                {
                                   "AttributeId" : "subject:preferences",
                                   "DataType" : "string",
                                   "Value" : "rs_si_sp"
                                },
                                {
                                   "AttributeId" : "subject:preferences",
                                   "DataType" : "string",
                                   "Value" : "rs_si_tp"
                                },
                                {
                                   "AttributeId" : "subject:preferences",
                                   "DataType" : "string",
                                   "Value" : "rs_co_sp"
                                },
                                {
                                   "AttributeId" : "subject:preferences",
                                   "DataType" : "string",
                                   "Value" : "rs_co_tp"
```

```
                                }
                            ]

            },
    "Action" : {
        "Attribute" : {

            "AttributeId" : "action:action-id",
            "DataType" : "string",
            "Value" : "comprar"

        }

    },
    "Resource" : {
        "Attribute" : [{

            "AttributeId" : "resource:resource-id",
            "DataType" : "string",
            "Value" : "www.sitedecompras.com:compras:produtoid"
        },
        {
            "AttributeId" : "resource:finalidade",
            "DataType" : "string",
            "Value" : "SC"
        },
        {
            "AttributeId" : "resource:tipo-dado",
            "DataType" : "string",
            "Value" : "PI"
        }
        ]
}
    }
}
```

REQUISIÇÃO 2.1


```
{
 "Request" : {
   "AccessSubject" : {
      "Attribute" : [
       {
           "AttributeId" : "subject:subject-id",
           "DataType" : "string",
           "Value" : "SP"
          },
          {
           "AttributeId" : "subject:preferences",
           "DataType" : "string",
           "Value" : "pi_si_pp"
       },
       {
           "AttributeId" : "subject:preferences",
           "DataType" : "string",
           "Value" : "pi_si_sp"
       },
       {
           "AttributeId" : "subject:preferences",
           "DataType" : "string",
           "Value" : "pi_si_tp"
       },
       {
           "AttributeId" : "subject:preferences",
           "DataType" : "string",
           "Value" : "pi_sc_pp"
       },
       {
           "AttributeId" : "subject:preferences",
           "DataType" : "string",
           "Value" : "pi_sc_sp"
       },
       {
           "AttributeId" : "subject:preferences",
           "DataType" : "string",
           "Value" : "pi_sc_tp"
       },
       {
           "AttributeId" : "subject:preferences",
           "DataType" : "string",
           "Value" : "pi_co_pp"
       },
       {
           "AttributeId" : "subject:preferences",
           "DataType" : "string",
```

```
    "Value" : "pi_co_sp"
},
{
    "AttributeId" : "subject:preferences",
    "DataType" : "string",
    "Value" : "pi_co_tp"
},
{
    "AttributeId" : "subject:preferences",
    "DataType" : "string",
    "Value" : "ah_si_pp"
},
{
    "AttributeId" : "subject:preferences",
    "DataType" : "string",
    "Value" : "ah_si_sp"
},
{
    "AttributeId" : "subject:preferences",
    "DataType" : "string",
    "Value" : "ah_si_tp"
},
{
    "AttributeId" : "subject:preferences",
    "DataType" : "string",
    "Value" : "ah_sc_pp"
},
{
    "AttributeId" : "subject:preferences",
    "DataType" : "string",
    "Value" : "ah_sc_sp"
},
{
    "AttributeId" : "subject:preferences",
    "DataType" : "string",
    "Value" : "ah_sc_tp"
},
{
    "AttributeId" : "subject:preferences",
    "DataType" : "string",
    "Value" : "ah_co_pp"
},
{
    "AttributeId" : "subject:preferences",
    "DataType" : "string",
    "Value" : "ah_co_sp"
},
{
    "AttributeId" : "subject:preferences",
    "DataType" : "string",
```

```
        "Value" : "ah_co_tp"
    },
    {
        "AttributeId" : "subject:preferences",
        "DataType" : "string",
        "Value" : "rs_si_pp"
    },
    {
        "AttributeId" : "subject:preferences",
        "DataType" : "string",
        "Value" : "rs_si_sp"
    },
    {
        "AttributeId" : "subject:preferences",
        "DataType" : "string",
        "Value" : "rs_si_tp"
    },
    {
        "AttributeId" : "subject:preferences",
        "DataType" : "string",
        "Value" : "rs_sc_pp"
    },
    {
        "AttributeId" : "subject:preferences",
        "DataType" : "string",
        "Value" : "rs_sc_sp"
    },
    {
        "AttributeId" : "subject:preferences",
        "DataType" : "string",
        "Value" : "rs_sc_tp"
    },
    {
        "AttributeId" : "subject:preferences",
        "DataType" : "string",
        "Value" : "rs_co_pp"
    },
    {
        "AttributeId" : "subject:preferences",
        "DataType" : "string",
        "Value" : "rs_co_sp"
    },
    {
        "AttributeId" : "subject:preferences",
        "DataType" : "string",
        "Value" : "rs_co_tp"
    }
  ]

},
```

```
  "Action" : {
      "Attribute" : {

          "AttributeId" : "action:action-id",
          "DataType" : "string",
          "Value" : "visualizar"

      }

  },
  "Resource" : {
      "Attribute" : [{

          "AttributeId" : "resource:resource-id",
          "DataType" : "string",
          "Value" : "www.sitedecompras.com:userid:historico"
      },
      {
          "AttributeId" : "resource:finalidade",
          "DataType" : "string",
          "Value" : "SI"
      },
      {
          "AttributeId" : "resource:tipo-dado",
          "DataType" : "string",
          "Value" : "PI"
      },
      {
          "AttributeId" : "resource:tipo-dado",
          "DataType" : "string",
          "Value" : "AH"
      }

      ]

  }
}
 }
```

REQUISIÇÃO 2.2


```
{
	"Request" : {
		"AccessSubject" : {
			"Attribute" : [
				{
				"AttributeId" : "subject:subject-id",
				"DataType" : "string",
				"Value" : "PP"
				},
				{
				"AttributeId" : "subject:preferences",
				"DataType" : "string",
				"Value" : "ah_si_sp"
				},
				{
				"AttributeId" : "subject:preferences",
				"DataType" : "string",
				"Value" : "ah_si_tp"
				},
				{
				"AttributeId" : "subject:preferences",
				"DataType" : "string",
				"Value" : "ah_co_sp"
				},
				{
				"AttributeId" : "subject:preferences",
				"DataType" : "string",
				"Value" : "ah_co_tp"
				},
				{
				"AttributeId" : "subject:preferences",
				"DataType" : "string",
				"Value" : "rs_si_sp"
				},
				{
				"AttributeId" : "subject:preferences",
				"DataType" : "string",
				"Value" : "rs_si_tp"
				},
				{
				"AttributeId" : "subject:preferences",
				"DataType" : "string",
				"Value" : "rs_co_sp"
				},
				{
				"AttributeId" : "subject:preferences",
				"DataType" : "string",
```

```
                                    "Value" : "rs_co_tp"
                                }
                            ]

            },
    "Action" : {
        "Attribute" : {

            "AttributeId" : "action:action-id",
            "DataType" : "string",
            "Value" : "visualizar"

        }

    },
    "Resource" : {
        "Attribute" : [{

            "AttributeId" : "resource:resource-id",
            "DataType" : "string",
            "Value" : "www.sitedecompras.com:userid:historico"
        },
        {
            "AttributeId" : "resource:finalidade",
            "DataType" : "string",
            "Value" : "SI"
        },
        {
            "AttributeId" : "resource:tipo-dado",
            "DataType" : "string",
            "Value" : "PI"
        },
        {
            "AttributeId" : "resource:tipo-dado",
            "DataType" : "string",
            "Value" : "AH"
        }

        ]

    }
}
 }
```

REQUISIÇÃO 3.1.a


```
{
 "Request" : {
   "AccessSubject" : {
      "Attribute" : [
       {
          "AttributeId" : "subject:subject-id",
          "DataType" : "string",
          "Value" : "PP"
         },
         {
          "AttributeId" : "subject:preferences",
          "DataType" : "string",
          "Value" : "pi_si_pp"
       },
       {
          "AttributeId" : "subject:preferences",
          "DataType" : "string",
          "Value" : "pi_si_sp"
       },
       {
          "AttributeId" : "subject:preferences",
          "DataType" : "string",
          "Value" : "pi_si_tp"
       },
       {
          "AttributeId" : "subject:preferences",
          "DataType" : "string",
          "Value" : "pi_sc_pp"
       },
       {
          "AttributeId" : "subject:preferences",
          "DataType" : "string",
          "Value" : "pi_sc_sp"
       },
       {
          "AttributeId" : "subject:preferences",
          "DataType" : "string",
          "Value" : "pi_sc_tp"
       },
       {
          "AttributeId" : "subject:preferences",
          "DataType" : "string",
          "Value" : "pi_co_pp"
       },
       {
          "AttributeId" : "subject:preferences",
          "DataType" : "string",
          "Value" : "pi_co_sp"
```

```
},
{
    "AttributeId" : "subject:preferences",
    "DataType" : "string",
    "Value" : "pi_co_tp"
},
{
    "AttributeId" : "subject:preferences",
    "DataType" : "string",
    "Value" : "ah_si_pp"
},
{
    "AttributeId" : "subject:preferences",
    "DataType" : "string",
    "Value" : "ah_si_sp"
},
{
    "AttributeId" : "subject:preferences",
    "DataType" : "string",
    "Value" : "ah_si_tp"
},
{
    "AttributeId" : "subject:preferences",
    "DataType" : "string",
    "Value" : "ah_sc_pp"
},
{
    "AttributeId" : "subject:preferences",
    "DataType" : "string",
    "Value" : "ah_sc_sp"
},
{
    "AttributeId" : "subject:preferences",
    "DataType" : "string",
    "Value" : "ah_sc_tp"
},
{
    "AttributeId" : "subject:preferences",
    "DataType" : "string",
    "Value" : "ah_co_pp"
},
{
    "AttributeId" : "subject:preferences",
    "DataType" : "string",
    "Value" : "ah_co_sp"
},
{
    "AttributeId" : "subject:preferences",
    "DataType" : "string",
    "Value" : "ah_co_tp"
```

```
    },
    {
        "AttributeId" : "subject:preferences",
        "DataType" : "string",
        "Value" : "rs_si_pp"
    },
    {
        "AttributeId" : "subject:preferences",
        "DataType" : "string",
        "Value" : "rs_si_sp"
    },
    {
        "AttributeId" : "subject:preferences",
        "DataType" : "string",
        "Value" : "rs_si_tp"
    },
    {
        "AttributeId" : "subject:preferences",
        "DataType" : "string",
        "Value" : "rs_sc_pp"
    },
    {
        "AttributeId" : "subject:preferences",
        "DataType" : "string",
        "Value" : "rs_sc_sp"
    },
    {
        "AttributeId" : "subject:preferences",
        "DataType" : "string",
        "Value" : "rs_sc_tp"
    },
    {
        "AttributeId" : "subject:preferences",
        "DataType" : "string",
        "Value" : "rs_co_pp"
    },
    {
        "AttributeId" : "subject:preferences",
        "DataType" : "string",
        "Value" : "rs_co_sp"
    },
    {
        "AttributeId" : "subject:preferences",
        "DataType" : "string",
        "Value" : "rs_co_tp"
    }
   ]

},
"Action" : {
```

```
        "Attribute" : {

            "AttributeId" : "action:action-id",
            "DataType" : "string",
            "Value" : "coletar"
        }

    },
    "Resource" : {
        "Attribute" : [{

            "AttributeId" : "resource:resource-id",
            "DataType" : "string",
            "Value" : "www.sitedecompras.com:userid:historico:qtde_compradores"
        },
        {
            "AttributeId" : "resource:finalidade",
            "DataType" : "string",
            "Value" : "CO"
        },
        {
            "AttributeId" : "resource:tipo-dado",
            "DataType" : "string",
            "Value" : "PI"
        },
        {
            "AttributeId" : "resource:tipo-dado",
            "DataType" : "string",
            "Value" : "AH"
        },
        {
            "AttributeId" : "resource:tipo-dado",
            "DataType" : "string",
            "Value" : "RS"
        }

        ]

    }
}
 }
```

REQUISIÇÃO 3.1.b


```
{
 "Request" : {
   "AccessSubject" : {
      "Attribute" : [
       {
           "AttributeId" : "subject:subject-id",
           "DataType" : "string",
           "Value" : "SP"
          },
          {
           "AttributeId" : "subject:preferences",
           "DataType" : "string",
           "Value" : "pi_si_pp"
       },
       {
           "AttributeId" : "subject:preferences",
           "DataType" : "string",
           "Value" : "pi_si_sp"
       },
       {
           "AttributeId" : "subject:preferences",
           "DataType" : "string",
           "Value" : "pi_si_tp"
       },
       {
           "AttributeId" : "subject:preferences",
           "DataType" : "string",
           "Value" : "pi_sc_pp"
       },
       {
           "AttributeId" : "subject:preferences",
           "DataType" : "string",
           "Value" : "pi_sc_sp"
       },
       {
           "AttributeId" : "subject:preferences",
           "DataType" : "string",
           "Value" : "pi_sc_tp"
       },
       {
           "AttributeId" : "subject:preferences",
           "DataType" : "string",
           "Value" : "pi_co_pp"
       },
       {
           "AttributeId" : "subject:preferences",
           "DataType" : "string",
           "Value" : "pi_co_sp"
```

```
},
{
    "AttributeId" : "subject:preferences",
    "DataType" : "string",
    "Value" : "pi_co_tp"
},
{
    "AttributeId" : "subject:preferences",
    "DataType" : "string",
    "Value" : "ah_si_pp"
},
{
    "AttributeId" : "subject:preferences",
    "DataType" : "string",
    "Value" : "ah_si_sp"
},
{
    "AttributeId" : "subject:preferences",
    "DataType" : "string",
    "Value" : "ah_si_tp"
},
{
    "AttributeId" : "subject:preferences",
    "DataType" : "string",
    "Value" : "ah_sc_pp"
},
{
    "AttributeId" : "subject:preferences",
    "DataType" : "string",
    "Value" : "ah_sc_sp"
},
{
    "AttributeId" : "subject:preferences",
    "DataType" : "string",
    "Value" : "ah_sc_tp"
},
{
    "AttributeId" : "subject:preferences",
    "DataType" : "string",
    "Value" : "ah_co_pp"
},
{
    "AttributeId" : "subject:preferences",
    "DataType" : "string",
    "Value" : "ah_co_sp"
},
{
    "AttributeId" : "subject:preferences",
    "DataType" : "string",
    "Value" : "ah_co_tp"
```

```
    },
    {
        "AttributeId" : "subject:preferences",
        "DataType" : "string",
        "Value" : "rs_si_pp"
    },
    {
        "AttributeId" : "subject:preferences",
        "DataType" : "string",
        "Value" : "rs_si_sp"
    },
    {
        "AttributeId" : "subject:preferences",
        "DataType" : "string",
        "Value" : "rs_si_tp"
    },
    {
        "AttributeId" : "subject:preferences",
        "DataType" : "string",
        "Value" : "rs_sc_pp"
    },
    {
        "AttributeId" : "subject:preferences",
        "DataType" : "string",
        "Value" : "rs_sc_sp"
    },
    {
        "AttributeId" : "subject:preferences",
        "DataType" : "string",
        "Value" : "rs_sc_tp"
    },
    {
        "AttributeId" : "subject:preferences",
        "DataType" : "string",
        "Value" : "rs_co_pp"
    },
    {
        "AttributeId" : "subject:preferences",
        "DataType" : "string",
        "Value" : "rs_co_sp"
    },
    {
        "AttributeId" : "subject:preferences",
        "DataType" : "string",
        "Value" : "rs_co_tp"
    }
   ]

},
"Action" : {
```

```
      "Attribute" : {

          "AttributeId" : "action:action-id",
          "DataType" : "string",
          "Value" : "coletar"

      }

  },
  "Resource" : {
      "Attribute" : [{

          "AttributeId" : "resource:resource-id",
          "DataType" : "string",
          "Value":"www.sitedecompras.com:userid:historico:qtde_compradores"
      },
      {
          "AttributeId" : "resource:finalidade",
          "DataType" : "string",
          "Value" : "CO"
      },
      {
          "AttributeId" : "resource:tipo-dado",
          "DataType" : "string",
          "Value" : "PI"
      },
      {
          "AttributeId" : "resource:tipo-dado",
          "DataType" : "string",
          "Value" : "AH"
      },
      {
          "AttributeId" : "resource:tipo-dado",
          "DataType" : "string",
          "Value" : "RS"
      }

      ]

  }
}
 }
```

REQUISIÇÃO 3.2


```
{
        "Request" : {
                "AccessSubject" : {
                        "Attribute" : [
                                {
                                    "AttributeId" : "subject:subject-id",
                                    "DataType" : "string",
                                    "Value" : "PP"
                                },
                                {
                                    "AttributeId" : "subject:preferences",
                                    "DataType" : "string",
                                    "Value" : "ah_si_sp"
                                },
                                {
                                    "AttributeId" : "subject:preferences",
                                    "DataType" : "string",
                                    "Value" : "ah_si_tp"
                                },
                                {
                                    "AttributeId" : "subject:preferences",
                                    "DataType" : "string",
                                    "Value" : "ah_co_sp"
                                },
                                {
                                    "AttributeId" : "subject:preferences",
                                    "DataType" : "string",
                                    "Value" : "ah_co_tp"
                                },
                                {
                                    "AttributeId" : "subject:preferences",
                                    "DataType" : "string",
                                    "Value" : "rs_si_sp"
                                },
                                {
                                    "AttributeId" : "subject:preferences",
                                    "DataType" : "string",
                                    "Value" : "rs_si_tp"
                                },
                                {
                                    "AttributeId" : "subject:preferences",
                                    "DataType" : "string",
                                    "Value" : "rs_co_sp"
                                },
                                {
                                    "AttributeId" : "subject:preferences",
                                    "DataType" : "string",
```

```
                                    "Value" : "rs_co_tp"
                                }
                            ]

            },
    "Action" : {
        "Attribute" : {

            "AttributeId" : "action:action-id",
            "DataType" : "string",
            "Value" : "coletar"

        }

    },
    "Resource" : {
        "Attribute" : [{

            "AttributeId" : "resource:resource-id",
            "DataType" : "string",
            "Value":"www.sitedecompras.com:userid:historico:qtde_compradores"
        },
        {
            "AttributeId" : "resource:finalidade",
            "DataType" : "string",
            "Value" : "CO"
        },
        {
            "AttributeId" : "resource:tipo-dado",
            "DataType" : "string",
            "Value" : "PI"
        },
        {
            "AttributeId" : "resource:tipo-dado",
            "DataType" : "string",
            "Value" : "AH"
        },
        {
            "AttributeId" : "resource:tipo-dado",
            "DataType" : "string",
            "Value" : "RS"
        }

        ]

    }
}
 }
```

APÊNDICE D – EXEMPLOS DE PRIVACY TOKEN

TOKEN 1

{
 estrutura:{
   typ: JWT
   alg: SH256
 }
 atributos:{
  sub: Alice
  iss: http://openid.com.br
  aud: Cliente-1
  iat: 1000101010
  preferencias:{
    PI_SI_PP: 0
    PI_SI_SP: 0
    PI_SI_TP: 0
    PI_SC_PP: 0
    PI_SC_SP: 0
    PI_SC_TP: 0
    PI_CO_PP: 0
    PI_CO_SP: 0
    PI_CO_TP: 0
    PCP_SI_PP: 0
    PCP_SI_SP: 0
    PCP_SI_TP: 0
    PCP_SC_PP: 0
    PCP_SC_SP: 0
    PCP_SC_TP: 0
    PCP_CO_PP: 0
    PCP_CO_SP: 0
    PCP_CO_TP: 0
    LO_SI_PP: 0
    LO_SI_SP: 0
    LO_SI_TP: 0
    LO_SC_PP: 0
    LO_SC_SP: 0
    LO_SC_TP: 0
    LO_CO_PP: 0
    LO_CO_SP: 0
    LO_CO_TP: 0
    AH_SI_PP: 0
    AH_SI_SP: 1
    AH_SI_TP: 1
    AH_SC_PP: 0
    AH_SC_SP: 0
    AH_SC_TP: 0
    AH_CO_PP: 0
    AH_CO_SP: 1
    AH_CO_TP: 1

RS_SI_PP: 0
RS_SI_SP: 1
RS_SI_TP: 1
RS_SC_PP: 0
RS_SC_SP: 0
RS_SC_TP: 0
RS_CO_PP: 0
RS_CO_SP: 1
RS_CO_TP: 1
}
}
}

TOKEN 2

{
estrutura:{
typ: JWT
alg: SH256
}
atributos:{
sub: Alice
iss: http://openid.com.br
aud: Cliente-1
iat: 1000101010
preferencias:{
PI_SI_PP: 1
PI_SI_SP: 1
PI_SI_TP: 1
PI_SC_PP: 1
PI_SC_SP: 1
PI_SC_TP: 1
PI_CO_PP: 1
PI_CO_SP: 1
PI_CO_TP: 1
PCP_SI_PP: 0
PCP_SI_SP: 0
PCP_SI_TP: 0
PCP_SC_PP: 0
PCP_SC_SP: 0
PCP_SC_TP: 0
PCP_CO_PP: 0
PCP_CO_SP: 0
PCP_CO_TP: 0
LO_SI_PP: 0
LO_SI_SP: 0
LO_SI_TP: 0
LO_SC_PP: 0
LO_SC_SP: 0
LO_SC_TP: 0
LO_CO_PP: 0

```
    LO_CO_SP: 0
    LO_CO_TP: 0
    AH_SI_PP: 1
    AH_SI_SP: 1
    AH_SI_TP: 1
    AH_SC_PP: 1
    AH_SC_SP: 1
    AH_SC_TP: 1
    AH_CO_PP: 1
    AH_CO_SP: 1
    AH_CO_TP: 1
    RS_SI_PP: 1
    RS_SI_SP: 1
    RS_SI_TP: 1
    RS_SC_PP: 1
    RS_SC_SP: 1
    RS_SC_TP: 1
    RS_CO_PP: 1
    RS_CO_SP: 1
    RS_CO_TP: 1
  }
 }
}
```

APÊNDICE E – EXEMPLOS DE POLÍTICAS

POLÍTICA CASO DE USO 1

```
<?xml version="1.0" encoding="UTF-8"?>
<!--CONJUNTO DE POLITICA DE PRIVACIDADE -->
<PolicySet xmlns="urn:oasis:names:tc:xacml:3.0:core:schema:wd-17"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        PolicySetId="privacy-policy-set-caso-1"
        Version="1.0"
        PolicyCombiningAlgId="urn:oasis:names:tc:xacml:1.0:policy-combining-
algorithm:deny-overrides">


    <Target>

            <AnyOf>
                <AllOf>
                    <Match
MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
                        <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">www.sitedecompras.com:compra
s:produtoid</AttributeValue>
                        <AttributeDesignator MustBePresent="true"

Category="urn:oasis:names:tc:xacml:3.0:attribute-category:resource"
                                                AttributeId="resource:resource-id"

DataType="http://www.w3.org/2001/XMLSchema#string"/>
                    </Match>
                </AllOf>
            </AnyOf>
            <AnyOf>
                <AllOf>
                    <Match
MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
                        <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">comprar</AttributeValue>
                        <AttributeDesignator MustBePresent="true"

Category="urn:oasis:names:tc:xacml:3.0:attribute-category:action"
                                                AttributeId="action:action-id"

DataType="http://www.w3.org/2001/XMLSchema#string"/>
                    </Match>
                </AllOf>
            </AnyOf>
    </Target>
 <!-- POLITICA DE TOKEN DE PRIVACIDADE -->
```

```
<Policy PolicyId="policy:privacytoken"
        Version="1.0"
        RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-combining-
algorithm:permit-overrides">

        <Target>

        </Target>

        <Rule RuleId="rule-1:privacytoken-01" Effect="Permit">

                <Condition >
                        <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:any-of-any">
                                <Function
FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-equal"/>

                                <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-bag">
                                        <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">pi_sc_pp</AttributeValue>
                                </Apply>
                                <AttributeDesignator
DataType="http://www.w3.org/2001/XMLSchema#string"
                                                MustBePresent="false"

Category="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject"

AttributeId="subject:preferences"/>

                        </Apply>
                </Condition>
        </Rule>
        <Rule RuleId="rule-2:deny" Effect="Deny"></Rule>

</Policy>

 <!-- POLITICA DE TIPO DE DADOS -->
<Policy PolicyId="policy:tipodedados"
        Version="1.0"
        RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-combining-
algorithm:permit-overrides">

        <Target/>
        <Rule RuleId="rule-1:tipodedados-01" Effect="Permit">
                <Target/>
                <Condition >
                        <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:any-of-any">
```

```
                                        <Function
FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-equal"/>

                                        <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-bag">
                                                <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">PI</AttributeValue>
                                        </Apply>
                                        <AttributeDesignator
DataType="http://www.w3.org/2001/XMLSchema#string"
                                                        MustBePresent="false"

Category="urn:oasis:names:tc:xacml:3.0:attribute-category:resource"
                                                        AttributeId="resource:tipo-
dado"/>

                                </Apply>

                        </Condition>
                </Rule>
                <Rule RuleId="rule-2:deny" Effect="Deny"></Rule>
        </Policy>

<!-- POLITICA DE FINALIDADE -->
        <Policy PolicyId="policy:finalidade"
                        Version="1.0"
                        RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-combining-
algorithm:permit-overrides">

                <Target/>

                <Rule RuleId="rule-1:finalidade-01" Effect="Permit">
                        <Target/>
                        <Condition >

                                <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:any-of-any">
                                        <Function
FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-equal"/>

                                        <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-bag">
                                                <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">SC</AttributeValue>
                                        </Apply>
                                        <AttributeDesignator
DataType="http://www.w3.org/2001/XMLSchema#string"
                                                        MustBePresent="false"

Category="urn:oasis:names:tc:xacml:3.0:attribute-category:resource"
```

```
AttributeId="resource:finalidade"/>
                                </Apply>
                        </Condition>
                </Rule>
                <Rule RuleId="rule-2:deny" Effect="Deny"></Rule>
        </Policy>

        <Policy PolicyId="policy:beneficiario"
                        Version="1.0"
                        RuleCombiningAlgId="urn:oasis:names:tc:xacml:3.0:rule-combining-
algorithm:permit-overrides">

                <Target/>
                <Rule RuleId="rule-1:beneficiario-01" Effect="Permit">
                        <Target/>
                        <Condition >
                                <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:any-of-any">
                                        <Function
FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-equal"/>

                                        <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-bag">
                                                <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">PP</AttributeValue>
                                        </Apply>
                                        <AttributeDesignator
DataType="http://www.w3.org/2001/XMLSchema#string"
                                                                MustBePresent="false"

        Category="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject"
                                                                AttributeId="subject:subject-id"/>


                                </Apply>
                        </Condition>
                </Rule>
                <Rule RuleId="rule-2:deny" Effect="Deny"></Rule>
        </Policy>


</PolicySet>
```

POLÍTICA CASO DE USO 2

```
<?xml version="1.0" encoding="UTF-8"?>
<!--CONJUNTO DE POLITICA DE PRIVACIDADE -->
<PolicySet xmlns="urn:oasis:names:tc:xacml:3.0:core:schema:wd-17"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        PolicySetId="privacy-policy-set-caso-2"
        Version="1.0"
        PolicyCombiningAlgId="urn:oasis:names:tc:xacml:1.0:policy-combining-
algorithm:deny-overrides">


    <Target>
        <AnyOf>
            <AllOf>
                <Match
MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
                    <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">www.sitedecompras.com:userid:
historico</AttributeValue>
                    <AttributeDesignator MustBePresent="true"

Category="urn:oasis:names:tc:xacml:3.0:attribute-category:resource"
                                        AttributeId="resource:resource-id"

DataType="http://www.w3.org/2001/XMLSchema#string"/>
                </Match>
            </AllOf>
        </AnyOf>
        <AnyOf>
            <AllOf>
                <Match
MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
                    <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">visualizar</AttributeValue>
                    <AttributeDesignator MustBePresent="true"

Category="urn:oasis:names:tc:xacml:3.0:attribute-category:action"
                                        AttributeId="action:action-id"

DataType="http://www.w3.org/2001/XMLSchema#string"/>
                </Match>
            </AllOf>
        </AnyOf>
    </Target>
```

```
<!-- POLITICA DE PRIVACY TOKEN -->
        <Policy PolicyId="policy:privacytoken"
                        Version="1.0"
                        RuleCombiningAlgId="urn:oasis:names:tc:xacml:3.0:rule-combining-
algorithm:permit-overrides">

                <Target/>
                <Rule RuleId="rule-1:privacytoken-01" Effect="Permit">
                        <Target/>
                        <Condition >
                                <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:and">
                                        <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:any-of-any">
                                                <Function
FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-equal"/>

                                                <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-bag">
                                                        <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">pi_si_sp</AttributeValue>
                                                </Apply>
                                                <AttributeDesignator
DataType="http://www.w3.org/2001/XMLSchema#string"
                                                                MustBePresent="false"

Category="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject"

AttributeId="subject:preferences"/>


                                        </Apply>

                                        <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:any-of-any">
                                                <Function
FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-equal"/>

                                                <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-bag">
                                                        <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">ah_si_sp</AttributeValue>
                                                </Apply>
                                                <AttributeDesignator
DataType="http://www.w3.org/2001/XMLSchema#string"
                                                                MustBePresent="false"

Category="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject"

AttributeId="subject:preferences"/>
```

```
                                        </Apply>
                                    </Apply>
                                </Condition>
                        </Rule>
                        <Rule RuleId="rule-2:deny" Effect="Deny"></Rule>
                </Policy>

                 <!-- POLITICA DE TIPO DE DADOS -->
                <Policy PolicyId="policy:tipodedados"
                                Version="1.0"
                                RuleCombiningAlgId="urn:oasis:names:tc:xacml:3.0:rule-combining-
algorithm:permit-overrides">

                        <Target/>
                        <Rule RuleId="rule-1:tipodedados-01" Effect="Permit">
                                <Target/>
                                <Condition >
                                        <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:and">
                                                <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:any-of-any">
                                                        <Function
FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-equal"/>

                                                        <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-bag">
                                                                <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">PI</AttributeValue>
                                                        </Apply>
                                                        <AttributeDesignator
DataType="http://www.w3.org/2001/XMLSchema#string"
                                                                MustBePresent="false"

Category="urn:oasis:names:tc:xacml:3.0:attribute-category:resource"
                                                                AttributeId="resource:tipo-
dado"/>

                                                </Apply>
                                                <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:any-of-any">
                                                        <Function
FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-equal"/>

                                                        <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-bag">
                                                                <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">AH</AttributeValue>
                                                        </Apply>
```

```
                                        <AttributeDesignator
DataType="http://www.w3.org/2001/XMLSchema#string"
                                                MustBePresent="false"
Category="urn:oasis:names:tc:xacml:3.0:attribute-category:resource"
                                                AttributeId="resource:tipo-
dado"/>

                                    </Apply>

                                </Apply>
                            </Condition>
                        </Rule>
                        <Rule RuleId="rule-2:deny" Effect="Deny"></Rule>
                </Policy>


<!-- POLITICA DE FINALIDADE -->
                <Policy PolicyId="policy:finalidade"
                            Version="1.0"
                            RuleCombiningAlgId="urn:oasis:names:tc:xacml:3.0:rule-combining-
algorithm:permit-overrides">

                        <Target/>

                        <Rule RuleId="rule-1:finalidade-01" Effect="Permit">
                            <Target/>
                            <Condition >
                                <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:any-of-any">
                                    <Function
FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-equal"/>

                                    <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-bag">
                                        <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">SI</AttributeValue>
                                    </Apply>
                                    <AttributeDesignator
DataType="http://www.w3.org/2001/XMLSchema#string"
                                                MustBePresent="false"
Category="urn:oasis:names:tc:xacml:3.0:attribute-category:resource"
AttributeId="resource:finalidade"/>
                                </Apply>
                            </Condition>
                        </Rule>
                        <Rule RuleId="rule-2:deny" Effect="Deny"></Rule>
                </Policy>
```

```
 <!-- POLITICA DE BENEFICIARIO -->
        <Policy PolicyId="policy:beneficiario"
                        Version="1.0"
                        RuleCombiningAlgId="urn:oasis:names:tc:xacml:3.0:rule-combining-
algorithm:permit-overrides">

                <Target/>
                <Rule RuleId="rule-1:beneficiario-01" Effect="Permit">
                        <Target/>
                        <Condition >
                                <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:any-of-any">
                                        <Function
FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-equal"/>

                                        <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-bag">
                                                <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">SP</AttributeValue>
                                        </Apply>
                                        <AttributeDesignator
DataType="http://www.w3.org/2001/XMLSchema#string"
                                                                MustBePresent="false"

        Category="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject"
                                                        AttributeId="subject:subject-id"/>


                                </Apply>
                        </Condition>
                </Rule>
                <Rule RuleId="rule-2:deny" Effect="Deny"></Rule>
        </Policy>

</PolicySet>
```

POLÍTICA CASO DE USO 3

```
<?xml version="1.0" encoding="UTF-8"?>
<!--CONJUNTO DE POLITICA DE PRIVACIDADE -->
<PolicySet xmlns="urn:oasis:names:tc:xacml:3.0:core:schema:wd-17"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        PolicySetId="privacy-policy-set-caso-3"
        Version="1.0"
        PolicyCombiningAlgId="urn:oasis:names:tc:xacml:1.0:policy-combining-
algorithm:deny-overrides">


    <Target>
        <AnyOf>
            <AllOf>
                <Match
MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
                    <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">www.sitedecompras.com:userid:
historico:qtde_compradores</AttributeValue>
                    <AttributeDesignator MustBePresent="true"

Category="urn:oasis:names:tc:xacml:3.0:attribute-category:resource"
                                                AttributeId="resource:resource-id"

DataType="http://www.w3.org/2001/XMLSchema#string"/>
                </Match>
            </AllOf>
        </AnyOf>
        <AnyOf>
            <AllOf>
                <Match
MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
                    <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">coletar</AttributeValue>
                    <AttributeDesignator MustBePresent="true"

Category="urn:oasis:names:tc:xacml:3.0:attribute-category:action"
                                                AttributeId="action:action-id"

DataType="http://www.w3.org/2001/XMLSchema#string"/>
                </Match>
            </AllOf>
        </AnyOf>
    </Target>
```

```
<!-- POLITICA DE PRIVACY TOKEN -->
        <Policy PolicyId="policy:privacytoken"
                        Version="1.0"
                        RuleCombiningAlgId="urn:oasis:names:tc:xacml:3.0:rule-combining-
algorithm:permit-overrides">

                <Target/>
                <Rule RuleId="rule-1:privacytoken-01" Effect="Permit">
                        <Target/>
                        <Condition >
                                <Apply  FunctionId="urn:oasis:names:tc:xacml:1.0:function:or">

                                        <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:and">
                                                <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:any-of-any"> <!-- pi_co_pp -->
                                                <Function
FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-equal"/>

                                                <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-bag">
                                                        <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">pi_co_pp</AttributeValue>
                                                </Apply>
                                                <AttributeDesignator
DataType="http://www.w3.org/2001/XMLSchema#string"

MustBePresent="false"

Category="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject"

AttributeId="subject:preferences"/>


                                        </Apply>

                                        <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:any-of-any">  <!-- ah_co_pp -->
                                                <Function
FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-equal"/>

                                                <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-bag">
                                                        <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">ah_co_pp</AttributeValue>
                                                </Apply>
                                                <AttributeDesignator
DataType="http://www.w3.org/2001/XMLSchema#string"

MustBePresent="false"
```

```
Category="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject"

AttributeId="subject:preferences"/>


                                        </Apply>

                                        <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:any-of-any">  <!-- rs_co_pp -->
                                                <Function
FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-equal"/>

                                                <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-bag">
                                                        <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">rs_co_pp</AttributeValue>
                                                </Apply>
                                                <AttributeDesignator
DataType="http://www.w3.org/2001/XMLSchema#string"

MustBePresent="false"

Category="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject"

AttributeId="subject:preferences"/>
                                        </Apply>
                                        <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:any-of-any"><!-- PP -->
                                                <Function
FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-equal"/>

                                                <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-bag">
                                                        <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">PP</AttributeValue>
                                                </Apply>
                                                <AttributeDesignator
DataType="http://www.w3.org/2001/XMLSchema#string"

        MustBePresent="false"

        Category="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject"

        AttributeId="subject:subject-id"/>

                                        </Apply>

                                </Apply>
```

```
                                        <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:and">
                                                <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:any-of-any">  <!-- pi_co_sp -->
                                                <Function
FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-equal"/>

                                                <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-bag">
                                                        <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">pi_co_sp</AttributeValue>
                                                </Apply>
                                                <AttributeDesignator
DataType="http://www.w3.org/2001/XMLSchema#string"

MustBePresent="false"

Category="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject"

AttributeId="subject:preferences"/>


                                                </Apply>

                                                <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:any-of-any">   <!-- ah_co_sp -->
                                                <Function
FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-equal"/>

                                                <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-bag">
                                                        <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">ah_co_sp</AttributeValue>
                                                </Apply>
                                                <AttributeDesignator
DataType="http://www.w3.org/2001/XMLSchema#string"

MustBePresent="false"

Category="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject"

AttributeId="subject:preferences"/>


                                                </Apply>

                                                <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:any-of-any">  <!-- rs_co_sp -->
                                                <Function
FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-equal"/>
```

```
                                                    <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-bag">
                                                        <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">rs_co_sp</AttributeValue>
                                                    </Apply>
                                                    <AttributeDesignator
DataType="http://www.w3.org/2001/XMLSchema#string"

MustBePresent="false"

Category="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject"

AttributeId="subject:preferences"/>
                                                </Apply>
                                                <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:any-of-any"><!-- SP -->
                                                    <Function
FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-equal"/>
                                                    <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-bag">
                                                        <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">SP</AttributeValue>
                                                    </Apply>
                                                    <AttributeDesignator
DataType="http://www.w3.org/2001/XMLSchema#string"

        MustBePresent="false"

        Category="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject"

        AttributeId="subject:subject-id"/>
                                                </Apply>

                                            </Apply>
                                        </Apply>
                                    </Condition>
                                </Rule>
                                <Rule RuleId="rule-2:deny" Effect="Deny"></Rule>
        </Policy>

         <!-- POLITICA DE TIPO DE DADOS -->
        <Policy PolicyId="policy:tipodedados"
                        Version="1.0"
                        RuleCombiningAlgId="urn:oasis:names:tc:xacml:3.0:rule-combining-
algorithm:permit-overrides">

                <Target/>
                <Rule RuleId="rule-1:tipodedados-01" Effect="Permit">
                        <Target/>
                        <Condition >
```

```
                        <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:and">
                            <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:any-of-any">
                                <Function
FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-equal"/>

                                <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-bag">
                                    <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">PI</AttributeValue>
                                </Apply>
                                <AttributeDesignator
DataType="http://www.w3.org/2001/XMLSchema#string"
                                    MustBePresent="false"

Category="urn:oasis:names:tc:xacml:3.0:attribute-category:resource"
                                    AttributeId="resource:tipo-
dado"/>

                            </Apply>
                            <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:any-of-any">
                                <Function
FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-equal"/>

                                <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-bag">
                                    <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">AH</AttributeValue>
                                </Apply>
                                <AttributeDesignator
DataType="http://www.w3.org/2001/XMLSchema#string"
                                    MustBePresent="false"

Category="urn:oasis:names:tc:xacml:3.0:attribute-category:resource"
                                    AttributeId="resource:tipo-
dado"/>

                            </Apply>

                            <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:any-of-any">
                                <Function
FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-equal"/>

                                <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-bag">
                                    <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">RS</AttributeValue>
```

```
                                        </Apply>
                                        <AttributeDesignator
DataType="http://www.w3.org/2001/XMLSchema#string"
                                                MustBePresent="false"

Category="urn:oasis:names:tc:xacml:3.0:attribute-category:resource"
                                                AttributeId="resource:tipo-
dado"/>

                                    </Apply>
                                </Apply>
                            </Condition>
                    </Rule>
                    <Rule RuleId="rule-2:deny" Effect="Deny"></Rule>
            </Policy>


<!-- POLITICA DE FINALIDADE -->
            <Policy PolicyId="policy:finalidade"
                            Version="1.0"
                            RuleCombiningAlgId="urn:oasis:names:tc:xacml:3.0:rule-combining-
algorithm:permit-overrides">

                    <Target/>

                    <Rule RuleId="rule-1:finalidade-01" Effect="Permit">
                            <Target/>
                            <Condition >
                                    <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:any-of-any">
                                        <Function
FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-equal"/>

                                        <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-bag">
                                            <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">CO</AttributeValue>
                                        </Apply>
                                        <AttributeDesignator
DataType="http://www.w3.org/2001/XMLSchema#string"
                                                MustBePresent="false"

Category="urn:oasis:names:tc:xacml:3.0:attribute-category:resource"

AttributeId="resource:finalidade"/>
                                    </Apply>
                            </Condition>
                    </Rule>
                    <Rule RuleId="rule-2:deny" Effect="Deny"></Rule>
            </Policy>
```

```
<!-- POLITICA DE BENEFICIARIO -->
        <Policy PolicyId="policy:beneficiario"
                        Version="1.0"
                        RuleCombiningAlgId="urn:oasis:names:tc:xacml:3.0:rule-combining-
algorithm:permit-overrides">

                <Target/>
                <Rule RuleId="rule-1:beneficiario-01" Effect="Permit">
                        <Target/>
                        <Condition >
                                <Apply  FunctionId="urn:oasis:names:tc:xacml:1.0:function:or">

                                        <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:any-of-any">
                                                <Function
FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-equal"/>

                                                <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-bag">
                                                        <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">PP</AttributeValue>
                                                </Apply>
                                                <AttributeDesignator
DataType="http://www.w3.org/2001/XMLSchema#string"
                                                                MustBePresent="false"

        Category="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject"
                                                                AttributeId="subject:subject-
id"/>

                                        </Apply>

                                        <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:any-of-any">
                                                <Function
FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-equal"/>

                                                <Apply
FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-bag">
                                                        <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">SP</AttributeValue>
                                                </Apply>
                                                <AttributeDesignator
DataType="http://www.w3.org/2001/XMLSchema#string"
                                                                MustBePresent="false"

        Category="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject"
                                                                AttributeId="subject:subject-
id"/>
```

```
                                        </Apply>

                                    </Apply>
                                </Condition>
                            </Rule>
                            <Rule RuleId="rule-2:deny" Effect="Deny"></Rule>
                        </Policy>

</PolicySet>
```

Desenvolvimento e uso de um formato de políticas de privacidade no controle de acesso.

André A. Vargas, Carla M. Westphall, Jorge Werner, Gerson L. Camillo, Maria E. Villareal

Curso de Bacharelado em Ciências da Computação – Universidade Federal de Santa Catarina (UFSC) – Campus de Florianópolis
88040-900 – Florianópolis – SC– Brasil

Julho de 2017

Abstract. *Many web applications that offer services to people or businesses on the Internet need data to identify these entities or perform services in an automated and more practical way. These applications have the duty to protect the access to this data and the responsible for this are access control systems. But we often make data available in cloud environments and we end up not knowing what data is collected, how it is used, or who will use it. Privacy has been a challenge for information security, so this work proposes to develop and use privacy policies through access control languages and using preference data defined by the owner of this information.*

Resumo. *Muitas aplicações na web que oferecem serviços á pessoas ou empresas na Internet precisam de dados para identificar estas entidades ou realizar serviços de forma automatizada e mais prática. Estas aplicações tem o dever proteger o acesso a esses dados e o responsável por isso são sistemas de controle de acesso. Porém muitas vezes disponibilizamos dados em ambientes em nuvem e acabamos não sabendo quais dados são coletados, como são utilizados ou quem fará uso deles. A privacidade tem sido um desafio para a segurança da informação, portanto este trabalho propõe desenvolver e utilizar políticas de privacidade através das linguagens de controle de acesso e utilizando dados de preferencias definidas pelo dono dessas informações.*

1. Introdução

O número de aplicações e serviços tem crescido bastante na Internet. Isso ocorre devido ao grande numero de pessoas que tem acesso à internet, de forma cada vez mais fácil. Com a chegada dos dispositivos móveis como *smartphones* e *tablets* houve uma verdadeira explosão de usuários e por consequência serviços disponibilizados na *Web.*

A computação em nuvem está modificando a forma de usar a Internet já que os serviços e recursos podem ser utilizados sob demanda, de acordo com a necessidade dos usuários. A computação em nuvem pode oferecer poder de processamento, oferecer serviços de software e de armazenamento ou mesmo uma infraestrutura de dados distribuídos.

Porém para que seja possível usufruir desses serviços se faz necessário disponibilizar dados pessoais para que exista uma maneira de identificar um usuário e

para automatizar estes serviços. Sem uma limitação de uso, divulgação, manipulação desses dados, pode-se ocorrer uso malicioso das informações dos usuários.

A necessidade de possuir a privacidade é que fez com que políticas de privacidade fossem necessárias para estas entidades, a fim de garantir ao usuário segurança e confiança na entidade que provêm serviços e declarando os fins que terão os dados disponibilizados. Para isso é necessário que o usuário tenha ciência do uso de seus dados e para onde vão, além de ter controle do uso dos próprios dados em um determinado sistema. Os usuários também têm preocupações tanto quanto empresas que necessitam coletar dados para prestar serviços, já que nem sempre sabem para qual fim, quem utiliza e quais são os dados utilizados.

A proposta deste trabalho visa criar políticas com regras de controle de acesso visando conservar a privacidade dos dados de um determinado usuário. Estas regras terão como alvo os atributos e dados disponibilizados por um *token* de atributos de privacidade, que terá a tarefa de selecionar as preferências de uso de dados que já foram devidamente agrupados. Assim só pode-se ter acesso ao recurso com os devidos atributos se ás preferências dadas por este *token* forem respeitadas. Ao fim, existirá um sistema que simulará um sistema que gerencia o controle de acesso, mais especificamente autorização, para testar essas políticas.

2. Privacidade

A privacidade é alvo de grande preocupação em computação em nuvem e sistemas Web. Essa preocupação existe, pois grande parte dos sistemas na web necessita utilizar dados de seus usuários para identificar e prestar serviços de forma adequada e aceitável e precisam seguir uma legislação aplicada por seu país para privacidade. Os usuários também têm preocupações tanto quanto empresas que necessitam coletar dados para prestar serviços, já que nem sempre sabem para qual fim, quem utiliza e quais são os dados utilizados.

Existem muitas fontes que definem de sua própria maneira o conceito de privacidade, pois esta pode ser observada de diversos pontos de vista e contextualizada em diferentes áreas. Para Alan Westin (1967) citado por (EDUCAUSE, 2013) privacidade é o controle dos dados pessoais que os usuários possuem. Portanto a tomada de decisão e ter consciência de como são manipuladas as informações devem ser do usuário proprietário ou de quem tem sua autorização.

Alguns dos termos doravante são definidos a norma ISO 29100 (2011), as informações podem ser definidas nos seguintes tipos:

- **Informações de identificação pessoal (PII):** informação que pode ser usada para identificar uma entidade,
- **Preferências de privacidade:** São dados que o usuário define para usar em conjunto a uma política a fim de proteger os dados escolhidos pelo usuário.
- **Controlador de dados:** parte que é competente para decidir sobre o conteúdo e o uso de dados pessoais.

A privacidade é alvo de grande preocupação em computação em nuvem e sistemas Web. Essa preocupação existe, pois grande parte dos sistemas na *Web* necessita utilizar dados de seus usuários para identificar e prestar serviços de forma

adequada e aceitável e precisam seguir uma legislação aplicada por seu país para privacidade.

Com o crescente desenvolvimento de sistemas computacionais existente, esses princípios acabaram por ser definidos como um guia regido por leis.

2.1. Princípios de privacidade

Com o crescente desenvolvimento de sistemas computacionais existente, esses princípios acabaram por ser definidos como um guia regido por leis. Os princípios segundo OECD (2013) são os seguintes:

1. **Princípio de abertura:** Política geral de abertura sobre desenvolvimentos, práticas e políticas em relação aos dados pessoais.
2. **Qualidade dos dados:** Os dados pessoais devem ser relevantes para os fins para os quais estão sendo usados, e deve ser preciso, completo e mantido atualizado.
3. **Princípio de participação individual:** Os indivíduos tem o direito de obter a confirmação de se o controlador de dados tem ou não os dados relacionados a eles de forma clara.
4. **Princípio da especificação do propósito:** Os fins para os quais os dados pessoais são coletados.
5. **Limitação de coleta:** Limitação para a coleta de dados pessoais e quaisquer dados obtidos por meios legais, com o conhecimento ou o consentimento da pessoa em causa.
6. **Princípio de limitação de uso:** Limitação da divulgação, disponibilização ou de outra forma de uso dos dados pessoais para fins diferentes dos especificados no acordo com exceção do consentimento da pessoa.
7. **Salvaguardas de segurança:** Os dados pessoais devem ser protegidos por garantias de segurança razoáveis contra riscos.
8. **Responsabilidade:** Um controlador de dados deve ser responsável pelo cumprimento das medidas que dão efeito aos princípios acima mencionados.

2.2. Políticas de privacidade

Políticas de privacidade referem-se a políticas implementadas por uma organização sobre o uso de PII pela organização. Bertino e Takahashi (2011).

Pode ser definida como um conjunto de regras criado por uma entidade para determinar quais outras entidades ou serviços podem ter acesso as suas informações e que ações podem ser realizadas para este fim. O importante é que o usuário dono do dado esteja consciente dos dados usados, dos destinos e destinatários e finalidades de uso além do controle sobre essas preferências ao usuário. Essas políticas são a maneira como o controle de acesso pode definir as proteções para os usuários.

3. Gerência de identidades

Com a crescente quantidade de aplicações desenvolvidas para a *Web*, houve uma necessidade de criação de uma maneira segura e inteligente de gerir as informações oriundas de usuários para prestação de serviços.

Os sistemas de gerenciamento de identidade são os frameworks, que permitem aos usuários gerenciar adequadamente suas PIIs (Informações de identificação pessoal). Assim, eles permitem que os usuários acessem recursos e serviços usando dados de identificação armazenados em provedores de identidade, dos quais um subconjunto dos atributos de identificação pode ser divulgado aos provedores de serviços (Werner et. al, 2017).

3.1. Conceitos e componentes de gerenciamento de identidades (IdM)

A ISO/IEC 24760-1 (2011) define identidade como um conjunto de atributos relacionados a uma entidade. Essa entidade pode vir a possuir uma ou mais dessas identidades e diversas entidades podem ter a mesma identidade. Esses dados podem ser de qualquer natureza, desde que possuam alguma relação de forma direta com usuário.

Essas informações são imprescindíveis para ajudar resolver algumas tarefas de sistemas sem a necessidade de uma ação manual e presencial do usuário. Portanto isso acaba por evitar questionamentos sucessivos ao usuário sobre uma determinada ação que necessitaria de informações para realizá-la e deixa os sistemas mais dinâmicos.

Para que um sistema de gerenciamento de identidades possam exercer suas funções é necessário ter atores ou elementos o constituam. Cada parte ou elemento tem seu papel dentro do IdM:

- **Usuário:** entidade com identidade que deseja acessar um serviço ou recurso do sistema;
- **Provedor de Identidades:** está é a parte que é responsável por armazenar e gerenciar as informações sobre os usuários.
- **Provedor de Serviços:** parte que oferece serviços a usuários autorizados.

Esses componentes interagem entre si de acordo com modelos de IdM. Um dos modelos mais utilizados são os modelos federados, o qual é utilizado como padrão em sistemas de gerenciamento de identidade como OpenID Connect.

4. Controle de acesso.

Em sistemas de gerenciamento de identidade e acesso (IAM), provedores de identidade e provedores de serviços são geralmente componentes IdM que lidam com autenticação e autenticação. (Rosset, 2004) O controle de acesso é um monitor de referência que controla o acesso de um usuário a recursos.

Um sistema de controle de acesso é baseado praticamente em dois conceitos: mecanismo de controle de acesso (geralmente políticas) e modelos de controle de acesso.

O modelo utilizado neste trabalho é o controle de acesso baseado em atributos (ABAC), que é modelo de lógica que controla o acesso a objetos, avaliando regras em

relação aos atributos de entidades (sujeito e objeto), ações e o ambiente relevante para uma solicitação. (Hu et. al., 2017).

Neste trabalho este modelo possui maior relevância por ser o modelo a qual foi baseado arquitetura da especificação XACML. Essa especificação também possui uma linguagem de política baseado em XML.

Requisições no contexto de controle de acesso são mensagens emitidas por alguma aplicação de pedido de autorização para acesso aos recursos ou atributos destes para realização de alguma ação por algum sujeito em um determinado ambiente.

Response consiste em um ou mais resultados após a avaliação da política. No caso desse trabalho a resposta é sempre única e é no formato JSON. Nela vem o resultado enviado podendo ser *Permit* para permissão, *Deny* para negação e no caso de um erro *Indeterminate*.

Política de controle de acesso é um exemplo dos mecanismos que como o próprio nome já diz, realiza o controle de acesso a recursos através de uma ação de um sujeito dando a autorização ou não de acordo com regras bem definidas nela estabelecida (Silva et al , 2014). Existem diversas linguagens que podem ser utilizadas para descrever uma política em um sistema de controle de acesso, uma delas é a linguagem XACML.

4.1. XACML

Linguagem XACML é uma linguagem é uma proposta do OASIS para modelar, armazenar e distribuir políticas descritivas de controle de acesso que utiliza o formato do XML, porém com a finalidade de tratar o controle de acesso, realizando consultas que verificam a permissão de uma ação realizada para acessar um determinado recurso.

As duas estruturas que podem iniciar um documento de política e possui o cabeçalho em sua composição é a ***Policy*** e ***PolicySe****t*. Em sua composição precisa conter um algoritmo de combinação e um identificador da politica ou conjunto de politicas.

Rule: é a regra responsável por retornar para a política uma decisão de acordo após avaliações feitas em seu escopo com atributos passados por requisição e dados pré-definidos no documento através de condições atribuídas a regra.

Target:é o elemento usado para que o PDP possa localizar a política ou regra, através da comparação do valor e tipo de dado definido por atributo contido na categoria a qual ele pertence. As categorias padrões são as mesmas que existem na requisição, *Subject*, *Action* e *Resource*.

Algoritmos de combinação: são algoritmos que são declarados, mas estruturas de políticas ou conjunto de políticas para combinar a decisão e as obrigações de múltiplas políticas ou regras, a fim de escolher como aplicar o acesso, as politicas e conjunto de politicas possuem alguns algoritmos de combinação de politicas ou de regras consecutivamente.

Para que seja possível interpretar a linguagem e as requisições e gerar respostas é necessária uma arquitetura para realizar o controle de acesso, onde cada componente dessa arquitetura tem um papel definido.

- **PDP:** O ponto de decisão política avalia as políticas e processa uma decisão de autorização de acesso a um recurso.

- **PEP:** Ponto de aplicação de política é responsável por receber as requisições e enviar para o PDP.
- **PIP:** o ponto de informações de política atua como uma fonte aonde é possível coletar os valores dos atributos de um sujeito, recursos.
- **PAP:** o ponto de administração da política é a entidade do sistema responsável por criar, gerenciar e armazenar as políticas.

O fluxo simplificado funciona assim. O PEP cria a requisição de acordo com a solicitação realizada pelo usuário. O PIP adiciona informações do usuário no escopo da requisição caso necessário. O PAP disponibiliza as políticas para que o PDP possa avaliar e fazer a decisão de acesso de acordo com elas. O PDP retorna para o PEP a decisão que por sua vez informa o usuário através da resposta (*response*) se obteve êxito em acessar o recurso desejado.

5. Trabalhos relacionados.

Mont (2011) propõe e trata confidencialidade no controle de acesso sobre os dados pessoais e sensíveis voltados para empresas. Trata especificamente o problema da aplicação de políticas de privacidade em dados pessoais armazenados em uma ampla variedade de repositórios de dados dentro das empresas, podendo ter os dados pessoais acessados por diferentes tipos de solicitantes.

O trabalho proposto por Villareal (2017) apresenta um modelo de definição de privacidade para sistemas de gerenciamento de identidade, o qual permite a um usuários cadastrados em um IdP configurar suas preferências de privacidade de acordo com uma série perfis padrões oferecidos pré-definidos ou personalizado por ele, gerando uma estrutura nomeada pela autora de *token* de privacidade.

A proposta de Camillo et. al (2017) , é uma arquitetura que permite um individuo acessar algum recurso ou e filtrando os dados pessoais desnecessários para tal operação. Para isso as políticas de controle de acesso são enviadas para o IdP tirando a responsabilidade de fazer a decisão do SP.

O trabalho de Werner e Westphall (2016), propõe um modelo de gestão de identidade com privacidade para ambientes de nuvem usando o OpenID Connect como IdM com objetivo de reduzir os riscos de violação de privacidade.

6. Desenvolvimento Prático

6.1. Proposta

Para obter uma permissão de acesso é preciso que sujeitos (usuários) realizem ações para ter acesso a recursos. Porém as ações nem sempre utilizam todos os dados coletados e esses dados muitas vezes são sensíveis. O dono dos dados podem não saber quais dados estão sendo usados, nem para qual finalidade e quem fará uso desses dados.

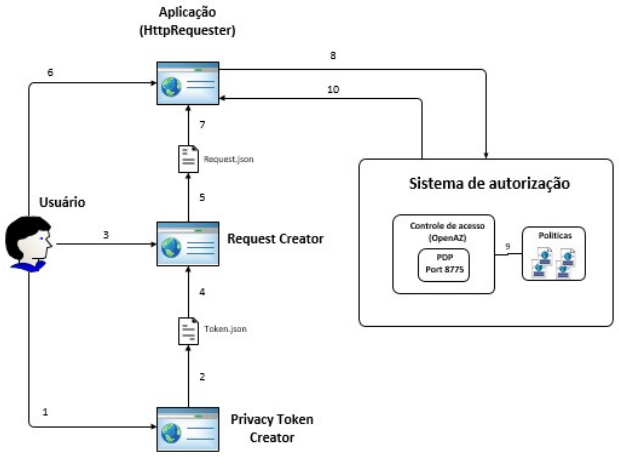


Figura 1. Modelo da proposta para controle de privacidade.

Considerando a privacidade, gerenciamento de identidades e a parte que trata de controle de acesso, a proposta se resume em modelar e utilizar políticas de privacidade utilizando uma linguagem de controle de acesso para sistema de autorização de um sistema de gerenciamento de identidades, para qualquer aplicação *Web* que faca uso de informações pessoal (PII) de uma determinada entidade. Para que isto seja possível precisa-se de vários componentes e passos ilustrados na Figura 1.

6.2. O modelo proposto de política de privacidade

Tendo entendido a estrutura de uma política de controle de acesso, foi possível estruturar baseado nos conceitos de privacidade e trabalhos relacionados de uma forma que pudesse tratar a privacidade. Desta maneira abstraiu-se a pergunta: Quem terá o benefício do uso do recurso para realizar uma ação com qual finalidade utilizando que tipos de dados?

Assim, foi possível criar um conjunto de políticas com 4 políticas. Uma política que trata de regras do beneficiário, outra que trata de regras de finalidade, outra que trata de regras de tipos de dados e outra responsável pelo *privacy token.*

O *privacy token* é um arquivo JSON com uma série de dados como nome do usuário e preferências de privacidade. As preferencias é dada por uma tripla que é uma combinação de 3 campos, tipos de dados, finalidade e beneficiário. Cada um desses campos possuem possíveis valores. Ex: O usuário deseja atribuir permissão para ele próprio ter o benefício do acesso a tipo de dados pessoais e de relacionamentos para finalidade científica. As triplas formada seriam *PI_SC_PP=1* e *RS_SC_PP=1*, as demais triplas teriam valor *0*.

- **Regra do *PrivacyToken*:** Trata os valores vindos da requisição extraídas do *token* de privacidade em comparação com as regras de privacidade.
- **Regra do Beneficiário:** Trata aquele que é beneficiado pelo uso do recurso, podendo nesse caso ser o provedor de serviços (SP), o próprio usuário (PP) e uma terceira pessoa (TP).
- **Regra da Finalidade:** Para a finalidade de uso do recurso, se pode ter os valores Comercial (CO), Científica (SC) e Melhoria de serviço (SI).

- **Regra do Tipo de dados:** Para o tipo de dado a ser usado é um agrupamento de dados como dados pessoal (PI), Características e preferências pessoal (PCP) , localização (LO), Hábitos e atividades (AH) e Relacionamentos (RS).

Cada política teria duas regras, uma com as condições que precisam ocorrer para a permissão e outra para negar automaticamente caso o resultado da primeira regra seja diferente de *Permit*. Os algoritmos de cada política seria *Permit Overrides*, que retornará *Permit* caso todas as condições sejam cumpridas. A política raiz usa o algoritmo *deny overrides* para caso de uma das políticas falhar. Se uma delas falhe, então retornará negação do recurso. Observe na Figura 2 a estrutura da política gerada.

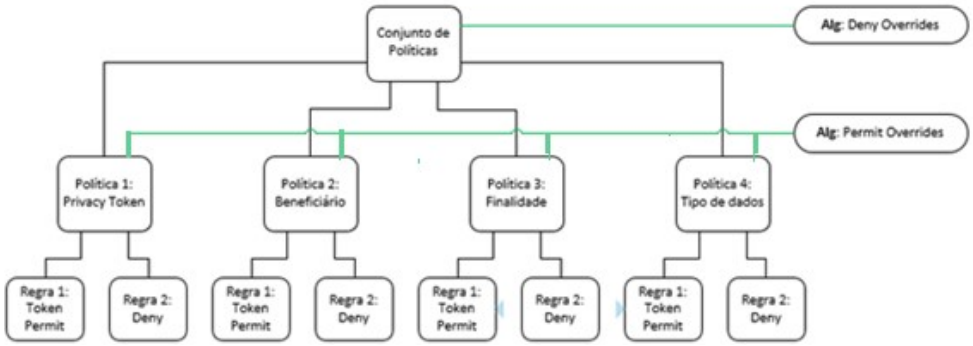


Figura 2. Modelo de política desenvolvido

Um exemplo da política em XACML está na Figura 8.

6.3. Passos do controle de privacidade

Para executar o sistema proposto na Figura 1, precisaram-se primeiramente de um sistema que se gera as preferências de privacidade. Esse sistema foi baseado em um dos trabalhos relacionados, referente ao *privacy token* de Villarreal (2017).

Os passos 1 e 2 representam a configuração do *token* de preferência do usuário e a geração deste token em formato JSON, pelo sistema respectivamente. Este sistema chamado *Privacy Token Creator* assim foi implementado utilizando a linguagem Java e o Spring como *framework*, por ser uma ferramenta robusta e bastante utilizada para desenvolvimento para *Web.* Observe a Figura 3.

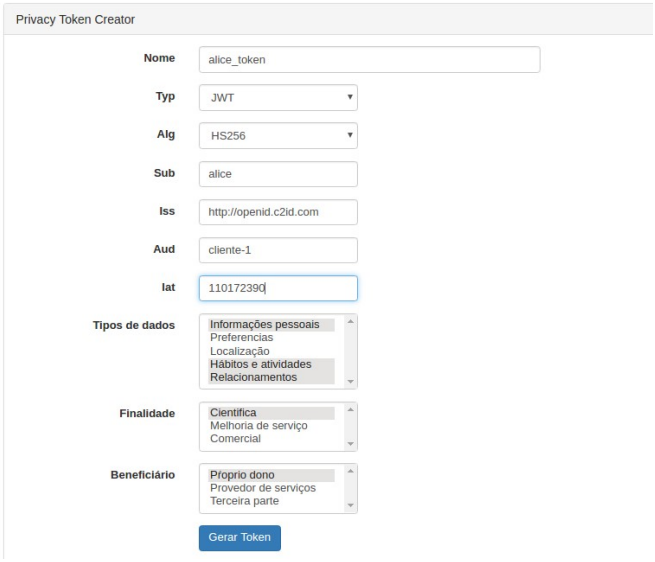


Figura 3. *Privacy Token Creator*

Os campos são preenchidos de acordo com a necessidade do usuário mas o mais importante são as preferencias, que é a combinação dos 3 ultimos campos. Todos são múltiplas seleções. Ex: O usuário deseja atribuir permissão para ele próprio ter o benefício do acesso a tipo de dados pessoais e de relacionamentos para finalidade científica. As triplas formada seriam *PI_SC_PP=1* e *RS_SC_PP=1*, as demais triplas teriam valor *0* como ilustra a Figura 4. O formato JSON foi escolhido para criar o *token* de privacidade (*privacy token*), pois alguns IdM utilizam esse formato para troca de mensagens, como o OpenID Connect.

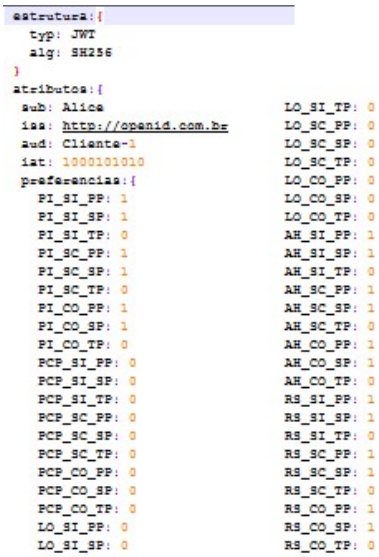


Figura 4. *Privacy Token*

Os passos 3, 4 e 5 são relacionados a outro sistema desenvolvido neste trabalho denominado *Request Creator.* Ele é responsável por utilizar o *token* de privacidade gerado pelo usuário e criar a estrutura da requisição que será enviada. O passo 3 é a configuração do sistema. Veja na figura 5.

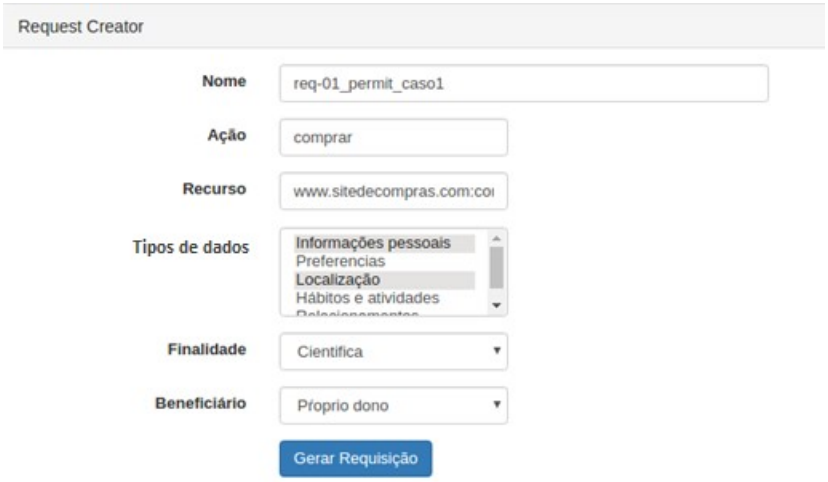


Figura 5. *Privacy Token*

O preenchimento é semelhante ao do *Privacy Token Creator* porem só se pode escolher um beneficiario e uma finalidade já que não teria sentido ter acesso simultaneo e finalidades diferentes neste trabalho. Para isso é necessário usar o *token* gerado no passo 3 para coletar as preferências. Ao fim o sistema captura todos os *tags* das triplas com valor 1 e coloca na requisição como uma *bag,* que é um conjunto de dados de mesmo tipo, nesse caso, *preferences*. O resultado do sistema é o arquivo JSON ilustrado na figura 6.


```
"Request" : {
  "AccessSubject" : {
          "Attribute" : [
              {
                  "AttributeId" : "subject:subject-id",
                  "DataType" : "string",
                  "Value" : "PP"
              },
              {
                  "AttributeId" : "subject:preferences",
                  "DataType" : "string",
                  "Value" : "pi_sc_pp"
              },
              {
                  "AttributeId" : "subject:preferences",
                  "DataType" : "string",
                  "Value" : "ah_sc_pp"
              }  ]
   },"Action" : {
          "Attribute" : {
              "AttributeId" : "action:action-id",
              "DataType" : "string",
              "Value" : "comprar"
          }
   },"Resource" : {
          "Attribute" : [{

              "AttributeId" : "resource:resource-id",
              "DataType" : "string",
              "Value" : "www.sitedecompras.com:compras:produtoid"
          },
          {
              "AttributeId" : "resource:finalidade",
              "DataType" : "string",
              "Value" : "SC"
          },
          {
              "AttributeId" : "resource:tipo-dado",
              "DataType" : "string",
              "Value" : "PI"
          }]
      }
}
```


Figura 6. Requisição em JSON

Pode-se perceber que requisição é dividida em atributos de sujeito(AccessSubject), atributos de ação (Action) e atributos de recurso (Resources).

Cada um possui seu identificador, seu valor e seu tipo de dado que devem condizer com os presentes na política a qual será comparada.

Os passos 6, 7 e 8 são realizados com a ajuda de um *plugin* do Mozilla Firefox chamado HTTP Requester que faz requisições para um determinado serviço. No passo 6 o usuário está solicitando um serviço qualquer. O passo 7 é a coleta da requisição já criada pelo Request Creator. Por fim o passo 8 envia essa requisição para ser usada pelo serviço e solicitar o acesso ao recurso pedido.

O passo 9 é quando o PDP avalia as políticas de privacidade pré-definidas, procurando para ver se existe uma aplicação comparando a requisição com o *target* da política. Ele toma a decisão de acordo com o resultado das regras definidas. O PDP neste trabalho é um *endpoint* configurado através do sistema de controle de acesso OpenAZ, um sistema desenvolvido pela AT&T e incubado pela Apache. O OpenAZ implementa módulos de controle de acesso como PEP e PAP, porem neste trabalho foi utilizada apenas o PDP.

Por fim no passo 10 PDP retorna para o usuário o resultado da solicitação, dando ou não a permissão de acesso.

7. Resultados Experimentais

Para os testes do caso de uso o OpenAZ será executado na porta 8775 para decisão do PDP. A política está em um diretório local para serem usadas para o controle de acesso.

A ação pretendida é a tentativa de um usuário realizar uma coleta de informações sobre o número de compradores no histórico de compras de um determinado usuário com finalidade comercial. O recurso se encontra em www.sitedecompras.com:userid:historico:qtde_compradores.

Para o caso de uso existe uma política criada manualmente para este trabalho, representada de forma mais clara pela política da Tabela 1, e três requisições que foram geradas pelo algoritmo RequestCreator e representadas na Tabela 2 para melhor compreensão. Duas das requisições possuem um caso de acesso permitido e outra de acesso negado.

Tabela 1. Representação da política.

DADOS DE PERMISSÃO DA POLÍTICA						
	Ação	**Recurso**	**Tipos de Dados**	**Finalidade**	**Beneficiário**	**Operação**
Caso de uso	Coletar	qtde_compradores	PI, AH, RS	CO	PP,SP	**Bem: OR**

Pode-se observar na Tabela 1, representa a política XACML na qual as requisições deste caso de uso buscam acessar através do *target* para localizá-la. O PDP usa a requisição e o *target* da política contendo recurso que se deseja acessar e a ação que se deseja fazer com este recurso, neste caso, coletar é a ação e o recurso é o número de compradores no histórico de compras de um usuário em questão.

Tabela 2. Representação das requisições

REQUISIÇÕES DO CASO DE USO						
Requisição	**Tipos de dados**	**Finalidade**	**Beneficiário**	**Token Válido?**	**Target**	**Status**
1.a	PI, AH,RS	CO	PP	SIM	OK	**PERMIT**
1.b	PI, AH,RS	CO	SP	SIM	OK	**PERMIT**
2	PI, AH,RS	CO	PP	NÃO	OK	**DENY**

Duas das requisições possuem um caso de acesso permitido e outra de acesso negado. A ação pretendida é a tentativa de um usuário realizar uma coleta de informações sobre o número de compradores no histórico de compras de um determinado usuário com finalidade comercial.

Os passos realizados são os seguintes para todos os casos:

Passo 1: Cria-se uma requisição para tentar obter acesso ao sistema, utilizando o Request Creator. A configuração da requisição criada pelo RequestCreator é exatamente como descrito no caso de uso Tabela 2.

Passo 2: A requisição é enviada para o serviço do PDP através do HTTP Requester.

Passo 3: O PDP localizará uma política aplicável com uma comparação do *target* da política e a requisição enviada e tomará a decisão de acordo com os resultados da comparação.

Passo 4: PDP retorna resposta para o usuário no HTTP Requester. Para que seja concedida a permissão, todas as políticas precisam ter a decisão *permit* para que o conjunto de políticas de privacidade conceda o controle de acesso. Caso contrário a autorização será negada (*deny*).

Na Figura 7 pode-se ver parte de dois *tokens*, o *token* 1 e o *token* 2, cada um com suas preferências que serão usadas nos testes.

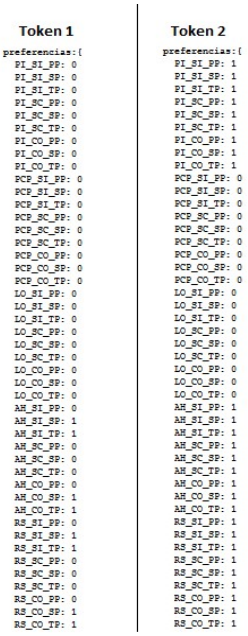


Figura 7. Representação das requisições

O Token 1 é um subconjunto do Token 2 e portanto fornece menos privilégios a quem for tentar algum acesso. O Token 1 será usado para os casos de negação e o Token 2 para os casos de permissão. A estrutura das requisições foi fixada para facilitar os testes.

O recurso se encontra em *www.sitedecompras.com:userid:historico:qtde_compradores.* Pode-se observar na Tabela 2, o caso de uso representa a política da Figura 8 na qual as requisições deste caso de uso buscam acessar através do alvo para localizá-la. O PDP usa a requisição e o *target* da política contendo recurso que deseja-se acessar e a ação que deseja-se fazer com este recurso, neste caso, coletar é a ação e o recurso é o número de compradores no histórico de compras de um usuário em questão.

```
<Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:or">
    <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:and">
        <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:any-of-any">
            <Function FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-equal"/>

            <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-bag"> <!-- pi_co_pp -->
                <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">pi_co_pp</AttributeValue>
            </Apply>
            <AttributeDesignator DataType="http://www.w3.org/2001/XMLSchema#string"
                                 MustBePresent="false"
                                 Category="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject"
                                 AttributeId="subject:preferences"/>
        </Apply>
        <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:any-of-any">  <!-- ah_co_pp -->
        <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:any-of-any">  <!-- rs_co_pp -->
        <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:any-of-any">  <!-- regra só se aplica se for para PP -->
            <Function FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-equal"/>

            <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-bag">
                <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">PP</AttributeValue>
            </Apply>
            <AttributeDesignator DataType="http://www.w3.org/2001/XMLSchema#string"
                        MustBePresent="false"
                        Category="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject"
                         AttributeId="subject:subject-id"/>

        </Apply>

</Apply>
    <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:and">
        <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:any-of-any">  <!-- pi_co_sp -->
        <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:any-of-any">  <!-- ah_co_sp -->
            <Function FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-equal"/>

            <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-bag">
                <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">ah_co_sp</AttributeValue>
            </Apply>
            <AttributeDesignator DataType="http://www.w3.org/2001/XMLSchema#string"
                         MustBePresent="false"
                         Category="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject"
```

Figura 8. Trecho da politica de privacidade 3.

Para a regra receber *permit* dentro da política que trata do *token*, o dado da requisição vindo do *subject:preferences*, precisa conter os dados presentes na política para cada tipo de dado solicitado. Além disso, o usuário beneficiário precisa ser avaliado junto ao *token* além da política que trata do beneficiário, para que o acesso seja coerente, ou seja, o *subject:subject-id* vinda da requisição precisa ser o usuário estipulado na regra.

Caso isso não seja feito, mesmo que os dados não tenham permissão para um determinado usuário, ele pode vir a ter acesso permitido. Essa exceção acontece, pois agora o privilégio foi dado a mais de um tipo de usuário. Caso a regra referente ao *token* não seja obedecida, a regra 2 retornará *deny* para a política do *token*. Analisaremos agora os casos de permissão e o caso de negação para a política do caso de uso 3.

7.1. Permissão concedida

A requisição usada neste exemplo é a requisição 1.a e 1.b da Tabela 2. O usuário tenta realizar a coleta do número de compradores para fim comercial. No caso o usuário é próprio dono do recurso (PP) na requisição 1.a e o provedor de serviço (SP) na 1.b. O

token 2 da Figura 7 está anexado a requisição. Podemos observar que todos os requisitos necessários para o acesso estão coerentes com os dados que a política. Não difere muito dos casos anteriores, a única diferença é que para este caso dois tipos de beneficiário provedor de serviço(SP) ou o próprio usuário (PP) tem o privilégio da ação de coleta para fim comercial do recurso desejado, pois o *token* está habilitado para este fim. O resultado de *Permit* é apresentado da mesma maneira como ilustrado na Figura 8.

7.2. Permissão negada

A requisição usada neste exemplo é a requisição 2 da Tabela 5. O usuário tenta realizar uma coleta de dados do histórico de compras de um usuário com a finalidade comercial. O token 1 da Figura 7 está anexado a requisição. Pode-se observar que todos os requisitos necessários para o acesso estão coerentes com os dados que a política. Porém existe um caso o qual não passa o acesso, no caso o *token* de privacidade não foi habilitado para permitir o acesso ao beneficiário realizar uma ação e, portanto sua permissão é negada a fim de proteger a privacidade do proprietário do recurso. Como citado anteriormente, para este caso além das regras de comparação do *token* na política do *token*, é necessário também comparar o beneficiário na mesma política do *token* além da comparação já realizada na política do beneficiário.

8. Conclusão

Neste trabalho foi realizada a criação de políticas de privacidade utilizando a linguagem de controle de acesso XACML além da criação de algumas ferramentas que ajudaram a tornar o ambiente de testes mais simples de se utilizar como o *Privacy Token Creator* e *Request Creator*. Os estudos das bibliografias coletadas deu base aos resultados positivos dos experimentos realizados neste trabalho.

Foi possível entender durante a concepção desse trabalho como funciona um sistema de controle de acesso e a dificuldade que se pode ter para montar politicas manualmente. Além disso, definir o papel de cada entidade participante do modelo proposto pode ser complicado. Um exemplo poderia ser quem é o responsável pela coleta de informações? Aonde são armazenados os dados de preferência no modelo mantendo a estrutura do sistema de controle de acesso existente?

Foi necessário um estudo de um sistema de controle de acesso assim como o entendimento da construção de uma política de controle de acesso para aderir o conceito de privacidade a ela.

O estudo envolveu outros trabalhos relacionados onde foi utilizado conceitos de um deles para que fosse possível concluir o trabalho e chegar no resultado. Também foi necessário desenvolver uma arquitetura que pudesse realizar todos os passos necessários para gerar o resultado, como sistemas que gerassem requisições e o *privacy token*.

As principais contribuições deste trabalho foram: Modelo de política de privacidade que pode ser usada junto a políticas de controle de acesso convencionais e a implementação de um sistema que cria um *Token* de privacidade baseado no modelo de *Privacy Token* de trabalhos relacionados.

Para trabalhos futuros poderiam ser sugeridos uma forma de integração das políticas com um sistema de gerenciamento de identidades, mas especificamente o OpenID Connect, para inserir a essa proposta em um modelo real um IdM.

Referências

Bertino, E. e Takahashi, K. (2011). **Identity Management: Concepts, Technologies, and Systems**. Artech House

Camillo, G. L. ; Westphall, C. M. ; Werner, J. ; Westphall C. B.; **Preserving Privacy with Fine-grained Authorization in an Identity Management System**. 2017.

EDUCAUSE review, **Information Privacy Revealed**, 2013 Disponível em: <http://er.educause.edu/articles/2013/1/information-privacy-revealed> Acesso em 2 de Julho de 2017.

Hu, V. C.; Kuhn, R.; Ferraiolo, D. F., **Attribute-Based Access Control National, Institute of Standards and Technology**, Gaithersburg, MD, USA Attribute-Based Access Control Disponível em: <https://www.researchgate.net/publication/273393378_Attribute-Based_Access_Control. > Acessado em : 10 de março 2017.

ISO/IEC 29100. **international standard - information technology- security techniques - pri vacy framework**, 2011.

OECD , **OECD Guidelines on the Protection of Privacy and Transborder Flows of Personal Data**, Cap 1, 2013.

Rosset, V. **Um modelo de autorização e distribuição de direitos de acesso sobre conteúdos digitais**. 2004. Dissertação (Mestrado) – Centro de Ciências Exatas Universidade Federal de Santa Catarina, Florianópolis.

Silva, E. F.; Muchaluat-Saadel, D. ; Fernandes, N. C. **Controle de Acesso Baseado em Políticas e Atributos para Federações**

Villareal, M. E.; Westphall, C. M. **Privacy Token: A Mechanism for User's Privacy Specification in Identity Management Systems for the Cloud**, 2017.

Werner, J.; Westphall, C., "**A Model for Identity Management with Privacy in the Cloud**" in 2016 IEEE Symposium on Computers and Communication (ISCC), Messina, Italy, June 2016.

Werner J., Westphall C.M, Westphall C. B., **Cloud identity management: A survey on privacy strategies**. Computer Networks 122, Elsevier. 2017