UNIVERSIDADE FEDERAL DO MARANHÃO

Curso de Ciência da Computação

Luiz Felipe Gomes Ribeiro

Extensões da Plataforma InterSCity para Suporte à Aplicações Seguras de Sensoriamento Participativo

São Luís

2025

Luiz Felipe Gomes Ribeiro

Extensões da Plataforma InterSCity para Suporte à Aplicações Seguras de Sensoriamento Participativo

Monografia apresentada ao curso de Ciência da Computação da Universidade Federal do Maranhão, como parte dos requisitos necessários para obtenção do grau de Bacharel em Ciência da Computação.

Orientador: Prof. Dr. Francisco José da Silva e Silva

São Luís

2025

Ficha gerada por meio do SIGAA/Biblioteca com dados fornecidos pelo(a) autor(a).
Diretoria Integrada de Bibliotecas/UFMA

Gomes Ribeiro, Luiz Felipe.
Extensões da Plataforma InterSCity para Suporte à Aplicações Seguras de Sensoriamento Participativo / Luiz Felipe Gomes Ribeiro. - 2025.
49 p.

Orientador(a): Francisco José da Silva e Silva.
Curso de Ciência da Computação, Universidade Federal do Maranhão, Laboratório de Sistemas Distribuídos Inteligentes, 2025.

1. Cidades Inteligentes. 2. Sensoriamento Participativo. 3. Segurança. 4. Gerenciamento de Identidade. 5. Identidade Autosoberana. I. da Silva e Silva, Francisco José. II. Título.

Luiz Felipe Gomes Ribeiro

Extensões da Plataforma InterSCity para Suporte à Aplicações Seguras de Sensoriamento Participativo

Monografia apresentada ao curso de Ciência da Computação da Universidade Federal do Maranhão, como parte dos requisitos necessários para obtenção do grau de Bacharel em Ciência da Computação.

Trabalho _____ em São Luís, 7 de agosto de 2025:

Prof. Dr. Francisco José da Silva e Silva
Orientador

Prof. Dr. Primeiro Membro
Examinador

Profa. Dra. Segundo Membro
Examinador

São Luís
2025

Agradecimentos

Agradeço profundamente e principalmente a Deus e aos meus pais, sem eles eu não estaria nem perto de onde eu estou hoje. Foram eles que nunca desistiram de mim e que me incentivaram a seguir em frente.

Agradeço também ao Hiroyuki Sawano e ao Shiro Sagisu pelas suas composições que me deram motivação em diversos momentos difíceis ao decorrer do curso. Melodias como *Perfect time*, *Number One*, *Clavar la Espada*, *Hands Up to the Sky*, *Before My Body Is Dry*, *Dora Gong4n*, *Treachery*, *Gravity Wall*, *Bios*, *Ninelie* e *Attack on Titan* me fizeram ter forças para continuar trabalhando mesmo nos momentos de maior dificuldade.

Agradeço aos meus Colegas do LSDi e da universidade pelo apoio e todas as interações que trouxeram leveza e alegria em diversos momentos complicados durante essa jornada.

Um Agradecimento especial para o professor Francisco que me acolheu como membro voluntário, e mais tarde, como estagiário e bolsista de iniciação científica do LSDi. Agradeço a ele por toda a orientação durante esse trabalho.

Por fim, agradeço aos demais professores do curso que contribuíram significativamente para minha formação acadêmica e profissional.

"Não fiques em terreno plano.
Não subas muito alto.
O mais belo olhar sobre o mundo
Está a meia encosta."

Friedrich Nietzsche, em *"A Gaia Ciência"*

Resumo

As Cidades Inteligentes são ambientes que visam prover maior qualidade de vida aos seus cidadãos por meio do amplo emprego de tecnologias de comunicação e informação. A segurança da informação torna-se crucial em Cidades Inteligentes devido à ampla troca de dados por meio de dispositivos de IoT, redes de computadores e infraestrutura de computação em nuvem. Este trabalho propõe o uso de extensões de software para garantir a segurança em aplicações de Cidades Inteligentes. Essas mesmas aplicações, executam em plataformas de middleware e realizam sensoriamento participativo através da coleta dados de multimídia a partir de dispositivos computacionais utilizados pelos cidadãos da cidade, como smartphones. Para esse projeto, foi utilizada a plataforma de middleware InterSCity, desenvolvida por um consórcio de universidades brasileiras do qual a UFMA participa através do seu Laboratório de Sistemas Distribuídos Inteligentes, e que é voltada ao desenvolvimento de aplicações no domínio das Cidades Inteligentes. Para validar a efetividade das extensões aplicadas, foi utilizado como estudo de caso uma aplicação denominada `Spotter`, cujo objetivo é o rastreamento de veículos em Cidades Inteligentes por meio de sensoriamento oportunístico. Os experimentos realizados através deste estudo de caso atestaram um desempenho satisfatório com relação aos processos de autenticação e inserção de dados, bem como a viabilidade do uso dessas extensões para o provimento da segurança necessária nesta importante classe de aplicações voltadas ao domínio das Cidades Inteligentes.

Palavras-chave: Cidades Inteligentes. Sensoriamento Participativo. Segurança. Gerenciamento de Identidade. Identidade Autosoberana.

Abstract

Smart Cities are environments that aim to provide a higher quality of life to their citizens through the widespread use of information and communication technologies. Information security becomes crucial in Smart Cities due to the extensive exchange of data through IoT devices, computer networks, and cloud computing infrastructure. This work proposes the use of software extensions to ensure security in Smart City applications. These applications run on middleware platforms and perform participatory sensing by collecting multimedia data from computing devices used by citizens, such as smartphones. For this project, the InterSCity middleware platform was used, developed by a consortium of Brazilian universities in which UFMA participates through its Laboratory of Intelligent Distributed Systems (LSDi). This platform is focused on supporting the development of applications within the Smart Cities domain. To validate the effectiveness of the proposed extensions, a case study was conducted using an application called `Spotter`, whose objective is to track vehicles in Smart Cities through opportunistic sensing. The experiments conducted in this case study demonstrated satisfactory performance in terms of authentication and data insertion processes, as well as the viability of using these extensions to provide the necessary security for this important class of applications within the Smart Cities domain.

Keywords: Smart Cities. Participatory Sensing. Security. Identity Management. Self-Sovereign Identity.

Lista de ilustrações

Lista de tabelas

Lista de abreviaturas e siglas

API	*Application Programming Interface*
CC	*Ciência da Computação*
CI	*Cidade Inteligente*
DID	*Identificador Descentralizado*
DIDComm	*Decentralized Identifier Communication*
IoT	*Internet das Coisas*
IPS Lib	*InterSCity Participatory Sensing Library*
JWT	*JSON Web Token*
LSDi	*Laboratório de Sistemas Distribuídos Inteligentes*
SSI	*Identidade Auto-soberana*
UFMA	*Universidade Federal do Maranhão*

Sumário

1 Introdução

O crescimento acelerado da população das grandes cidades tem se tornado um dos grandes problemas enfrentados no século XXI. A velocidade com que a população cresce frequentemente supera a capacidade de desenvolvimento urbano das cidades, resultando em diversas deficiências. Entre essas deficiências estão o trânsito desordenado, a infraestrutura carente, a insuficiência da segurança pública e outros problemas correlatos. Esse fenômeno leva a um cenário urbano em que a qualidade de vida dos habitantes é diretamente afetada.

Dados recentes da Organização das Nações Unidas indicam que, até 2050, mais de dois terços da população mundial viverá em áreas urbanas (Organização das Nações Unidas, 2022). No Brasil, esse processo de urbanização é particularmente acelerado, com mais de 85% da população já vivendo em cidades, segundo o IBGE (Instituto Brasileiro de Geografia e Estatística, 2025). Esse êxodo rural pressiona ainda mais os sistemas urbanos, exigindo novas soluções tecnológicas e sustentáveis. Neste contexto, as cidades inteligentes emergem como um modelo viável para solucionar esses problemas, oferecendo soluções inovadoras e eficazes que podem ser integradas à estrutura urbana existente.

O conceito de cidade inteligente ainda não possui uma definição sólida e universalmente aceita de seus atributos descritivos (BRITO et al., 2023). Contudo, há um consenso geral de que uma cidade, para ser considerada inteligente, deve utilizar a tecnologia de forma ampla e generalizada para automatizar e aumentar a eficiência de seus processos internos. Essa utilização tecnológica abrange uma gama de dispositivos e sistemas que interagem entre si, proporcionando uma gestão urbana mais eficiente. Tais dispositivos vão desde sensores em ruas, semáforos e veículos, até aplicações móveis utilizadas pelos próprios cidadãos, promovendo uma conectividade constante entre os elementos da cidade.

Para que a implementação dessas diversas tecnologias seja possível, o emprego de middleware [1] torna-se essencial. O middleware atua como uma ponte entre os dispositivos físicos, os sensores e as aplicações, sendo responsável por orquestrar a troca de informações e a integração dos diversos sistemas da cidade. A aplicação de middleware facilita a integração, a comunicação e a gestão eficiente dos diferentes sistemas e dispositivos, permitindo uma operação harmoniosa e eficaz da infraestrutura urbana. Essa camada intermediária é vital para o funcionamento de serviços inteligentes, pois oferece abstrações que permitem o desenvolvimento e a escalabilidade, além de implementarem novas funcionalidades.

o uso amplo dessas diversas tecnologias permite o surgimento de uma nova classe de aplicações: as de sensoriamento participativo. Essa abordagem consiste na

[1] software de computador que fornece serviços para softwares aplicativos além daqueles disponíveis pelo sistema operacional

colaboração ativa dos próprios cidadãos, que utilizam seus dispositivos pessoais, como smartphones, para coletar e compartilhar informações sobre o ambiente urbano. Em vez de depender exclusivamente de sensores instalados pela administração pública, o sensoriamento participativo amplia a cobertura e a diversidade dos dados capturados, transformando cada cidadão em um agente colaborador do ecossistema da cidade inteligente.

O sensoriamento participativo possibilita o monitoramento de uma variedade de fenômenos, como o tráfego de veículos, buracos nas vias, alagamentos, falhas na iluminação pública, entre outros. Além de viabilizar a coleta de informações em tempo real, essa estratégia fortalece o engajamento social e promove uma relação mais próxima entre o cidadão e a gestão pública. Aplicações que adotam esse modelo são, portanto, essenciais para tornar a cidade mais responsiva, eficiente e conectada com sua população.

Apesar dos benefícios evidentes dessas aplicações, o uso do sensoriamento participativo também levanta preocupações quanto à segurança e a privacidade dos dados coletados. Como os cidadãos atuam como fontes diretas de informação, é necessário garantir que os dados transmitidos sejam autênticos, íntegros e provenientes de fontes confiáveis. A ausência de mecanismos de validação pode resultar na inserção de informações falsas ou maliciosas, comprometendo a confiabilidade das análises e das decisões baseadas nesses dados. Dessa forma, é essencial que os sistemas de sensoriamento participativo e as plataformas de cidades inteligentes implementem controles de acesso, canais de comunicação seguros e mecanismos de autenticação.

1.1 Justificativa e caracterização do problema

O sensoriamento participativo é um conceito em que usuários de dispositivos móveis coletam e compartilham dados ambientais para que provedores de serviços os utilizem na execução de atividades urbanas (CONNOLLY et al., 2019). Essa abordagem transforma o cidadão de mero consumidor de informações em agente ativo da coleta e geração de dados, possibilitando as mais diversas aplicações, como monitoramento de enchentes, buracos em vias e denúncias de situações suspeitas. No entanto, para que esse modelo seja viável, é fundamental garantir que os dados sejam coletados e transmitidos com segurança, assegurando sua confiabilidade, a privacidade dos participantes e a integridade do sistema como um todo.

Um exemplo concreto dessa abordagem é a aplicação desenvolvida neste trabalho, nomeada de `Spotter`, essa aplicação foi desenvolvida com o objetivo de rastrear veículos. Nesse caso, cidadãos voluntários utilizam smartphones para capturar e enviar imagens e localizações de carros de interesse público. Essa aplicação ilustra o potencial do sensoriamento participativo ao aproveitar a infraestrutura já existente — no caso, as câmeras dos celulares dos cidadãos — de forma escalável e de baixo custo. Dessa forma, o

`Spotter` ajuda na atuação eficiente da segurança pública com a participação direta do cidadão. Todavia, sem mecanismos robustos de segurança, tal sistema estaria vulnerável a falsificações de identidade, manipulação de dados e violações de privacidade, o que pode causar deturpações no funcionamento correto do sistema.

As plataformas de gerenciamento de cidades inteligentes, como o InterSCity, desempenham papel crucial nesse ecossistema ao mediar a comunicação entre a cidade e as aplicações urbanas. Operando em cenários distribuídos e heterogêneos, essas plataformas enfrentam o desafio de garantir a confiabilidade das trocas de informação. Portanto, o uso de mecanismos de segurança que assegurem a integridade, confidencialidade e disponibilidade são essenciais (JAMES, 2020). Tais medidas são ainda mais críticas quando se lida com dados sensíveis e pessoais de cidadãos.

A integração de extensões de segurança nas CIs, cria uma rede segura que viabiliza o uso otimizado de recursos urbanos e promove cidades inteligentes mais eficientes. Isso reforça a relevância deste trabalho, que propõe uma abordagem estruturada para aplicações participativas seguras, equilibrando assim, a inovação tecnológica com proteção de dados e privacidade cidadã.

1.2 Objetivos

O objetivo geral deste trabalho de conclusão de curso é investigar mecanismos e extensões que possibilitem o compartilhamento seguro e protegido de dados em aplicações que realizem sensoriamento participativo no contexto de cidades inteligentes da plataforma InterSCity. São objetivos específicos desse trabalho:

1.2.1 Objetivos Específicos

- Conceber uma arquitetura de sensoriamento participativo a ser integrado a uma plataforma de software de apoio ao desenvolvimento de aplicações para cidades inteligentes.
- Aplicar métodos e extensões seguras para o transporte de dados provenientes do sensoriamento participativo de cidadãos, além de aplicar formas de se estabelecer canais seguros de comunicação que permitam garantir a autenticação e a integridade dos dados registrados pelos cidadãos da cidade.
- Projetar e implementar uma biblioteca que facilite a integração de extensões de segurança do InterSCity em aplicações que realizem sensoriamento participativo.
- Projetar e implementar um estudo de caso voltado ao rastreamento de veículos em cidades inteligentes que permitam avaliar a aplicabilidade da arquitetura de

sensoriamento participativo desenvolvida e de extensões de segurança aplicadas ao projeto.

- Realizar experimentos e testes que coloquem a prova a efetividade da solução e avaliar o peso de seus efeitos a fim de validar a proposta.

2 Fundamentação Teórica

2.1 Blockchain

A blockchain é uma tecnologia que serve como uma fonte de dados distribuída que armazena registros em uma cadeia de blocos. Essa estrutura garante assim as seus príncipios base, tais como a descentralização, a imutabilidade, a transparência e a auditabilidade (MELO, 2021). Inicialmente foi criada para ser a infraestrutura por trás da primeira criptomoeda, o Bitcoin, mas a tecnologia blockchain se expandiu a ponto de conseguir suportar uma ampla gama de aplicações diferentes.

A tecnologia blockchain possui várias características principais. Entre essas características se destacam a confiabilidade, integridade, inviolabilidade, imutabilidade, versatilidade, transparência e privacidade, tornando-o uma espécie de banco de dados de ativos altamente confiável com recursos seguros e transparentes. O uso de Blockchain garante a integridade dos dados e fornecendo um ambiente transparente e protegido pela privacidade no qual os usuários confiam para gerar valor (ALI; NORMAN; AZZUHRI, 2023).

Além disso, o blockchain opera como um sistema descentralizado em que as transações são armazenadas em blocos que estão interligados, formando uma cadeia contínua. Cada bloco contém um *hash*[1] do bloco anterior, uma coleção de registros e uma simples árvore de valores em *hash*, garantindo assim a integridade e a segurança das informações armazenadas. Essa tecnologia permite o registro seguro e transparente de transações, pois os dados se tornam imutáveis assim que registrados (PIERRO, 2017). Dizer que há natureza descentralizada no blockchain, significa dizer que cada participante da rede tem acesso ao mesmo livro, dessa forma, eliminando a necessidade de uma autoridade central e aumentando a segurança por meio da propriedade distribuída de informações (GUPTA, 2022).

A tecnologia do Blockchain evoluiu com o surgimento da ideia de contratos inteligentes. Os contratos inteligentes são acordos digitais executados de forma automática quando determinadas condições específicas são atendidas. Com isso, é possível eliminar dessa forma a necessidade de intermediários e por consequência aumentar a segurança nas transações automáticas (KAUR; DABAS, 2022). A evolução dessa tecnologia se expandiu de forma que seu uso está além das transações monetárias, demonstrando seu potencial em transformar a área da saúde, da Internet das Coisas e de diversos outros setores.

[1] Em blockchain, é a apresentação criptografada da identificação da transação.

2.2 Identidades e Gerenciamento de Identidades

No contexto digital, a identidade pode ser definida como um conjunto de informações estruturadas que caracterizam um indivíduo ou organização dentro de um sistema específico (JØSANG; POPE, 2005). Por exemplo, em um hospital, os dados cadastrais de um paciente constituem sua identidade dentro daquele mesmo domínio. As identidades podem representar desde pessoas físicas e até entidades corporativas, sendo possível que uma única entidade possua múltiplas identidades válidas em um mesmo contexto(ZHANG; XUE; LIU, 2019). Essas identidades são compostas por identificadores—atributos como nome, CPF ou endereço eletrônico que podem ser únicos, compartilhados, temporários ou permanentes, a depender do propósito e do domínio de aplicação (JØSANG; POPE, 2005).

Nesse cenário, a Gestão de Identidade e Acesso (IAM) emerge como um mecanismo crítico para assegurar que apenas entidades autorizadas tenham acesso a recursos sensíveis(ZHANG; XUE; LIU, 2019). Tradicionalmente, os sistemas de IAM baseiam-se em políticas centralizadas e tecnologias de autenticação convencionais. No entanto, com a crescente digitalização e a proliferação de identidades distribuídas na internet, os modelos tradicionais enfrentam desafios de privacidade e segurança.

2.2.1 Identidades Auto-soberanas

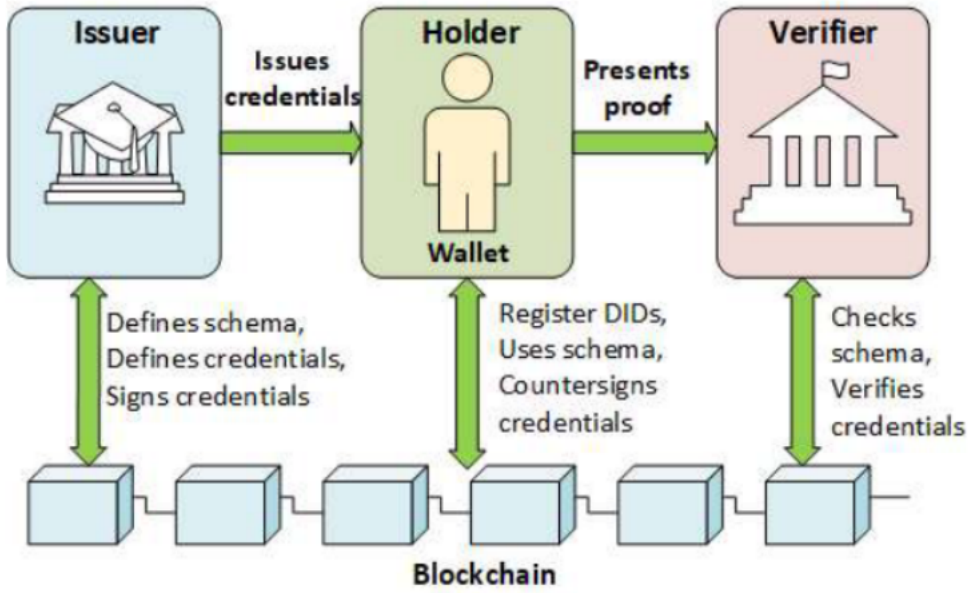
A identidade auto-soberana (SSI) é um modelo emergente de gerenciamento de identidade digital que concede ao próprio indivíduo o controle total sobre suas credenciais. Diferente de modelos tradicionais, em que a identidade é gerenciada por terceiros centralizados, como governos ou plataformas privadas, a identidade auto-soberana permite que os usuários armazenem, compartilhem e validem suas credenciais sem depender de uma autoridade central (LAMPROPOULOS; KYRIAKOULIS; DENAZIS, 2022).

A abordagem do SSI, aliada à tecnologia blockchain, permite que os indivíduos gerenciem suas credenciais de forma descentralizada, compartilhando apenas as informações estritamente necessárias para cada contexto sem comprometer sua privacidade. Por exemplo, um indivíduo pode comprovar sua maioridade sem revelar sua data de nascimento, utilizando provas de conhecimento zero (*Zero-Knowledge Proofs*) (BHATTACHARYA; ZAVARSKY; BUTAKOV, 2020). No contexto das identidades auto-soberanas, há a presença de três atores fundamentais (os quais têm as sua interações descritas na figura 1), são eles o *Holder*, *Issuer* e o *Verifier*. A seguir há a explicação das competências de cada um desses atores:

- **Holder:** Esse ator é composto pela própria entidade ou indivíduo, é o portador de sua própria identidade e gerenciador de seus documentos credenciais.

- **Issuer:** É o ator que atua como emissor de credenciais. Ele detêm o poder de apossar algum *Holder* de determinada credencial.
- **Verifier:** Tem por função verificar credenciais apresentadas por *Holders*, para dessa forma evitar tentativas de falsificações e uso de credenciais inautênticas.

Figura 1 – Fluxo de apresentação de credenciais em um modelo de SSI.



Fonte: (BHATTACHARYA; ZAVARSKY; BUTAKOV, 2020)

2.2.2 Identidades Auto-soberanas em Cidades Inteligentes

No contexto de cidades inteligentes, o paradigma de SSI oferece uma série de vantagens. Primeiramente, ele fortalece a privacidade dos cidadãos, pois reduz o compartilhamento de dados sensíveis com múltiplas entidades. Além disso, garante interoperabilidade entre diferentes plataformas e serviços urbanos, já que as credenciais necessárias para acessar esses serviços devem seguir padrões abertos e podem ser verificadas de forma padronizada.

O InterSCity Seguro (Descrito na seção 3.2) incorpora um modelo de identidade auto-soberana ao tratar cidadãos como portadores de suas próprias credenciais digitais. Cada cidadão armazena suas credenciais em sua própria carteira digital, gerenciada por eles mesmos (Holder), e pode apresentá-las diretamente a serviços urbanos, como aplicações de sensoriamento participativo. O papel das entidades administrativas, nesse modelo, é atuar como emissores confiáveis (Issuer), validando e assinando digitalmente as credenciais emitidas. A verificação, por sua vez, é feita pelos serviços consumidores por meio do (Verifier), sem a necessidade de consultar a entidade emissora a cada requisição.

Essa abordagem proporciona maior descentralização, escalabilidade e segurança no uso de dados sensíveis dentro de cidades inteligentes, promovendo uma relação de confiança entre cidadãos e serviços urbanos. Ela também facilita a implementação de aplicações que exigem diferentes níveis de acesso, pois permite controlar com precisão quem pode acessar cada funcionalidade com base em suas credenciais válidas.

2.3 Sensoriamento Participativo em Cidades Inteligentes

No contexto das cidades inteligentes, o sensoriamento participativo permite que os cidadãos atuem como agentes ativos na coleta e compartilhamento de dados urbanos. Essa abordagem transforma os habitantes de meros consumidores de informações em colaboradores essenciais para o monitoramento e otimização dos serviços urbanos. No entanto, a implementação eficaz desse modelo enfrenta desafios significativos, especialmente no que diz respeito à segurança, privacidade e confiabilidade dos dados coletados.

A plataforma InterSCity (descrita na seção 3.1), em sua versão padrão, não oferece suporte nativo ao sensoriamento participativo. Essa limitação se deve principalmente à ausência de mecanismos robustos para gerenciar identidades digitais e garantir a autenticidade e integridade dos dados provenientes dos cidadãos. Sem um sistema de gestão de identidades, torna-se impossível assegurar que os dados coletados sejam enviados por indivíduos confiáveis e que as informações compartilhadas sejam verdadeiras. Além disso, a falta de canais seguros de comunicação entre os cidadãos e a plataforma expõe o sistema a riscos como manipulação de dados, falsificação de identidades e violações de privacidade.

Em um cenário onde o sensoriamento participativo depende da colaboração voluntária dos cidadãos, a ausência desses mecanismos de segurança pode desencadear uma série de problemas. Por exemplo, dados maliciosos ou falsificados podem comprometer a qualidade das análises realizadas pela cidade inteligente, levando a decisões equivocadas ou à degradação e desperdício dos serviços urbanos. Da mesma forma, a exposição de informações sensíveis dos cidadãos durante o processo de coleta e transmissão de dados pode resultar em violações de privacidade, desencorajando a participação da população.

3 Resultados

3.1 A Plataforma InterSCity

O InterSCity é uma iniciativa com o objetivo de gerar uma infraestrutura robusta composta de tecnologias open-source criadas para atender as demandas das cidades inteligentes. Essa plataforma foi criada para atender às demandas de pesquisa, desenvolvimento e experimentação de forma colaborativa em cidades inteligentes. Baseada em uma arquitetura de componentes de microsserviços[1], o projeto InterSCity tem como objetivo oferecer as tecnologias e métodos necessários para apoiar o amplo desenvolvimento de aplicações para as cidades inteligentes no futuro (ESPOSTE et al., 2017).

3.1.1 Funcionamento do InterSCity

O InterSCity funciona com a manipulação dos recursos da cidade, que são abstrações de entidades físicas que estão presentes na cidade. Os recursos são representações de dispositivos de IoT. Esses dispositivos são classificados como sensores e atuadores. Por exemplo, um dispositivo de IoT capaz de medir a temperatura em uma parada de ônibus é recurso da cidade classificado como sensor. Uma capacidade pode ser considerada como a abstração de um atributo de um recurso, um exemplo de capacidade pode ser a localização de um dispositivo de IoT capaz de enviar as coordenadas de um trem em circulação.

3.1.2 Microsserviços do InterSCity

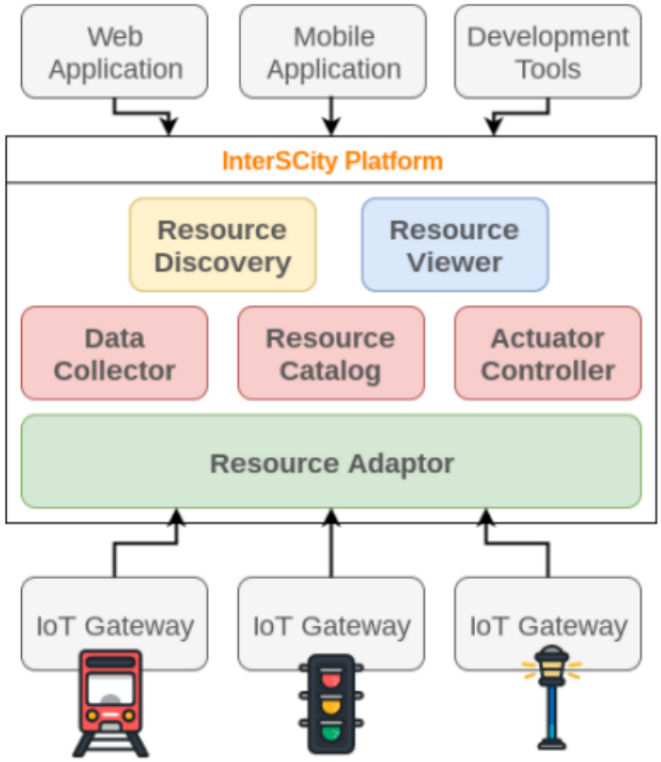
A arquitetura do InterSCity é baseada em microsserviços, esse modelo arquitetural emerge das melhores práticas da atual indústria de software para a construção de aplicações distribuídas. Essa arquitetura é composta por pequenos componentes interconectados (microsserviços), que suportam escalabilidade, evolutividade e facilitam eventuais manutenções.

O InterSCity apresenta em sua estrutura seis microsserviços que estão esquematizados na Figura 2. Esses microsserviços possibilitam e implementam um amplo e escalável trâmite de informação dentro da cidade. Segundo (ESPOSTE et al., 2017), os microsserviços disponíveis no InterSCity são:

- `Resource Adaptor`: Esse microsserviço funciona como um ponto direto de comunicação dos recursos com a cidade inteligente.

[1] estilo de desenvolvimento que organiza a aplicação como um conjunto de serviços pequenos e independentes.

Figura 2 – Estrutura arquitetural do InsterSCity.



Fonte: (ESPOSTE et al., 2017)

- `Data Collector`: É o responsável pela coleta dados obtidos por diversos sensores e dispositivos de IoT da cidade. Além disso, ele também entrega os dados coletados quando solicitado.
- `Resource Catalog`: É um serviço essencial para o funcionamento correto do InterSCity, ele é responsável por guardar informações sobre a identificação de todos os recursos presentes na plataforma.
- `Actuator Controller`: Tem por função, receber e validar as solicitações de ações destinadas aos recursos.
- `Resource Discovery`: Permite as aplicações pesquisarem e descobrirem recursos que estejam presentes na cidade.
- `Resource Viewer`: Possuí informações de recursos da cidade e os disponibiliza de forma enxuta e útil.

Ademais, esse sistema tem uma arquitetura descentralizada e baseada em microsserviços bem separados e definidos (ESPOSTE et al., 2017). Contudo, apesar dessas características, a versão básica do InterSCity sofre com a falta de canais seguros de comunicação e de acesso com os diversos dispositivos da cidade.

A ausência desses mecanismos de segurança facilitam a atuação de indivíduos mal intencionados no sistema da cidade inteligente, possibilitando vários problemas dentro do contexto da CI. Tais problemas podem causar a exposição de dados sensíveis dos cidadãos e das entidades públicas; A manipulação dos recursos públicos para fins pessoais; E causar interrupções significativas nos serviços essenciais, como transporte público, fornecimento de água, energia e até mesmo serviços de emergência (CARDOSO, 2024).

3.2 Extensão para Inserção Segura de Dados na Plataforma InterSCity

Tendo em vista as deficiências do InterSCity com relação a segurança de dados, um modelo de segurança foi concebido por (CARDOSO, 2024). Esse modelo tem o objetivo de assegurar a autenticação segura de dispositivos IoT operados por entidades públicas. A proposta visa mitigar riscos relacionados à segurança, como a inserção de dados maliciosos ou o acesso não autorizado aos sistemas urbanos da cidade inteligente. Para isso, a extensão adota uma arquitetura baseada em blockchain, com ênfase na rastreabilidade e no controle descentralizado das identidades dos dispositivos de IoT que estão conectados ao ecossistema da cidade. Os atores da solução são divididos em duas classes:

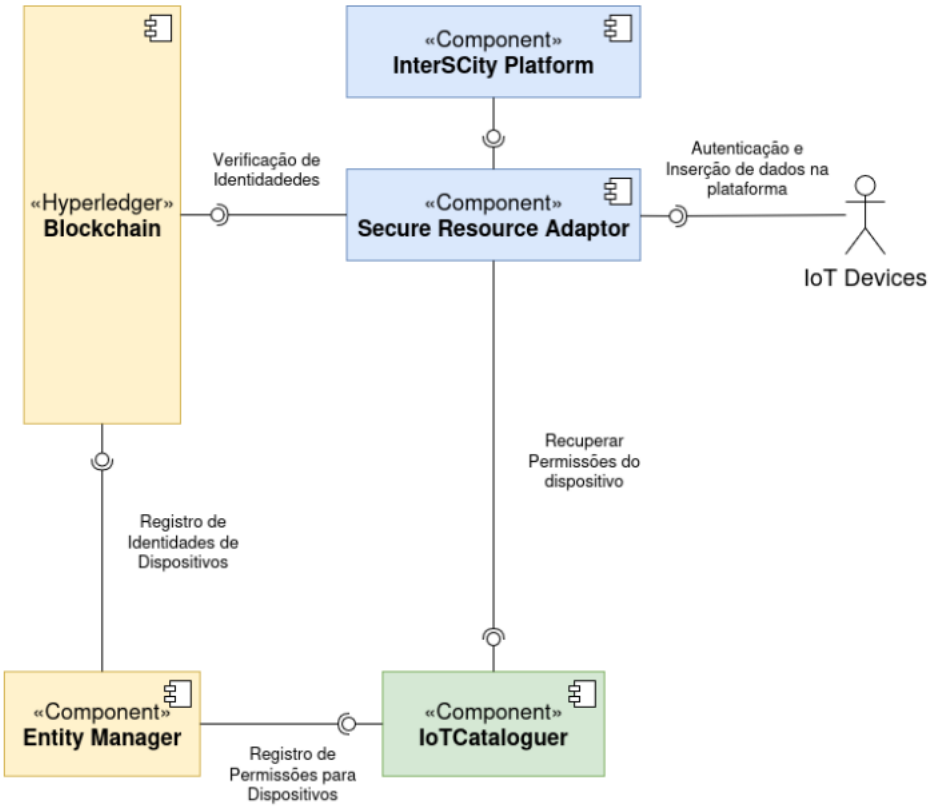
- `Entidades administrativas`: Essa classe é composta por órgãos públicos e administrativos os quais são responsáveis pela administração e operação da cidade. Essa classe incluí prefeituras, secretarias, empresas contratadas e outros entes administrativos. Todos os atores pertencentes a essa classe são imbuídos do poder para tomar decisões e administrar os recursos urbanos. Para exercer tal função com segurança, é preciso que o seu acesso ao sistema seja restrito e protegido.
- `Dispositivos IoT, Componentes de Software e Outros Atores`: Esses atores podem ser compostos por dispositivos atuadores, câmeras, sensores, aplicações, componentes de software e qualquer outra entidade transmissora de dados. É importante que os dados obtidos por esses atores sejam transmitidos de forma segura e íntegra.

A proposta desse modelo de identidade é que a rede blockchain utilizada deve ser permissionada por um consorcio de entidades administrativas. Cada entidade administrativa, atuando como provedor de identidade, tem direito de emitir sua própria identidade e utiliza-la para gerar identidades das demais entidades não administrativas. Dessa forma, existe uma hierarquia onde cada identidade armazenada na rede blockchain foi emitida por uma entidade administrativa. (CARDOSO, 2024).

Ainda com o intuito de prover segurança ao InterSCity, também foram aclopados ao InterSCity os microsserviços `Secure Resource Adaptor` e `IoT Cataloguer` dispostos na arquitetura representada na figura 3.

- **Secure Resource Adaptor**: O *Secure Resource Adaptor* surge como uma extensão do `Resource Adaptor`. Foi estendido para acomodar o conceito de recursos de capacidades do InterSCity introduzindo o conceito de que dispositivo IoT, que deve possuir uma identidade para assim publicar dados sobre as capacidades de um determinado recurso.
- **IoT Cataloguer**: Esse serviço funciona mantendo um mapeamento que relaciona o dispositivo IoT, sua identidade no blockchain, o recurso e as capacidades associadas a esse dispositivo.
- **Entity Manager**: É a entidade administrativa, e tem o direito de registrar dispositivos de IoT no `IoT Cataloguer`. Além disso, assinaturas digitais são utilizadas para garantir que apenas o `Entity Manager` emissor da identidade do dispositivo possa adicionar as permissões de escrita. (CARDOSO, 2024).

Figura 3 – Arquitetura proposta por (CARDOSO, 2024).



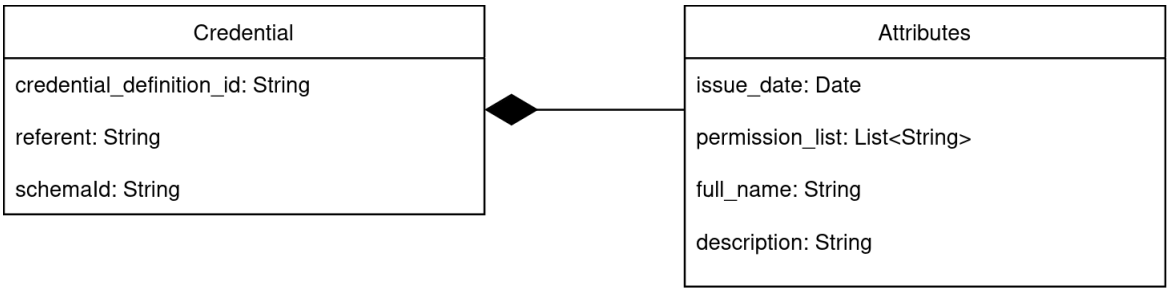
Fonte: (CARDOSO, 2024)

3.3 Extensão para uso de Identidade Auto-soberana na Plataforma InterSCity

Se baseando no modelo proposto anteriormente, também foi criado uma extensão de gerenciamento de identidades que estende o modelo apresentado por (CARDOSO, 2024) e que possibilitasse a inserção dos cidadãos da cidade inteligente como atores do InterSCity. Se definiu os cidadãos como novos atores, de modo que pudessem usufruir de serviços aos quais eles estivessem credenciados a realizar. Para possibilitar esse modelo, cada cidadão é considerado auto-soberano de suas credenciais, recebendo assim uma identidade auto-soberana.

As Credenciais são os documentos identificadores que permitem o acesso a um serviço, uma credencial nesse modelo é caracterizada pelo diagrama de classes presente na Figura 4. Portanto, os cidadãos podem obter e usar as credenciais cedidas pelo `Entity Manager` para executar determinados serviços que podem verificar a validade dessa credencial.

Figura 4 – Diagrama de classe de credenciais do cidadão.



Fonte: (MAIA, 2025).

Para esse segundo modelo, foi inserido os componentes `HolderController`, `IssuerController` e `VerifierController`. O `HolderController` tem como objetivo realizar as funções do *Holder* para o cidadão. Ou seja, o `HolderController` possibilita a gestão das carteiras de credenciais digitais dos cidadãos e também possibilita o acesso aos serviços aos quais estejam credenciados. O `IssuerController` integra o `Entity Manager` proposto por (CARDOSO, 2024), adicionando aos poderes da entidade administrativa o papel de *Issuer* que é a capacidade de emitir credenciais para os cidadãos. Já o `VerifierController` implementa o *Verifier*, possibilitando que serviços possam validar credenciais apresentadas pelos cidadãos.

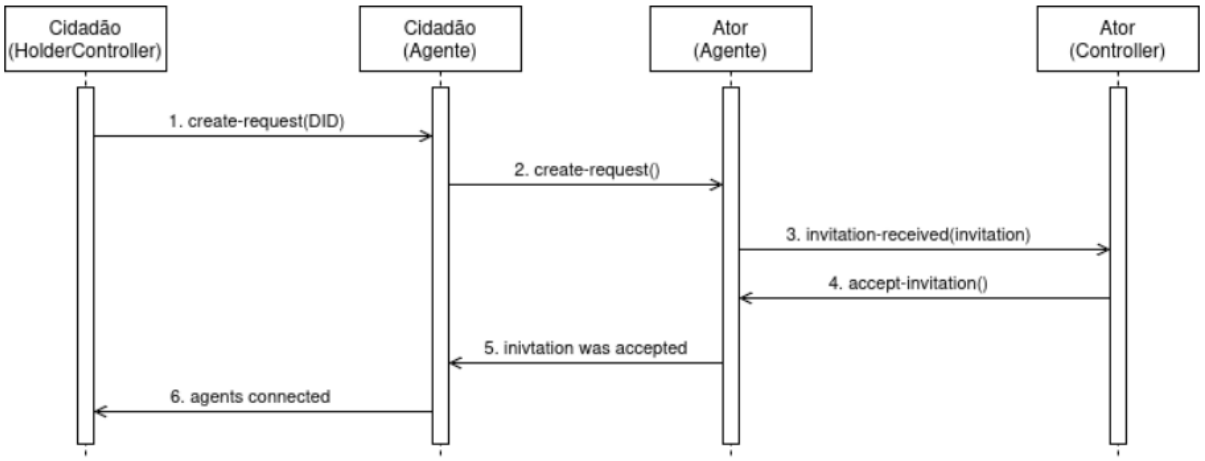
3.3.1 Fluxo de Estabelecimento de Conexão

O `HolderController` é o portador e o único gerenciador de suas credenciais. Com isso, para manter a segurança, ele deve apresentar suas credenciais apenas para instâncias

confiáveis de forma íntegra e privativa. Para se obter esse teor de segurança nas interações, é necessário que um canal seguro de comunicação entre o cidadão (`Holder`) e outros agentes da cidade seja estabelecido.

O processo para se estabelecer esse canal seguro de comunicação está ilustrado na figura 5 e é explicado mais abaixo:

Figura 5 – Diagrama de sequência representando as etapas para estabelecer um canal seguro de comunicação entre o HolderController e outro ator do modelo.



Fonte: (MAIA, 2025).

- O processo de estabelecimento de conexão tem início quando o cidadão, por meio de seu `HolderController`, envia uma solicitação de conexão para o seu respectivo agente. Essa solicitação contém o DID[2] público (Identificador Descentralizado) do agente com o qual se deseja estabelecer comunicação, sendo esse identificador obtido diretamente da blockchain **(1)**.

- Em seguida, o agente do cidadão redireciona essa solicitação para o endereço correspondente ao agente de destino **(2)**, o qual também é recuperado a partir da blockchain. Ao receber essa solicitação, o agente da outra parte interpreta-a como um convite de conexão, o que é processado pelo seu *controller* **(3)**.

- Após essa etapa, o *controller* da parte receptora aceita o convite e notifica seu próprio agente da decisão **(4)**.

- Este, por sua vez, responde ao agente solicitante informando que a solicitação de conexão foi aceita **(5)**.

[2] identificador único, autogerado e verificável, usado para representar entidades digitais em sistemas descentralizados, sem depender de uma autoridade central.

- Por fim, a conexão é estabelecida com sucesso, e o `HolderController` do cidadão é informado da conclusão do processo **(6)**, indicando que a comunicação segura entre os agentes foi efetivada.

A comunicação entre os agentes é viabilizada por meio do protocolo DIDComm[3], o qual oferece um mecanismo seguro para a troca de mensagens. Esse protocolo assegura tanto a autenticidade quanto a integridade das interações por meio do uso de mensagens criptografadas e assinadas digitalmente.

Para o estabelecimento dessa comunicação segura, os agentes realizam a troca de seus Identificadores Descentralizados (DIDs) e respectivas chaves públicas, as quais são fundamentais para os processos de criptografia e descriptografia das mensagens trocadas. Dessa forma, cria-se um canal de comunicação com criptografia de ponta a ponta, impedindo acessos indevidos e garantindo a confidencialidade das informações transmitidas.

3.3.2 Fluxo de Emissão de Credenciais

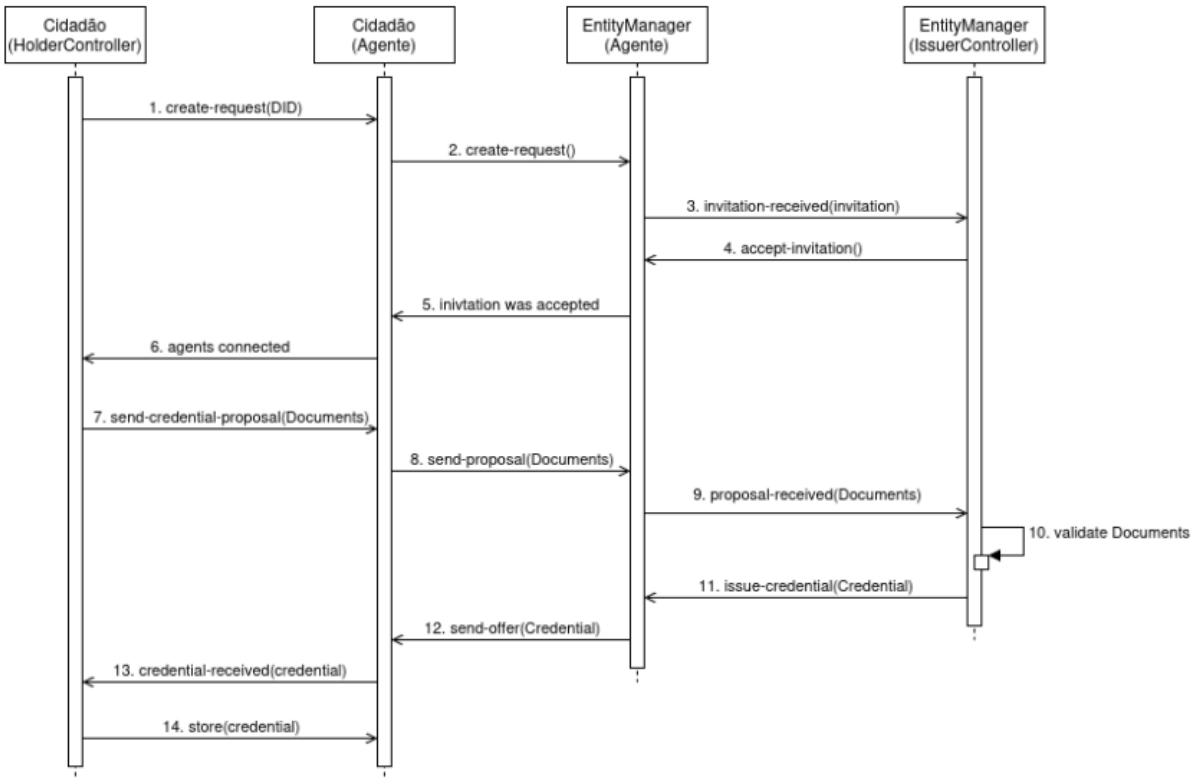
O processo de emissão de credenciais digitais começa com a criação de uma conexão segura entre o emissor (*Issuer*) e o cidadão (*Holder*). Essa etapa é fundamental para garantir que a troca de informações subsequente ocorra de forma autenticada, privada e criptograficamente protegida.

A figura 6, representa o fluxo completo de emissão de uma credencial seguindo as seguintes etapa:

- **Estabelecimento da conexão (1-6)**: Essa etapa está relacionada ao estabelecimento da conexão entre o *Holder* (cidadão) e o *Issuer* (entidade administrativa). O passo a passo dessa estapa está descrito em 3.3.1.
- **Proposta de Credencial (7-10)**:
  - Com a conexão previamente estabelecida, o `HolderController` inicia a etapa de identificação ao enviar uma proposta de credencial **(7)**. Essa proposta contém documentos do cidadão, como o CPF, que serão utilizados para a validação por parte da entidade emissora.
  - O agente responsável pelo cidadão transmite essa proposta ao agente vinculado ao `IssuerController` **(8)**.
  - Ao receber a requisição **(9)**, o `IssuerController` procede com a verificação dos dados apresentados **(10)**, assegurando a validade das informações enviadas.
- **Emissão e Persistência da Credencial**:

[3] https://didcomm.org

Figura 6 – Diagrama de sequência representando as etapas para a emissão bem sucedida de uma credencial.



Fonte: (MAIA, 2025).

– Após a validação bem sucedida, o `IssuerController` emite a credencial verificável correspondente **(11)**.

– Em seguida, o agente da entidade emissora prepara e envia uma oferta de credencial ao agente do cidadão **(12)**.

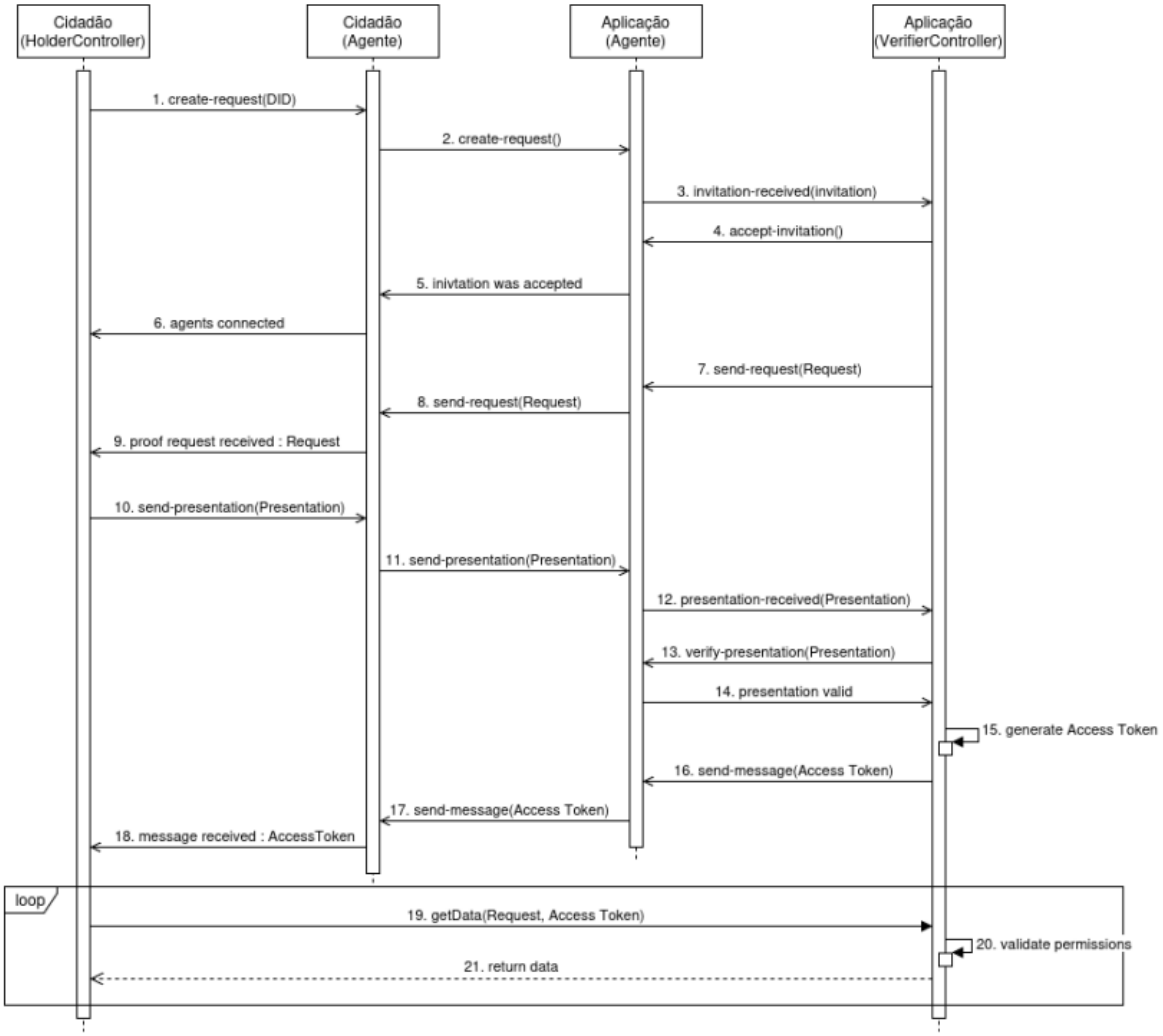
– Por fim, o agente do `HolderController` repassa essa credencial para o controlador do cidadão, que então realiza a persistência da credencial em seu armazenamento local ou carteira digital **(13-14)**, concluindo o ciclo de emissão.

3.3.3 Autenticação do Cidadão

A autenticação do cidadão é quando o cidadão apresenta a sua credencial para algum serviço a fim de receber a permissão para exercer determinado serviço. A permissão é retornada como um token JWT (JSON Web Tokens) e esse token é apresentado antes de usar determinado serviço. O serviço valida esse token através de seu `VerifierController` e se o token estiver Válido, o acesso do cidadão é permitido.

O diagrama de sequência na figura 7 ilustra o processo completo de autenticação, desde o estabelecimento da conexão até a emissão do token JWT que concede acesso ao cidadão. Detalhando cada etapa, tem-se:

Figura 7 – Diagrama de sequência representando as etapas para a autenticação bem sucedida de uma credencial.



Fonte: (MAIA, 2025).

- **Estabelecimento da Conexão (1-6)**: Essa fase trata da criação de um canal seguro de comunicação entre o *Holder* (cidadão) e o *Verifier* (microsserviço). O detalhamento completo dessa etapa encontra-se na seção 3.3.1.
- **Solicitação de Prova de Credencial (7-9)**:
  - Após a conexão ter sido devidamente estabelecida, o processo de verificação de identidade é iniciado. O `VerifierController` envia uma solicitação para que o cidadão apresente uma prova de credencial **(7)**.

  - Essa solicitação é encaminhada pelo agente do `VerifierController` ao agente do `HolderController` **(8)**, que então repassa a requisição ao próprio `HolderController` **(9)**.
  - Ao receber a solicitação, o `Holder` avalia quais credenciais disponíveis em sua posse podem ser utilizadas para responder ao pedido de forma adequada.

- **Apresentação da Prova (10-12)**:
  - O cidadão seleciona uma credencial compatível com a solicitação recebida e, por meio do HolderController, envia uma apresentação de prova **(10)**.
  - Essa apresentação é transmitida ao agente do `VerifierController` **(11)** e, em seguida, recebida pelo `VerifierController` **(12)**, que irá realizar a verificação do conteúdo apresentado.

- **Verificação da Credencial (13-15)**:
  - O `VerifierController` realiza a validação da prova recebida **(13)**, utilizando seu agente para consultar dados registrados na cadeia blockchain, assegurando a autenticidade da informação apresentada.
  - O agente executa essa verificação e retorna o resultado ao `VerifierController` **(14)**.
  - Com a prova validada, o sistema codifica as informações essenciais, como nome completo do cidadão e permissões de acesso, em um token JWT **(15)**, o qual será utilizado para autenticação nas próximas interações.

- **Envio do Token JWT (18-21)**:
  - O token JWT é encaminhado ao agente do `VerifierController` **(16)**.
  - Com isso, o token é transmitido ao agente do *Holder* **(17)**, sendo finalmente recebido pelo `HolderController` **(18)**. Com isso, o cidadão passa a possuir um token de acesso válido, que poderá ser utilizado para interações autenticadas na aplicação.

- **Realizar Serviço (19-21)**:
  - De posse do token, o cidadão utiliza o `HolderController` para realizar requisições a endpoints privados da aplicação **(19)**, empregando o token JWT para fins de autenticação.
  - A aplicação, por sua vez, valida tanto a autenticidade do token quanto as permissões associadas **(20)**, assegurando que o recurso solicitado esteja autorizado.
  - Em caso positivo, a aplicação retorna os resultados dos serviços requisitados **(21)**.

3.4 Suporte à Aplicações Seguras de Sensoriamento Participativo

3.4.1 Aplicação das Extensões de Gerenciamento de Identidades e Inserção de Dados

Apesar do InterSCity sofrer com problemas que dificultam a integração do sensoriamento participativo, se as extensões explicadas na seção 3.2 e 3.3 forem aplicadas ao InterSCity, esses problemas seriam solucionados. Isso se dá ao motivo do modelo dessas extensões ser baseado em blockchain e identidades auto-soberanas. Esses artifícios permitem o suporte ao sensoriamento participativo de forma segura, garantindo a autenticidade dos dados e a privacidade dos cidadãos. A seguir, foram descritas as contribuições das extensões que permitem a implantação da funcionalidade de sensoriamento participativo no InterSCity:

- **Canais Seguros de Comunicação**: O estabelecimento de conexões seguras entre os cidadãos e a plataforma, conforme descrito na seção 3.3.1, garante que os dados transmitidos sejam criptografados e protegidos contra interceptações. Criando um ambiente confiável para a colaboração dos cidadãos voluntários.
- **Gerenciamento de Acesso**: O `VerifierController` permite que os serviços de sensoriamento participativo validem as credenciais apresentadas pelos cidadãos, consultando a blockchain para verificar sua autenticidade. Essa abordagem descentralizada elimina a necessidade de uma autoridade central. Além disso, essa abordagem permite que apenas cidadãos que gozem da confiança das entidades administrativas possam executar determinadas tarefas.
- **Emissão de Credenciais Específicas**: O `IssuerController`, pode emitir credenciais personalizadas para diferentes tipos de sensoriamento participativo. Por exemplo, um cidadão pode receber uma credencial específica para relatar condições de tráfego de carros, enquanto outro pode ser autorizado a enviar dados sobre o tráfego de pessoas. Essa possibilidade no controle de acesso permite que a plataforma gerencie de forma eficiente as credenciais e permissões dos cidadãos.

3.4.2 IPS Lib (*InterSCity Participatory Sensing Library*)

Apesar do extensão proposta por (MAIA, 2025) ser ideal para aplicações de sensoriamento participativo, o seu funcionamento é baseado em diversas chamadas de API RESTful[4]. Isso pode causar certa dificuldade no processo de desenvolvimento dessas aplicações.

[4] interface de programação que segue os princípios do estilo arquitetural REST, permitindo a comunicação entre sistemas via requisições HTTP padronizadas.

Para facilitar o desenvolvimento dessas aplicações no contexto do InterSCity, foi desenvolvida uma biblioteca que facilita esse processo. A IPS Lib abstrai a complexidade da integração com as extensões de gestão de identidades propostas nas seções 3.2 e 3.3, criando métodos que simplificam processos essenciais do modelo. Dessa forma, é permitido que os desenvolvedores foquem na lógica de suas aplicações sem se preocupar com detalhes de autenticação e comunicação com a plataforma.

Considerando que o objetivo dessa biblioteca é facilitar a autenticação de cidadãos e o envio de dados de sensoriamento participativo para o InsterSCity de forma de segura, é esperado que os softwares que usem essa biblioteca sejam aplicações móveis. Com isso, a linguagem Java[5] foi escolhida para o desenvolvimento da IPS Lib. A linguagem Java é conhecida pela sua portabilidade, robustez e ampla adoção no mercado. Além disso, as bibliotecas Java também tem interoperabilidade com diversas outras linguagens presentes no mercado, inclusive com Kotlin[6] que é uma das linguagens mais utilizadas no âmbito de desenvolvimento móvel.

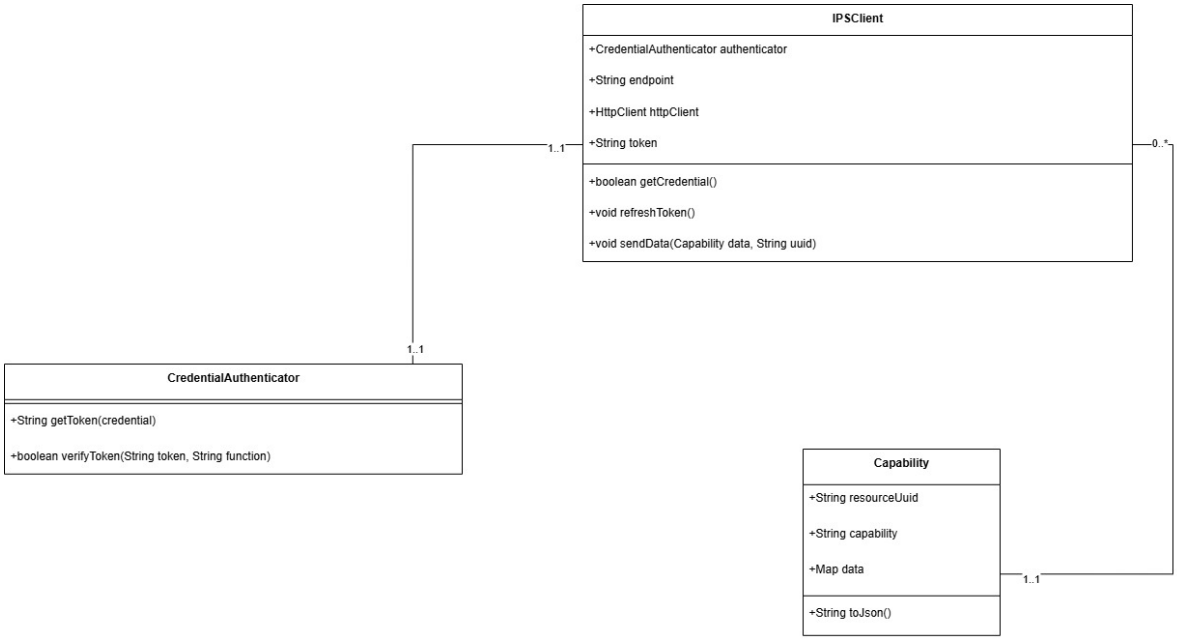
3.4.2.1 Estrutura da IPS Lib

A biblioteca IPS Lib é composta por três classes principais exemplificadas no diagrama de classes representado na figura 8, cada uma responsável por uma funcionalidade específica dentro do processo da biblioteca:

- **IPSClient**: Essa classe central gerencia a comunicação com o serviço de sensoriamento participativo e as outras classes. Suas funções incluem a autenticação, envio de dados e tratamento de erros. Ela garante que as requisições sejam autenticadas usando tokens JWT obtidos através do modelo de identidades auto-soberanas.
- **CredentialAuthenticator**: Responsável por realizar o fluxo autenticação do usuário utilizando o modelo SSI proposto por (MAIA, 2025). Essa classe implementa o fluxo completo de obtenção de um token de acesso, desde a criação da conexão até a apresentação da credencial e recebimento do token JWT, assim como exemplificado na seção 3.3.3.
- **Capability**: Representa uma capacidade no contexto do InterSCity, permitindo a modelagem flexível dos dados a serem enviados. Essa classe facilita a construção de objetos JSON que serão transmitidos para a plataforma, suportando campos dinâmicos e tipos de dados variados, como strings, números e instantes de tempo.

[5] https://www.java.com/pt-BR
[6] https://kotlinlang.org

Figura 8 – Diagrama de classe da Biblioteca IPS Lib.



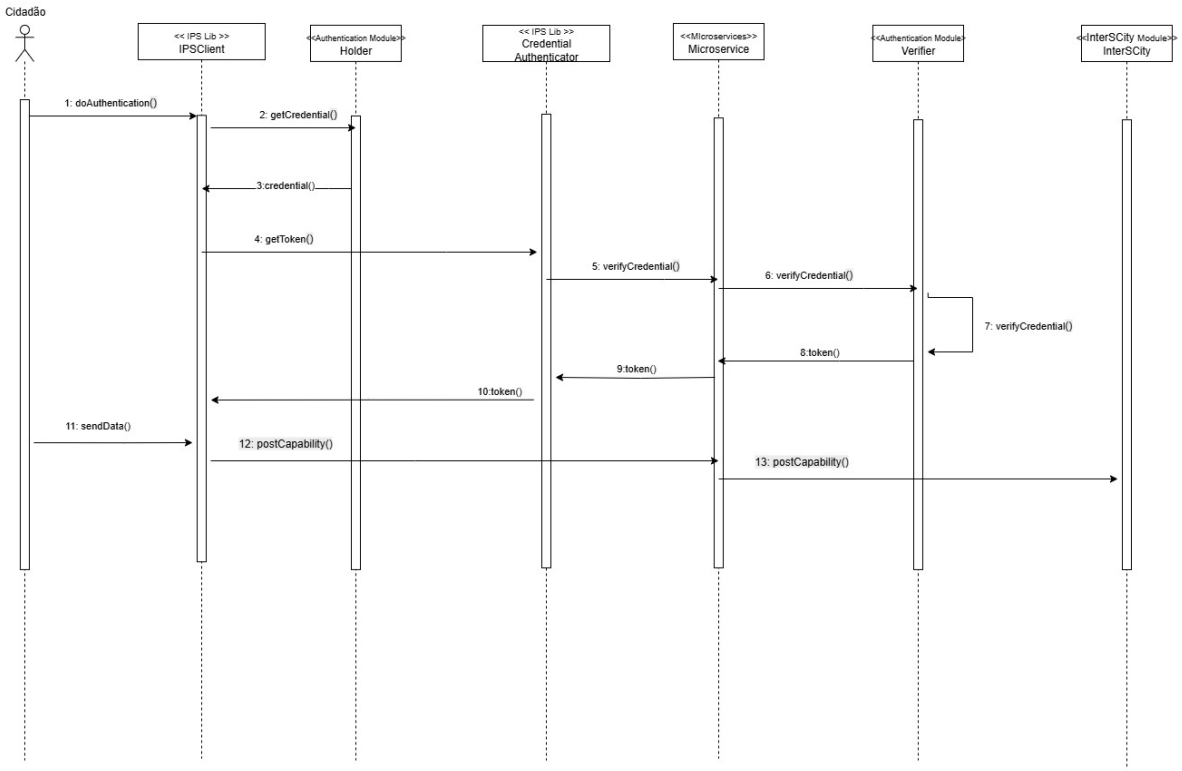
Fonte: Autor.

3.4.2.2 Fluxo de Autenticação e Envio de Dados

O diagrama da figura 9 ilustra o processo de autenticação e envio de dados no modelo de sensoriamento participativo, integrando a biblioteca Java desenvolvida com os componentes do InterSCity e do sistema de identidades auto-soberanas (SSI). A sequência é detalhada a seguir:

- No início, o cidadão utiliza uma aplicação em seu dispositível móvel que aplica a biblioteca IPS Lib por meio da classe `IPSClient` **(1)**.
- A biblioteca solicita a credencial do cidadão através do método `getCredential()` **(2-3)** e a envia ao `CredentialAuthenticator` **(4)**.
- O `CredentialAuthenticator` inicia o fluxo de autenticação visto na seção 3.3.3 **(5)**, solicitando um token ao serviço de identidade.
- O Verifier valida a credencial **(6-7)** e retorna um token JWT **(8)**, que é armazenado localmente **(9-10)**.
- Com o token válido, o cidadão envia os dados de sensoriamento via método `senData()` **(11)**.
- O IPSClient transmite os dados ao microsserviço usando o método `postCapability()` **(12)**, que valida o token antes de processar a requisição **(13)**.

Figura 9 – Diagrama de Sequência da Biblioteca IPS Lib.



Fonte: Autor.

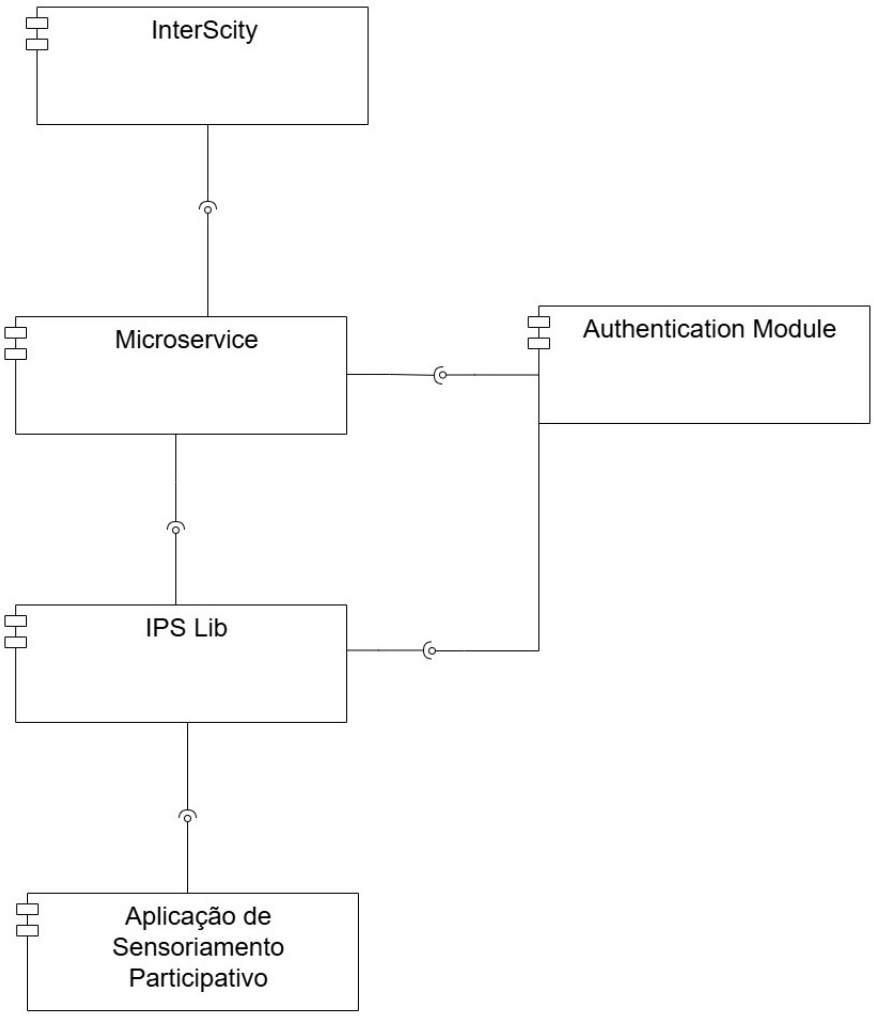
3.4.3 Arquitetura de Sensoriamento Participativo no InterSCity

A fim de que o InterSCity possa suportar plenamente um software de sensoriamento participativo, é necessário integrar as extensões de gestão de identidades propostas por (CARDOSO, 2024) e (MAIA, 2025) à plataforma. Essa integração permite que os cidadãos, atuando como agentes ativos, enviem dados de forma segura e autenticada, garantindo a confiabilidade e a privacidade das informações compartilhadas.

Para o funcionamento correto de uma aplicação de sensoriamento participativo nessa versão estendida do InterSCity, é necessário que se use uma arquitetura de microsserviços. A ideia por trás do uso dessa arquitetura é que cada microsserviço dessa aplicação realize a vericação de tokens recebidos através do `VerifierController`.

Com isso, foi projetada uma arquitetura, ilustrada no diagrama de componentes da figura 10, ela consiste na utilização de cinco elementos principais: o Authentication Module, a IPS Lib, os microsserviços da aplicação, o front-end da aplicação de sensoriamento participativo e o InterSCity. A seguir, será explicado cada componente dessa arquitetura proposta:

Figura 10 – Diagrama de componentes representando a arquitetura proposta para aplicações de sensoriamento participativo no InterSCity.



Fonte: Autor.

- `Authentication Module`: O `Authentication Module` é responsável por gerenciar as identidades dos cidadãos e dispositivos no ecossistema do InterSCity. Ele incorpora o componente `Verifier`, conforme descrito na Seção 3.3. Suas funções incluem:
  - **Autenticação**: O `VerifierController` valida as credenciais apresentadas pelos cidadãos antes que estes possam acessar os serviços da aplicação.
  - **Validação**: O `Verifier` pode realizar validações de token, para dizer se a credencial representada naquele token é válida para acessar determinado serviço.

  Esse módulo utiliza a tecnologia blockchain para garantir a imutabilidade e a auditabilidade das identidades, seguindo o modelo hierárquico proposto por (CARDOSO, 2024), onde cada identidade é emitida por uma entidade administrativa confiável.

- `IPS Lib`: A IPS Lib é uma biblioteca desenvolvida para simplificar a integração de aplicações de sensoriamento participativo com o InterSCity. Ela, ao ser acoplada nessas aplicações, abstrai a complexidade dos fluxos de autenticação e comunicação segura. Sua principal função é possibilitar uma autenticação simplificada. A classe `CredentialAuthenticator` implementa o fluxo completo de autenticação descrito na Seção 3.3.3. Dessa maneira, esse módulo se comunica diretamente com o `Authentication Module`

- `Aplicação de Sensoriamento Participativo`: A aplicação móvel, que utiliza a IPS Lib, é a interface entre o cidadão e o InterSCity. Ela funciona como um front-end coletor de dados, após coletar ela repassa para o `IPSClient` da IPS Lib que realiza o envio desses dados.

- `Microsserviços da Aplicação`: É a representação dos microsserviços da aplicação, eles recebem requisições do uso de determinados serviços. Cabe a esses microsserviços se comunicarem com o `Authentication Module` para validar o acesso e então concedem a permissão ao serviço se o cidadão estiver credenciado. Se o serviço for de envio de dados provenientes do sensoriamento participativo para a cidade inteligente, cabe a esse módulo enviar os dados ao InterSCity.

- `InterSCity`: Esse componente representa uma instância do InterSCity (explicado anteriormente na seção 3.1), ele recebe os dados provenientes dos microsserviços da aplicação e persiste esses dados. Esses dados ainda podem ser consumidos por outras aplicações.

A adoção dessa arquitetura traz benefícios significativos para o ecossistema de cidades inteligentes baseados no InterSCity. Ao implementar o modelo de gerenciamento de identidades, a plataforma passa a oferecer um ambiente seguro para o desenvolvimento de aplicações urbanas inovadoras. Essa solução não apenas viabiliza o uso das identidades auto-soberanas dos cidadãos, conforme proposto por (MAIA, 2025), mas também estabelece as bases necessárias para implantação em larga escala do sensoriamento participativo em ambientes urbanos reais.

A integração direta com o InterSCity garante que os dados coletados sejam incorporados ao ecossistema da cidade inteligente, onde podem ser processados, analisados e disponibilizados para outras aplicações e serviços urbanos. Essa interoperabilidade facilita a criação de uma grande gama de aplicações diversificadas voltadas tanto para órgãos públicos quanto para cidadãos comuns.

3.5 Estudo de Caso: Spotter

Com base nos fundamentos teóricos e arquiteturais estabelecidos nas seções anteriores, foi desenvolvido um estudo de caso prático com o objetivo de validar a aplicabilidade da arquitetura de sensoriamento participativo apresentado na seção 3.4.3. A aplicação desenvolvida foi incorporada a uma instância InterSCity que aplica as extensões descritas nas seções 3.2 e 3.3.

A aplicação em questão foi denominada `Spotter`, ela consiste em um sistema de rastreamento de veículos com a participação cidadã. Nessa aplicação, os moradores da cidade atuam como sensores móveis, utilizando seus dispositivos pessoais para enviar dados de imagem, placas e localização de veículos procurados pela gestão pública.

Conforme descrito na seção 3.1, o ecossistema do InterSCity funciona baseado em recursos e capacidades. Para que a funcionalidade de rastreamento de veículos possa ser adicionado a esse ambiente se considerou os veículos procurados como recursos da cidade e esses recursos, do tipo veículo, têm por capacidade a sua localização. A localização apresenta em sua estrutura as coordenadas do veículo, a data da captura dessa localização e uma imagem do veículo naquele espaço.

3.5.1 Tagger

Nativamente, o InterSCity não possui mecanismos para atribuir características aos seus recursos. Como o `Spotter` é uma aplicação que atua diretamente com os veículos, é necessário que haja formas de se atribuir características para os recursos do InterSCity. Para resolver essa problema, foi utilizado o `Tagger`, que é uma extensão do InterSCity desenvolvida para enriquecer a gestão dos recursos.

O `Tagger` adiciona ao InterSCity marcadores personalizáveis (tags). Ele atua como uma camada de metadados, permitindo associar características dinâmicas a cada recurso cadastrado. Essa adição permite diversas possibilidades como realizar buscas complexas, classificar e realizar controle de estado nos recursos.

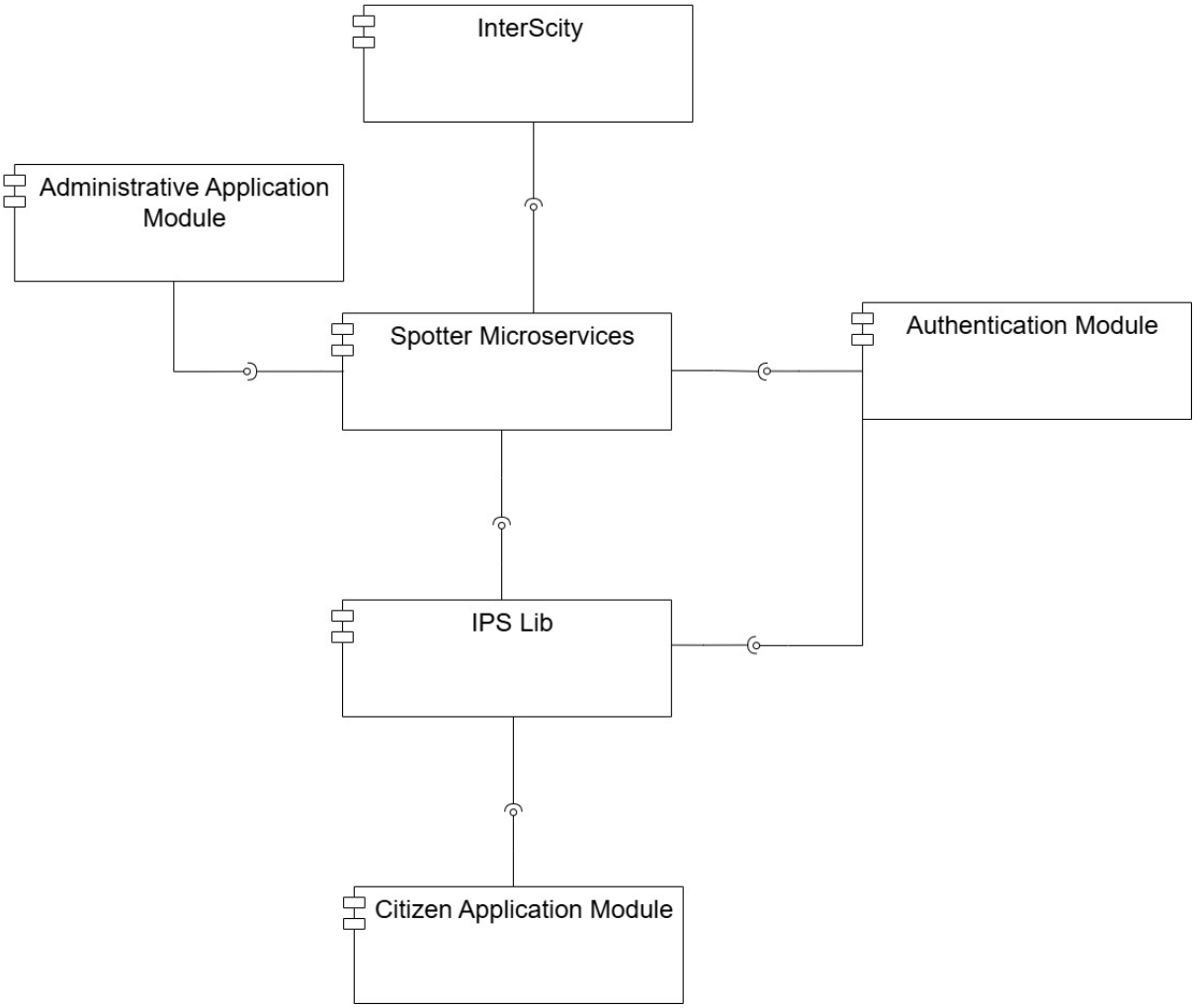
3.5.2 A Arquitetura do Spotter

Definir a arquitetura no desenvolvimento do sistema é crucial para garantir segurança, qualidade e eficiência (ANIKEEVA; SELIFANOV, 2022). Portanto, essa é uma parte essencial para se conceber um sistema eficientemente seguro e íntegro. Em termos arquiteturais, a aplicação foi dividida em dois módulos principais: o **módulo do cidadão**, voltado à coleta de dados, e o **módulo administrativo**, direcionado a usuários autenticados com privilégios mais elevados (como agentes públicos ou policiais). Essa divisão reflete diretamente os papéis estabelecidos no modelo de identidade auto-soberana,

onde os cidadãos atuam como *Holders* de suas credenciais, enquanto os serviços públicos desempenham o papel de emissão de credenciais para o cidadão.

Com isso, foi adaptado a arquitetura proposta na seção 3.4.3 para que ela pudesse ser condizente com o projeto do `Spotter`. Na figura 11 está representado um diagrama de sequência que exemplifica a arquitetura do `Spotter` e a seguir os módulos dessa arquitetura serão detalhados:

Figura 11 – Diagrama de componentes representando a arquitetura proposta para o estudo de caso "Spotter".



Fonte: Autor.

- **IPS Lib**: Esse módulo foi descrito na seção 3.4.3
- **Spotter Microsservices**: Esse módulo apresenta os microsserviços do `Spotter`, tanto os utilizados pela aplicação administrativa, quanto os usados pela aplicação do cidadão. A função desse módulo está descrito na seção 3.4.3
- **Authentication Module**: Esse módulo está descrito na seção 3.4.3

- **InterSCity**: Esse módulo foi aprofundado na seção 3.4.3
- **Módulo de Aplicação do Cidadão**: Esse módulo é voltado aos cidadãos, ele é uma aplicação móvel que permite que com a câmera, o smartphone possa capturar a placa imagens dos veículos. Após isso, ele pede para um microsserviço a lista dos veículos procurados e verifica se o carro capturado está sendo procurado.

  Se o veículo estiver sendo procurado, a aplicação inicia o envio de imagens e da geolocalização da onde o veículo foi avistado. Para realizar o envio de dados, esse módulo utiliza diretamente a `IPS Lib` para realizar a autenticação do cidadão e o envio de dados para o microsserviço descrito em **(2)**. A seguir, está descrito os microsserviços que devem ser acessados por esse módulo. Com isso, para que o cidadão esteja habilitado a realizar esse serviço de sensoriamento participativo (rastreamento de veículos), ele deve ter uma credencial que permita os serviços abaixo:

  - **Consultar Veículos Procurados (1)**: Esse microsserviço retorna uma lista dos veículos que estão sendo procurados pela administração da cidade. Alimentar e gerenciar essa lista é de competência da entidade administrativa responsável e os dados dessa lista estão persistidas no InterSCity.
  - **Adicionar Localização de Avistamento (2)**: Esse serviço é responsável por enviar os dados provenientes do rastreamento de veículos para o InterSCity, é um intermediador do fluxo de dados do cidadão para o InterSCity. Sua função é verificar a validade da credencial do cidadão antes de enviar seus dados ao InterSCity.

- **Módulo de Aplicação Administrativa:** O módulo administrativo do Spotter é destinado a agentes públicos (como policiais e gestores da cidade inteligente), permitindo o gerenciamento avançado do sistema de rastreamento de veículos.

  Diferentemente do módulo de cidadãos, que tem foco na coleta de dados, este módulo é responsável por permitir o gerenciamento de veículos procurados (adicionando ou removendo veículos da lista de procurados) e consumir os dados que foram recebidos do sensoriamento participativo. Os microsserviços relacionados a esse módulo são:

  - **Marcar como Procurado (3)**: Adiciona através do `Tagger` a tag de procurado a um veículo já cadastrado no sistema, sinalizando que ele é de interesse para as autoridades.
  - **Desmarcar como Procurado (4)**: Remove a tag de procurado de um veículo, indicando que ele não é mais uma prioridade de rastreamento.
  - **Adicionar Carro para a Lista de Procurados (5)**: Esse serviço cadastra um novo veículo no sistema e automaticamente o marca como procurado. Necessita

de metadados como placa, cor, modelo e marca para efetivar o cadastro. Esses dados são tags que são implementadas pelo `Tagger`.

- **Consulta Localização de Veículo (6)**: Retorna o histórico completo dos dados de avistamentos de um veículo específico, incluindo as coordenadas geográficas, datas/horários e imagens associadas.
- **Obter Filtros (7)**: Retorna uma lista com todos os filtros disponíveis no `Tagger`. Essa lista é útil para saber quais são os critérios disponíveis para realizar buscas no sistema.
- **Pesquisar Veículos (8)**: Realiza uma busca por veículos com base em combinações de filtros ou uma sequência de caracteres da placa do veículo.
- **Consultar Todos os Veículos (9)**: Lista todos os veículos cadastrados no sistema, seja eles marcados como procurados ou não.

3.5.3 Desenvolvimento do Estudo de Caso

Primeiramente, para o desenvolvimento do estudo de caso, se usou uma instância segura do InterSCity que aplicasse as extensões de (CARDOSO, 2024) e (MAIA, 2025). Também se tornou necessário que a instância tivesse a extensão `Tagger` instalada. Com isso, se iniciou o desenvolvimento dos outros módulos da aplicação.

3.5.3.1 O framework Flask

O desenvolvimento da aplicação Spotter seguiu uma abordagem modular e orientada a microsserviços, utilizando a linguagem Python[7] e o framework Flask[8] para definir os microsserviços do `Spotter`. O Flask é um microframework web leve e minimalista, amplamente utilizado para a construção de APIs RESTful e aplicações web. Sua arquitetura enxuta permite que os desenvolvedores tenham controle total sobre o fluxo da aplicação, o que é particularmente útil em projetos com requisitos específicos, como a integração com microsserviços externos, autenticação personalizada e publicação segura de dados.

A escolha do Flask se deu, principalmente, pela sua facilidade de uso, ampla documentação e pela compatibilidade com o ecossistema Python. Diferente de frameworks mais robustos como o Django[9], que seguem uma estrutura mais rígida, o Flask permite maior liberdade arquitetural, isso foi importante para adaptar os serviços às particularidades da biblioteca IPS Lib e aos fluxos de identidade definidos pela arquitetura do InterSCity. Além disso, por ser um framework leve, o Flask apresenta um desempenho satisfatório. Essa combinação de simplicidade, flexibilidade

[7] https://www.python.org
[8] https://flask.palletsprojects.com/en/stable
[9] https://www.djangoproject.com

e compatibilidade justificou sua adoção como base para o desenvolvimento de todos os serviços da aplicação.

3.5.3.2 Construção dos Microsserviços

Assim como visto na seção 3.5.2, os microsserviços do `Spotter` são destinado a dois grupo de atores: cidadãos e agentes públicos. Com isso, foram criadas duas credenciais, uma para os cidadão normais e outra para os agentes administrativos. Essas credenciais permitem o acesso dos atores aos seus respectivos microsserviços que foram descritos na arquitetura.

Cada ação dos microsserviços foram protegidas por autenticação baseada em credenciais auto-soberanas, seguindo o fluxo de verificação via *VerifierController*, conforme descrito anteriormente na seção 3.3.3. Isso garantiu que somente usuários com credenciais válidas pudessem acessar os serviços correspondentes ao seu perfil.

Além disso, foi utilizada a extensão Tagger descritas em 3.5.1 para registrar informações semânticas dos veículos, como placa, modelo, cor e ano, facilitando o processo de identificação e organização dos dados. Dessa forma foi possível implementar os microsserviços **(1)**, **(3)**, **(4)**, **(5)**, **(7)**, **(8)** que foram descritos na seção 3.5.2.

Os microsserviços do `Spotter` foram construídos com endpoints REST, respeitando princípios de separação de responsabilidades e reaproveitamento de código. Com a estrutura implementada, tornou-se possível validar as interações básicas da aplicação e garantir que os fluxos de autenticação, publicação e consulta de dados fossem realizados com segurança e rastreabilidade.

3.5.3.3 Construção do Módulo da Administração e do Cidadão

A publicação de dados na plataforma InterSCity por parte do cidadão foi feita utilizando a IPS Lib, que encapsula as chamadas para envio de capacidades. Essa biblioteca simplificou o desenvolvimento da comunicação entre os microsserviços da aplicação e a infraestrutura da cidade.

Também foi construído uma Plataforma para ser acessada pelos agentes administrativos para adicionar, remover carros da lista de procurados, localizar e pesquisar veículos. Essa plataforma foi desenvolvida utilizando o framework Flask para seu back-end e JavaScript[10] em seu front-end. o back-end dessa aplicação faz requisições diretamente aos microsserviços da parte administrativa do `Spotter` e apresenta ao agente no front-End.

[10] https://developer.mozilla.org/en-US/docs/Web/JavaScript

A implementação desse estudo de caso comprova que a arquitetura de sensoriamento participativo proposta em 3.4.3 pode ser aplicada e estendida para outras aplicações participativas dentro do ecossistema InterSCity.

3.6 Experimentos e Validação do Estudo de Caso

A realização de experimentos e validações práticas é uma etapa fundamental no desenvolvimento de sistemas, especialmente quando envolvem dados sensíveis e autenticação de usuários com base em credenciais descentralizadas. No contexto deste trabalho, torna-se essencial avaliar se a aplicação desenvolvida é capaz de operar de forma satisfatória em um contexto de CI. A validação busca evidenciar o valor prático da proposta, demonstrando que cidadãos e agentes públicos podem interagir com o sistema de forma eficiente, segura e auditável.

3.6.1 Latência de autenticação de credencial

O objetivo desse experimento é medir o tempo médio da autenticação de uma credencial do cidadão. Para esse experimento, foi utilizado o estudo de caso `Spotter`. Além disso, foi também criado em rede local uma instância de `HolderController` que tinha consigo uma credencial que permite o acesso a todos os serviços do `Spotter`, após isso, o experimento de autenticar uma credencial foi executado 20 vezes, a cada repetição se registrava o tempo que demorou para realizar o experimento.

A Tabela 1 apresenta os dados estatísticos obtidos ao fim da bateria de experimentos. Já na Figura 12, nós temos um gráfico que apresenta todos os tempos obtidos em cada repetição do experimento. Pode-se verificar que os tempos obtidos não tiveram uma grande variação. Logo, como o tempo médio é de apenas um segundo e meio aproximadamente e essa operação é executada apenas na inicialização da aplicação, considera-se que a solução possui um desempenho satisfatório.

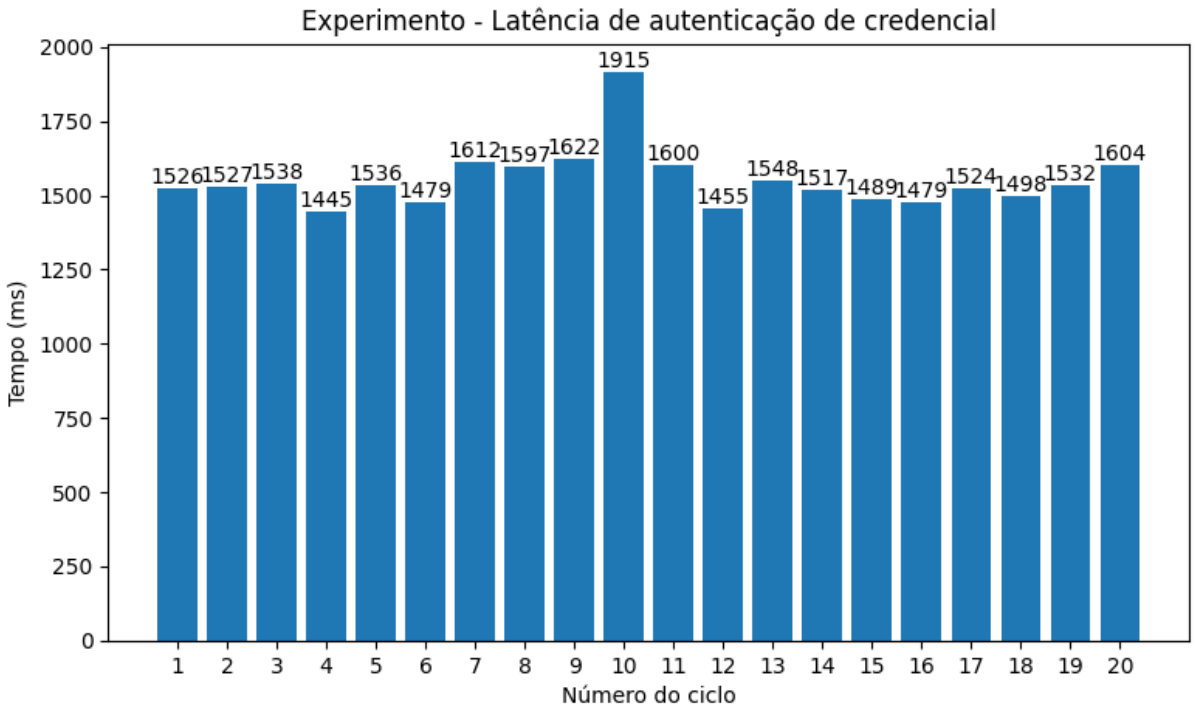
Tabela 1 – Tempo médio para autenticação de credencial

Estatística	**Latência**
Média	1552.1ms
Mediana	1529.5ms
Máximo	1915.0ms
Mínimo	1445.0ms
Desvio Padrão	97.4

3.6.2 Latência no compartilhamento de dados ao Spotter e ao InterSCity

Já nesse experimento, se teve como objetivo medir a latência média de comunicação da aplicação de sensoriamento participativo com o microsserviço `Adicionar Localização`

Figura 12 – Latência média de autenticação de credencial.



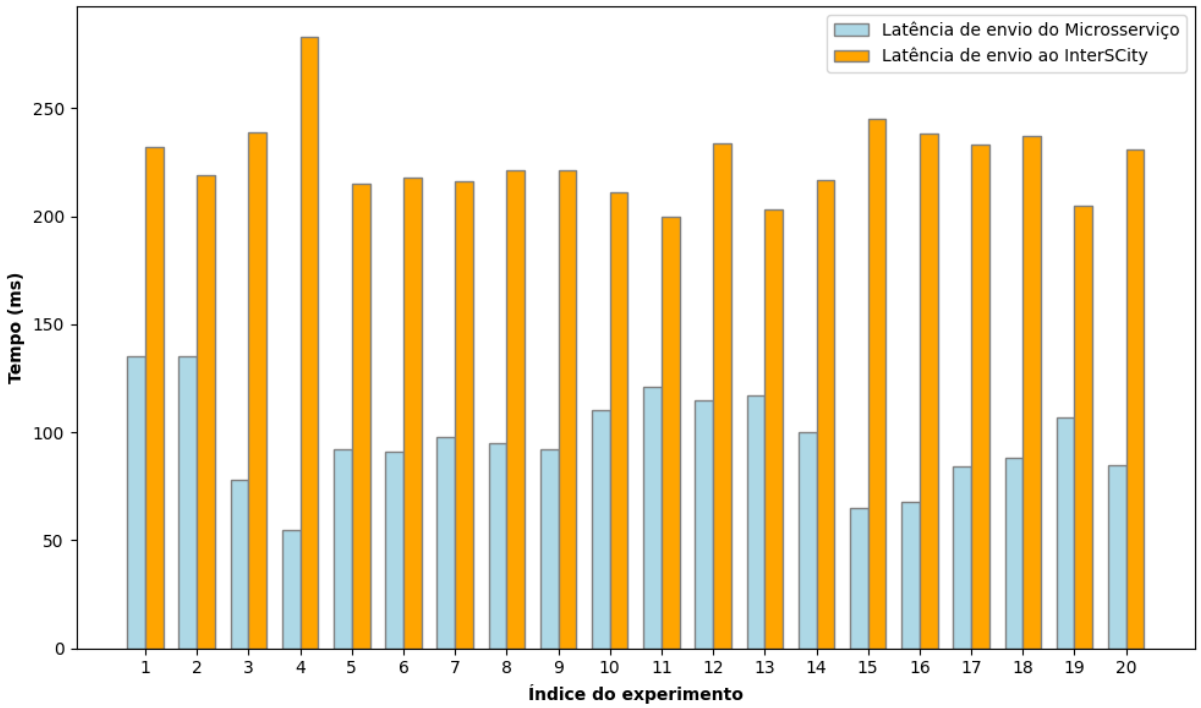
Fonte: Autor.

`de Avistamento` **(2)** do `Spotter`. Esse microsserviço realiza o sensoriamento participativo ao transmitir dados sobre a localização do veículo procurado, a data do avistamento e de dados de multimídia, por exemplo, uma imagem comprovando o avistamento de determinado veículo na determinada localização. Também se teve como objetivo medir a latência média de comunicação dos dados desse microsserviço com o InterSCity seguro.

De forma semelhante ao experimento anterior, nesse experimento também foi executado em rede local uma instância de `HolderController` que tinha consigo uma credencial que permissiona o acesso a todos os serviços do `Spotter`. Após isso, o experimento foi executado 20 vezes, onde em cada repetição foi requisitado o uso do microsserviço de avistamento de veículo procurado. Assim, foi possível realizar a medição da latência média para que os dados recebidos do microsserviço pudessem ser processados e enviados ao InterSCity.

A Tabela 2 apresenta os dados de Média, Mediana, Máximo, Mínimo e Desvio Padrão dos dados de latência recebidos por cada ciclo de experimento. O gráfico da Figura 13 mostra todos os dados oriundos dos experimentos, onde cada experimento apresenta como resultado duas barras. A barra azul apresenta o tempo gasto pelo `Spotter` para aceitar a requisição, verificar a autenticidade do *token* recebido e tratar os dados recebidos. Já a barra laranja, apresenta o tempo gasto para autenticar e enviar os dados para o InterSCity seguro. Um resumo do fluxo de informação dessas etapas de processamento foi esquematizado na Figura 14.

Figura 13 – Gráfico de latência no compartilhamento de dados ao Spotter e ao InterSCity.

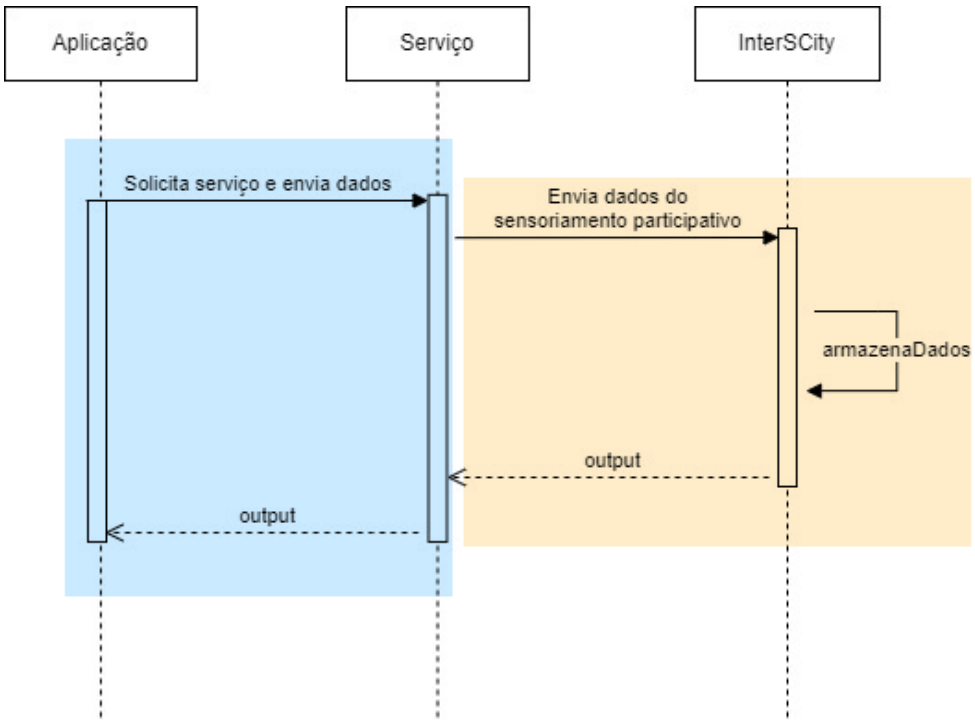


Fonte: Autor.

Tabela 2 – Gráfico de latência no compartilhamento de dados ao Spotter e ao InterSCity

Estatística	**Latência do InterSCity**	**Latência do Microsserviço**
Média	322.4ms	225.9ms
Mediana	317.0m	221.0ms
Máximo	367.0ms	283.0ms
Mínimo	306.0ms	200.0ms
Desvio Padrão	16.5	18.5

Figura 14 – Diagrama de sequência que apresenta o fluxo de dados do Spotter ao InterSCity.



Fonte: Autor.

4 Conclusão e Trabalhos Futuros

Ao longo desse trabalho de conclusão de curso, foi desenvolvido um modelo arquitetural que aplica extensões de segurança para aplicações que realizam sensoriamento participativo de dados no InterSCity. Com isso, também, foram discutidos os principais conceitos relacionados à identidade digital e que são importantes para possibilitar o sensoriamento participativo no InterSCity. Além disso, a IPS Lib que é uma biblioteca com o objetivo de facilitar o desenvolvimento de aplicações de sensoriamento participativo foi desenvolvida.

Dessa forma, foi possível desenvolver o caso de uso `Spotter`, que é uma aplicação com o objetivo de realizar rastreamento de veículos de forma participativa no contexto de cidades inteligentes e além disso, transmitir dados de multimídia de forma segura e confiável por pessoas estritamente autorizadas a realizar esse serviço. Essa aplicação também é baseada em microsserviços que funcionam de forma segura e protegida. Também foram realizados experimentos com o `Spotter` que validaram o modelo que foi proposto, atestando assim que o modelo consegue realizar sua proposta.

4.0.1 Trabalhos Futuros

Para um funcionamento ainda mais efetivo da posposta no contexto de cidades inteligentes, futuramente se planeja focar em alguns pontos que não foram tocados no decorrer do projeto. Abaixo, destacam-se os principais pontos a serem explorados:

- **Processo de alistamento do cidadão**: Ainda não foi definida uma estratégia clara de como um cidadão pode solicitar ou receber uma credencial digital que permita sua participação em serviços específicos da cidade inteligente. A especificação desse fluxo é essencial para garantir o funcionamento adequado do sistema em situações reais.

- **Verificação da veracidade das informações enviadas**: A aplicação atual confia que cidadãos credenciados enviarão dados legítimos, porém, não foram implementados mecanismos para verificar se os registros (como imagens ou localizações) são autênticos ou forjados. Investigar técnicas para validação automática ou cruzada de dados é uma etapa fundamental para promover a confiança no sistema.

- **Incentivo à participação cidadã**: Outro desafio é a ausência de mecanismos que estimulem ativamente a participação voluntária dos cidadãos no sensoriamento participativo. Estratégias como gamificação, recompensas, ou integração com políticas

públicas podem ser consideradas como formas de aumentar o engajamento e a efetividade da coleta de dados.

- **Mecanismos de Reputação do cidadão**: Considerando a implementação de mecanismos de checagem de veracidade de dados, é interessante que haja formas de pontuar a reputação dos cidadãos. Essa pontuação funciona de forma que cidadãos com histórico contínuo de disseminação de informações falsas tenham uma reputação com baixa pontuação. Por outro lado, cidadãos com um histórico de informações verídicas poderiam ter uma reputação com alta pontuação.

Referências

ALI, V.; NORMAN, A. A.; AZZUHRI, S. R. B. Characteristics of blockchain and its relationship with trust. *Ieee Access*, IEEE, v. 11, p. 15364–15374, 2023. Citado na página 16.

ANIKEEVA, V. V.; SELIFANOV, V. Risk assessment in the process of determining the system architecture. , 2022. Citado na página 36.

BHATTACHARYA, M. P.; ZAVARSKY, P.; BUTAKOV, S. Enhancing the security and privacy of self-sovereign identities on hyperledger indy blockchain. In: IEEE. *2020 International Symposium on Networks, Computers and Communications (ISNCC)*. [S.l.], 2020. p. 1–7. Citado 2 vezes nas páginas 17 e 18.

BRITO, B. de B.; FREITAS, S. C. L. de; SENHORINI, K. C. C. O.; SILVA, J. C. da. Cidades inteligentes energia e sustentabilidade. *Academic Journal on Computing, Engineering and Applied Mathematics*, v. 4, n. 2, p. 21–24, 2023. Citado na página 12.

CARDOSO, A. L. A. Um modelo de gestão de identidades para cidades inteligentes baseado na tecnologia blockchain. Universidade Federal do Maranhão, 2024. Citado 7 vezes nas páginas 7, 22, 23, 24, 33, 34 e 39.

CONNOLLY, M.; DUSPARIC, I.; IOSIFIDIS, G.; BOUROCHE, M. Privacy aware incentivization for participatory sensing. *Sensors*, 2019. Citado na página 13.

ESPOSTE, A. d. M. d.; KON, F.; COSTA, F. M.; LAGO, N. Interscity: A scalable microservice-based open source platform for smart cities. In: *Proceedings*. [S.l.: s.n.], 2017. Citado 2 vezes nas páginas 20 e 21.

GUPTA, B. Understanding blockchain technology: How it works and what it can do. *Metaverse Basic and Applied Research*, v. 1, p. 18–18, 2022. Citado na página 16.

Instituto Brasileiro de Geografia e Estatística. *Panorama da Urbanização*. 2025. <https://agenciadenoticias.ibge.gov.br/agencia-noticias/2012-agencia-de-noticias/noticias/41901-censo-2022-87-da-populacao-brasileira-vive-em-areas-urbanas>. Acesso em: 6 jul. 2025. Citado na página 12.

Ding James. *Securing middleware data access*. 2020. Citado na página 14.

JØSANG, A.; POPE, S. User centric identity management. In: BRISBANE, QLD. *AusCERT Asia Pacific information technology security conference*. [S.l.], 2005. v. 22, p. 2005. Citado na página 17.

KAUR, J.; DABAS, D. Literature review of smart contracts using blockchain technology. In: SPRINGER. *International Workshop on New Approaches for Multidimensional Signal Processing*. [S.l.], 2022. p. 171–187. Citado na página 16.

LAMPROPOULOS, K.; KYRIAKOULIS, N.; DENAZIS, S. Identity management through a global discovery system based on decentralized identities. *arXiv preprint arXiv:2202.06394*, 2022. Citado na página 17.

MAIA, D. I. S. *Autenticação e Controle de Acesso Baseados em Blockchain e Identidades Autosoberanas para Cidades Inteligentes*. 53 p. Dissertação (Monografia (Bacharelado em Ciência da Computação)) — Universidade Federal do Maranhão, São Luís, 2025. Citado 9 vezes nas páginas 24, 25, 27, 28, 30, 31, 33, 35 e 39.

MELO, M. E. B. Blockchain. *Advances in computer and electrical engineering book series*, 2021. Citado na página 16.

Organização das Nações Unidas. *Relatório Mundial das Cidades 2022*. 2022. <https://brasil.un.org/pt-br/188520-onu-habitat-populaÃğÃčo-mundial-serÃą-68-urbana-atÃŀ-2050>. Acesso em: 6 jul. 2025. Citado na página 12.

PIERRO, M. D. What is the blockchain? *Computing in Science & Engineering*, IEEE, v. 19, n. 5, p. 92–95, 2017. Citado na página 16.

ZHANG, R.; XUE, R.; LIU, L. Security and privacy on blockchain. *ACM Computing Surveys (CSUR)*, ACM New York, NY, USA, v. 52, n. 3, p. 1–34, 2019. Citado na página 17.