

# Internet das Coisas Aplicada ao Gerenciamento de Presença e Encontros de Pessoas em Prédios Inteligentes

Internet of Things Applied to Presence and Meeting Management of People in Smart Buildings

Thiago Wallass, André  
Cardoso, Alysson Silva,  
Daniel Carvalho,  
Francisco Silva  
LSDi\*  
Universidade Federal do  
Maranhão

Markus Endler  
LAC  
PUC-Rio

Marcos Roriz  
Universidade Federal de  
Goiás

Ariel Soares Teles  
LSDi  
Instituto Federal do Maranhão

## RESUMO

Com o avanço tecnológico e a popularização da Internet das Coisas (IoT), aplicações para Cidades Inteligentes tem sido mais frequentes, em particular, nos prédios inteligentes, que utilizam sensores para melhor gerir seus recursos. Uma das áreas de atuação da IoT, no contexto de prédios inteligentes, consiste na gestão de presença e encontros de pessoas em ambientes fechados. A detecção de pessoas em ambientes internos torna-se relevante, especialmente em tempos de pandemia, em que é fundamental identificar os locais que uma pessoa esteve, por quanto tempo e se houve encontro entre pessoas. Este trabalho apresenta uma arquitetura de software distribuída que serve de base para o desenvolvimento de aplicações que requeiram a gestão de presenças e encontros de pessoas em ambientes internos. Composta por quatro componentes, a arquitetura proposta utiliza *beacons* Bluetooth e dispositivos Android para representar pessoas e espaços físicos. Com a integração dos componentes, a arquitetura pode detectar presença de pessoas em tempo real, calcular o tempo total de permanência, e verificar encontros de pessoas. A efetividade da solução proposta foi demonstrada através da obtenção de resultados promissores em uma avaliação experimental que comparou o tempo real de permanência de uma pessoa em um espaço físico com o tempo computado pela arquitetura.

\*Laboratório de Sistemas Distribuídos Inteligentes

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

## Palavras-chave

Cidades Inteligentes, Internet das Coisas, *Rendezvous*

## ABSTRACT

With technological advances and the popularization of the Internet of Things (IoT), more concepts of Smart Cities have been adopted in large urban centers, in particular, in smart buildings, that use sensors to better control their facilities. One of the areas of smart buildings is the presence and meeting management that manages the displacement and location of occupants in the building. The detection of people in indoor environments is becoming increasingly useful, especially in times of pandemic, when it is important to identify which people were close to each other in the same place and for how long. Considering this scenario, this study presents a distributed software architecture that uses four components that provide services for storing and consulting data, meeting notifications, and identifying the devices involved. In a flexible way, beacons and Android devices can be used to represent both people and physical spaces. In addition, the proposed architecture enables to determine attendances in real time, calculate the total time of stay, and check meetings of people. The effectiveness of the proposed solution was demonstrated through an experimental evaluation simulating its use.

## Keywords

Smart Cities, Internet of Things, *Rendezvous*

## CCS Concepts

•Human-centered computing → Ubiquitous and mobile computing; •Computer systems organization → *Distributed architectures*;

## 1. INTRODUÇÃO

O êxodo rural, isto é, o deslocamento de pessoas das áreas rurais para áreas urbanas, tem se tornado cada vez mais comum, e desde 2009 mais da metade da população mundial vive em cidades [1]. Este fato trás consigo uma série de desafios para os grandes centros urbanos. Torna-se cada vez mais comum problemas inerentes a estes centros, tais como a superlotação do sistema de transporte público, engarrafamentos, e violência.

De forma a minimizar esses problemas típicos de grandes centros urbanos, surge o conceito de Cidades Inteligentes. Uma cidade inteligente é aquela que utiliza a tecnologia para transformar sua infraestrutura básica, objetivando a otimização de seus recursos [2]. As cidades inteligentes conseguem reduzir os custos necessários para sua manutenção e aumentar sua eficiência, gerando uma melhor qualidade de vida aos cidadãos.

O objetivo de uma cidade inteligente pode ser alcançado através do paradigma de Internet das Coisas (IoT) [3]. A IoT é um paradigma que visa conectar objetos do dia a dia à Internet, de maneira a formar um sistema em que os mundos real e virtual se encontram e continuamente interagem [4]. Existem várias soluções para cidades inteligentes que fazem o uso da IoT. Alguns exemplos são: o monitoramento da estrutura de prédios históricos, detecção de barulho perto de áreas críticas como escolas e hospitais, controle inteligente do funcionamento de semáforos e lâmpadas de iluminação pública [5].

Em cidades inteligentes, o conceito de Prédios Inteligentes (do inglês, *Smart Buildings*) está relacionado a prédios instrumentados com Tecnologias de Informação e Comunicação (TICs) baseadas em elementos de IoT com o objetivo de tornar seu uso mais eficiente e produtivo sob diversas perspectivas, como uma gestão mais eficiente de energia, impactos ambientais e de auxílio às pessoas e suas interações sociais, sejam de trabalho ou lazer [6]. Para isso, os prédios são equipados com diferentes tipos de sensores que coletam dados. Os dados coletados, no contexto de um prédio inteligente, podem ser utilizados para extrair informações relevantes que auxiliem o gerenciamento do prédio facilitando o processo de tomada de decisões. Um aspecto importante na percepção do espaço físico, em prédios inteligentes, consiste na movimentação das pessoas e suas interações.

A gestão de presença e encontros de pessoas é uma das possíveis áreas de atuação da IoT em prédios inteligentes. Essa gestão tem o objetivo de identificar locais que estão ocupados em um prédio e possíveis interações entre os seus ocupantes. Apesar de funcionar bem espaços abertos, o GPS não é preciso o bastante para determinar a localização exata de pessoas em ambientes fechados/internos [7]. Sendo necessário, portanto, o uso de novas tecnologias de localização para detecção de pessoas neste tipo de ambiente. Uma tecnologia que se destaca nesse contexto é a utilização de pontos de referência (do inglês, *landmarks*) *Bluetooth*, de forma que, cada ponto é associado a um ambiente do prédio, conseguindo assim uma localização mais precisa dentro do prédio.

Um exemplo da aplicabilidade de soluções nesse sentido seria em um ambiente de trabalho, em que, após um funcionário ser diagnosticado com COVID-19, seria possível identificar quais outros funcionários estiveram em contato com ele, quais locais possivelmente estariam contaminados, e a probabilidade de algum outro funcionário ter

sido contaminado, baseado no tempo de permanência dos dois funcionários em um mesmo local. Um outro exemplo, seria em um laboratório de pesquisa onde o professor orientador deseja saber quanto tempo os alunos permanecem no laboratório. O professor poderia verificar o tempo de permanência dos alunos no laboratório e se os estudantes/pesquisadores de um grupos de pesquisa se reuniram durante a semana.

Este artigo é uma versão estendida do trabalho publicado em [8], que apresenta uma arquitetura distribuída IoT para gestão de presença e encontros em ambientes internos. Através de quatro componentes de software e *beacons Bluetooth*, a arquitetura distribuída serve de base para o desenvolvimento de aplicações voltadas para o gerenciamento de presença em prédios inteligentes, sendo possível determinar o compartimento do prédio em que determinada pessoa está ou esteve, o seu tempo no permanência no local e se duas ou mais pessoas estiveram no mesmo ambiente ao mesmo tempo.

O restante do artigo está organizado da seguinte forma. A Seção 2 introduz o *middleware* M-Hub/CDDL e o conceito de *beacons Bluetooth*, necessários para o entendimento da solução proposta. A Seção 3 detalha a solução proposta. A Seção 5 discute os trabalhos relacionados na área de gestão de presença em prédios inteligentes. A Seção 4 apresenta uma avaliação que demonstra a funcionalidade da solução. Por fim, a Seção 6 conclui o trabalho e apresenta as perspectivas de trabalhos futuros.

## 2. FUNDAMENTAÇÃO

Esta seção apresenta os conceitos fundamentais para o entendimento da solução proposta. Será apresentado o conceito de Internet das Coisas, o *middleware* M-Hub/CDDL, utilizado como base para o desenvolvimento da solução, assim como os *beacons Bluetooth*.

### 2.1 Internet das Coisas

A Internet das Coisas é um paradigma de comunicação, em que objetos do dia a dia são equipados com tecnologias que permitem que se comuniquem com seus usuários e com outros objetos, se tornando parte integral da Internet [9]. O seu conceito visa interconectar objetos do mundo virtual ao mundo real, tornando a Internet cada vez mais imersiva e pervasiva.

No contexto da IoT, as "coisas" (do inglês, *things*) são denominados "objetos inteligentes" (*smart objects*) [10]. O *Cluster of European Research Projects on IoT* [11] define "coisas" como participantes ativos da rede capazes de interagir e comunicar-se com outros elementos da rede e com o ambiente. Esses objetos inteligentes podem ser computadores, celulares, mas também objetos do cotidiano, como roupas, móveis, pontos de referência, monumentos, e obras de arte [12].

Em algumas aplicações, os requisitos de mobilidade e baixa latência são muito importantes [13]. Essa classe de aplicações que requerem uma alta mobilidade dos dispositivos IoT é chamada Internet das Coisas Móveis (IoMT) [14]. Na IoMT, tanto objetos quanto *gateways* IoT podem se mover ou serem movidos, permanecendo controláveis e acessíveis através da Internet [15]. A IoMT faz parte de diversos domínios de aplicação, tais como cidades inteligentes, logística, segurança pública, saúde e bem estar [16].

A principal característica da IoT é seu impacto na vida

cotidiana de usuários [9]. Domótica, transporte inteligente, sensores urbanos, são apenas alguns exemplos de possíveis cenários de aplicação da IoT no dia a dia. Já nos negócios, a IoT se faz presente com a automação industrial, transformações na logística e soluções na agricultura.

Diversas cidades ao redor do mundo já possuem aplicações de cidades inteligentes baseadas em IoT. Singapura, considerada uma das cidades mais inteligentes do mundo, criou em 2014 o programa *Smart Nation*, que envolveu a instalação de diversos sensores pela cidade, que medem desde o nível de limpeza de uma determinada área até o quão lotado está um evento. No Brasil, temos como exemplo a cidade do Rio de Janeiro, que tem como sua principal iniciativa a criação do Centro de Operações Rio (COR), que monitora mais de 15.000 sensores que proveem informações para gestão de crise, risco e incidentes em toda a cidade [17].

## 2.2 O middleware M-Hub/CDDL

Desenvolvido em parceria entre o Laboratório de Sistemas Distribuídos Inteligentes (LSDi) da Universidade Federal do Maranhão (UFMA) e o *Laboratory for Advanced Collaboration* (LAC) da Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio), o M-Hub/CDDL é um *middleware* IoT que possui recursos para a aquisição, processamento e distribuição de dados com qualidade de contexto entre dispositivos [18]. O *middleware* é composto por dois componentes de software: um *gateway* móvel, o *Mobile Hub* (M-Hub) e uma camada de distribuição de dados, o *Context Data Distribution Layer* (CDDL).

O componente M-Hub atua em *smartphones Android* e funciona como um *gateway* móvel para a aquisição de dados. O *middleware* é capaz de descobrir, registrar e coletar dados provenientes de sensores, como também enviar comandos de atuação aos mesmos. Estes dados podem ser obtidos através dos sensores internos do dispositivo (por exemplo, acelerômetro, giroscópio, GPS), do *Bluetooth* clássico e do *Bluetooth 4.0* (*Bluetooth Low Energy* - BLE). A comunicação entre o M-Hub e sensores é feita por meio da abstração de diferentes tecnologias de comunicação de curto alcance em uma única interface chamada de *Short-range Sensing, Presence and Actuation* (S2PA) [15]. Ao utilizar o M-Hub, o *smartphone* é transformado em um *gateway* móvel IoT/IoMT, capaz de coletar dados de sensores, dispositivos vestíveis, veículos, e ainda, enviar comandos de atuação.

Versões mais recentes do M-Hub contam com melhorias na detecção de objetos inteligentes. Por exemplo, em [19], é proposto o *Neighborhood-aware Mobile Hub* (NAM-Hub), uma versão do M-Hub com mecanismo para eleição de líderes. De forma, em um ambiente com vários objetos inteligentes, os diversos NAM-Hubs entram em acordo sobre quais objetos cada um deles deve monitorar, evitando o desperdício de recursos ao ocorrer o monitoramento do mesmo objeto por mais de um M-Hub. Utilizando informações de contexto do dispositivo móvel, é possível escolher o NAM-Hub mais adequado para cada objeto.

Já o CDDL é responsável pela distribuição dos dados com suporte de qualidade de contexto [20]. Atuando em clientes móveis *Android*, *desktops* e *web*, o CDDL oferece um meio de comunicação baseado em tópicos que media a comunicação entre aplicações consumidoras e serviços de objetos inteligentes detectados pelo M-Hub. O CDDL utiliza inter-

nação o *Message Queuing Telemetry Transport* (MQTT)<sup>1</sup> como protocolo de comunicação para transferência de dados, seja em conexões locais ou remotas [18]. Cada instância do CDDL possui internamente uma implementação leve em Java de um *broker* MQTT, chamado de *Microbroker*.

Através do conjunto de interfaces oferecidas pelo M-Hub/CDDL é possível desenvolver aplicações voltadas para IoT/IoMT que são capazes de detectar e se comunicar com objetos inteligentes embarcados com sensores e atuadores. Portanto, o M-Hub/CDDL é utilizado tanto na detecção e aquisição de dados de objetos inteligentes quanto na distribuição desses dados para outras instâncias do CDDL.

A Figura 1 ilustra os domínios de aplicação do M-Hub/CDDL. As setas representam o fluxo de dados, sendo possível a comunicação entre os sensores e o M-Hub, e entre diferentes instâncias do CDDL, móveis ou na nuvem. No domínio A, o *middleware* é utilizado para a aquisição de dados oriundos de objetos inteligentes, repartir dados entre instâncias móveis do CDDL, e para distribuir dados para instâncias do CDDL que trabalham na nuvem. No domínio B, aplicações específicas, através do CDDL, consomem os dados coletados.

## 2.3 Beacons BLE

O BLE é uma tecnologia incorporada a especificação *Bluetooth* na versão 4.0. Essa tecnologia evoluiu para transmitir uma baixa taxa de dados, mas com uma grande eficiência energética, adequada para aplicações IoT/IoMT com restrições de energia [21]. O BLE é considerado um rádio “otimizado por hardware”, o que significa que sua principal diferença em relação ao *Bluetooth* clássico reside no transceptor de rádio, processamento de sinal digital de banda base e formato de pacote de dados [22].

Os *beacons* BLE são dispositivos compostos por um *chipset Bluetooth*, uma bateria e uma antena que implementam o BLE. O funcionamento deste dispositivo se dá pela emissão regular de sinais de rádio, que são captados por aparelhos que estão com a interface *Bluetooth* ligada dentro de um raio de alcance do sinal. No entanto, os aparelhos não realizam tentativas de conexão com os *beacons*, como ocorre no *Bluetooth* clássico. Ao contrário, eles detectam sua presença e recebem os pacotes de dados transmitidos.

O raio de alcance de sinais do BLE é normalmente menor que o *Bluetooth* clássico, bem como menos dados são transmitidos, pois na maior parte do tempo o BLE permanece no modo *sleep*, saindo dele apenas para fazer rápidas trocas de pacotes. Devido a isso, o BLE traz benefícios como o baixo consumo de energia, demorando de meses a anos para terem suas baterias esgotadas [23], além de serem dispositivos relativamente baratos [24].

## 3. SOLUÇÃO PROPOSTA

Este trabalho apresenta uma arquitetura de software distribuída que serve de base para o desenvolvimento de aplicações voltadas para a gestão de pessoas em prédios inteligentes. A arquitetura provê serviços para o registro e controle do tempo de permanência de pessoas em espaços físicos.

A arquitetura foi desenvolvida visando os seguintes requisitos funcionais:

1. Fornecer consultas de localização em tempo real;

<sup>1</sup><https://mqtt.org>

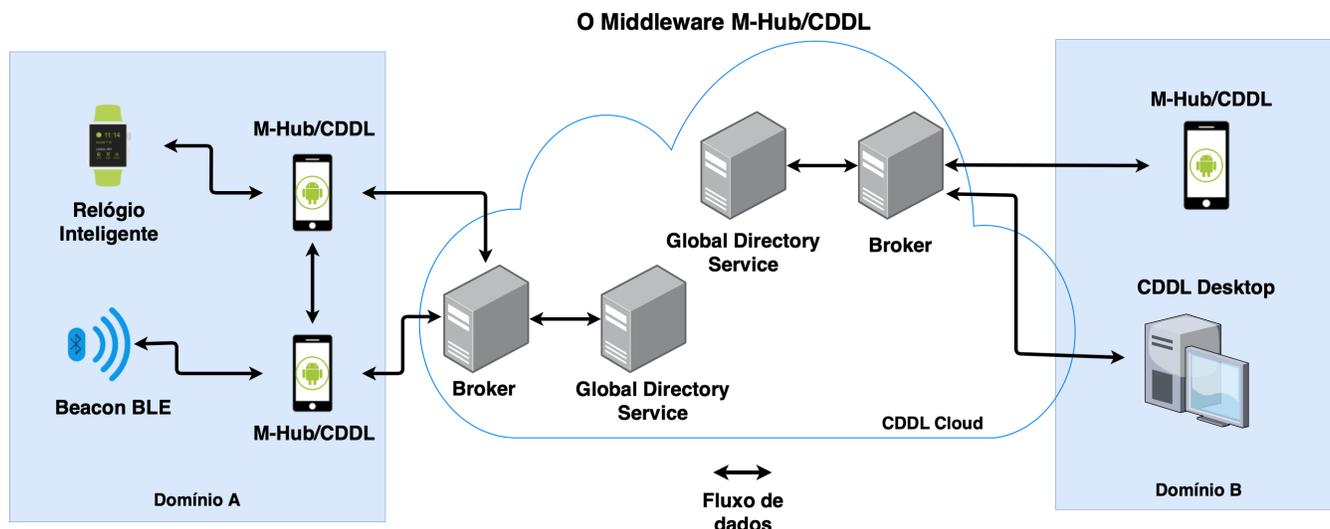


Figura 1: Domínios de Atuação do M-Hub/CDDL.

2. Fornecer consultas de localização em dados históricos, tanto individuais quanto para um grupo de pessoas;
3. Possibilitar a identificação nominal dos usuários e espaços físicos envolvidos no sistema.

Além disso, alguns requisitos não funcionais foram elicitados:

1. Escalabilidade, a qual define que o sistema pode ser usado por um grande número de usuários sem que haja perda de desempenho;
2. Flexibilidade, o sistema deve ser fácil de ser adaptado aos diferentes contextos de aplicação. Por exemplo, ele deve poder ser utilizado no contexto de um laboratório de pesquisa, mas também em um hospital.

### 3.1 Arquitetura Geral

A Figura 2 exibe uma visão geral da arquitetura. No lado esquerdo temos os *beacons* BLE, que correspondem aos espaços físicos reais, e o Serviço Android que, através do *middleware* M-Hub/CDDL, detecta os *beacons* e notifica o HORYS, que armazena essas informações. No lado direito estão as aplicações consumidoras, que se comunicam com o Serviço Atendente para obter informações úteis para a gestão de presença e encontros. O Serviço Atendente se comunica com os serviços HORYS e Semântico, a fim de montar a informação a ser entregue.

Para um melhor entendimento da arquitetura, faz-se necessário conhecer o conceito de *rendezvous*. O *rendezvous* simples, ou simplesmente *rendezvous*, é um evento que representa a detecção de um objeto inteligente por um dispositivo móvel. Essa detecção indica que o dispositivo está próximo a um *beacon*.

Após detectar o *beacon*, o dispositivo envia ao HORYS um conjunto de dados contendo os identificadores da aplicação (*appID*), do dispositivo móvel (*mhubID*) e do *beacon* encontrado (*thingID*). Além disso, é acrescentado a esses dados o atributo *timestamp*, o qual representa o instante de tempo em que ocorreu o encontro, e também o atributo *signal*, que

representa a intensidade do sinal recebido, conforme mostra a Figura 3.

Uma sucessão de *rendezvous* pode ser resumida para gerar um *rendezvous* composto, que representa o tempo total em que as duas entidades (ou seja, o M-Hub e o *beacon*) permaneceram próximas. O *rendezvous* composto se difere de um *rendezvous* simples por possuir os campos *found* e *lost*, os quais representam os instantes de tempo em que um M-Hub e um *beacon* estiveram próximos.

#### 3.1.1 Serviço Android

O primeiro componente da arquitetura a ser apresentado é o Serviço Android. Sua função é notificar *rendezvous* ao Serviço HORYS. Este é um serviço Android<sup>2</sup> do tipo *Background Service* que utiliza o *middleware* M-Hub/CDDL para detectar *beacons* BLE e notificar o serviço com os dados do encontro. Esta detecção ocorre pois os *beacons* enviam pacotes de dados em *broadcast* para os dispositivos que se encontram dentro do seu raio de alcance. A partir daí, o *middleware* M-Hub/CDDL, através da interface *Bluetooth*, consegue receber esses pacotes e notificar ao HORYS, via CDDL, que aconteceu um *rendezvous*. O serviço funciona em *background* e durante longos períodos de tempo, para que nenhum *rendezvous* seja perdido.

Para evitar a notificação de objetos inteligentes não relacionados à arquitetura, em sua inicialização, o Serviço Android requisita ao Serviço Semântico uma lista dos objetos cadastrados, e apenas estes são reconhecidos e notificados.

Para evitar o envio de dois ou mais *rendezvous* com *beacons* próximos, indicando a permanência em dois espaços físicos ao mesmo tempo, o serviço utiliza uma janela de seis detecções, em que o dispositivo armazena localmente seis *rendezvous* e envia ao servidor somente o de sinal mais forte.

#### 3.1.2 HORYS

O HORYS (acrônimo para *Hub-Object Rendezvous Registry Service*) [16] é o serviço responsável por armazenar

<sup>2</sup><https://developer.android.com/guide/components/services>

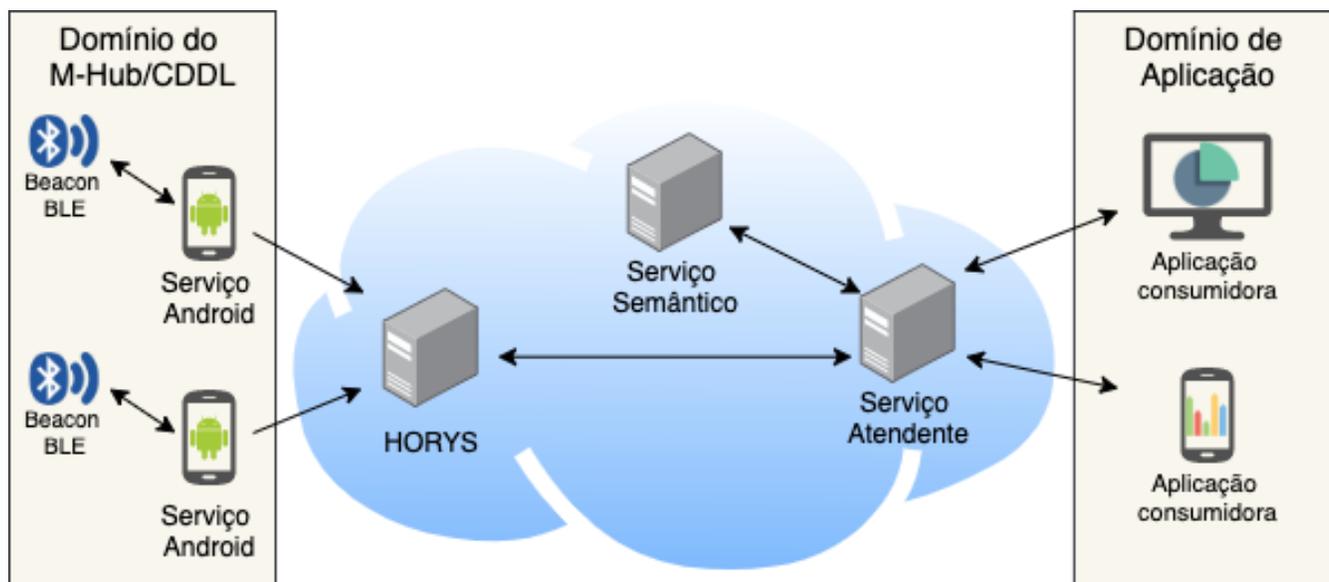


Figura 2: Visão geral dos componentes.

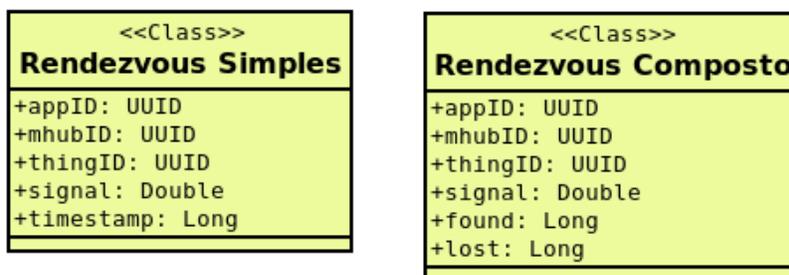


Figura 3: Modelo de *rendezvous* simples e composto.

e fornecer consultas aos *rendezvous*. O HORYS roda uma instância do CDDL, na versão desktop, através da qual recebe os *rendezvous* enviados pelo Serviço Android. Ao ser notificado de um *rendezvous*, o serviço o armazena em seu banco de dados e inicia o processo de sumarização. Para a comunicação com os outros serviços da arquitetura, o HORYS dispõe de *web-services* com métodos de consultas aos *rendezvous* e demais entidades envolvidas.

A sumarização de um *rendezvous* ocorre da seguinte maneira: um primeiro *rendezvous* simples é recebido pelo serviço, a partir daí, cria-se um *rendezvous* composto com os campos *found* e *lost* possuindo o mesmo *timestamp* do *rendezvous* recebido. Os próximos *rendezvous* irão atualizar o campo *lost* do *rendezvous* composto com o novo *timestamp*. Neste processo, utiliza-se um valor limite, em segundos, chamado de frequência de funcionamento, que define se um novo *rendezvous* pertence ou não a sumarização atual. Caso um novo *rendezvous* seja recebido, com uma diferença de tempo, em relação ao *rendezvous* anterior, maior que a frequência, então a sumarização é encerrada e o novo *rendezvous* se torna o primeiro de uma nova sucessão.

A partir dos *rendezvous* simples e compostos salvos em sua base de dados e entidades envolvidas, o serviço consegue fornecer diversos métodos de busca a esses dados. É

possível, por exemplo, passar o identificador do objeto inteligente (*thingID*) e verificar quais dispositivos (ou seja, M-Hubs) estão atualmente próximos. Através ainda de um *timestamp* arbitrário e um valor delta, que representa o intervalo de busca, é possível verificar se determinados objetos e M-Hubs estavam em contato naquela data e horário específicos. A Figura 4 lista os principais métodos do HORYS.

Nota-se que o HORYS não tem ciência do que representa os UUIDs de um encontro. Sua função é apenas registrar *rendezvous* simples e compostos. Dessa maneira, o serviço se torna flexível e pode ser utilizado em diversos tipos de soluções, deixando a critério de outro componente a identificação nominal dos encontros, o Serviço Semântico.

### 3.1.3 Serviço Semântico

Uma vez que os encontros estão armazenados no HORYS, existe a necessidade de se identificar as entidades envolvidas nos encontros. Este componente, denominado Serviço Semântico, tem como função mapear os relacionamentos entre pessoas, espaços físicos, M-Hubs e objetos inteligentes. Através de uma base de dados relacional, este serviço consegue cadastrar e identificar os UUIDs armazenados no componente anterior. Sua Interface de Programação de Aplicações (API) é voltada para o cadastro e a identificação das

<<Interface>> <b>HORYS</b>
<pre> +getConnectedMhubs(thingID) +getConnectedThings(mhubID) +getDurationThings(thingID) +getDurationMHub(mhubID) +getConnectedThings(mhubID,timestamp,delta) +getConnectedMhubs(thingID,timestamp,delta) +getLocationMhubs(timestamp,delta) +getLocationThings(timestamp,delta) +getDurationThings(thingID,timestamp,delta) +getDurationMHub(mhubID,timestamp,delta) </pre>

Figura 4: Principais métodos do HORYS.

entidades envolvidas nos encontros.

O Serviço Semântico parte do princípio de que espaços físicos possuem subdivisões, oferecendo assim uma estrutura em formato de árvore para armazenar esses dados. Por exemplo, uma casa pode ter como subdivisões uma sala, dois quarto, e uma cozinha. M-Hubs e objetos inteligentes podem pertencer tanto a usuários quanto a espaços físicos. Além disso, o Serviço Semântico pode atribuir papéis aos usuários, para que estes possam ser melhor identificados de acordo com sua função. Dessa forma, os usuários no contexto de um grupo de pesquisa podem ser associado aos papéis: aluno, professor ou pesquisador.

Para gerir presenças e encontros em espaços físicos, convencionou-se que pessoas portam M-Hubs e espaços físicos possuem objetos inteligentes (ou seja, *beacons*). Atuando flexivelmente, o Serviço Semântico pode inverter a convenção: pessoas portam objetos, e espaços físicos possuem M-Hubs. Dessa forma, a configuração é deixada a critério da lógica de negócio utilizada. A Figura 5 lista os principais métodos do Serviço Semântico.

<<Interface>> <b>Serviço Semântico</b>
<pre> +getAllMhubs(): &lt;Iterable&lt;Device&gt; +getMhubByUUID(uuid:String): Device +createMhub(mhub:Device): Device +getAllPhysicalSpaces(): Iterable&lt;PhysicalSpace&gt; +getPhysicalSpaceById(id:Long): PhysicalSpace +getPhysicalSpaceDescendantThings(id:Long): Iterable&lt;Device&gt; +createPhysicalSpace(physicalSpace:PhysicalSpace): PhysicalSpace +getAllThings(): Iterable&lt;Device&gt; +getThingByUUID(uuid:String): Device +createThing(thing:Device): Device +getAllPersons(): Iterable&lt;Person&gt; +getPersonByEmailOrId(emailOrId:String): Person </pre>

Figura 5: Principais métodos do Serviço Semântico.

### 3.1.4 Serviço Atendente

Apenas os componentes já apresentados não conseguem fornecer informações completas. O HORYS armazena os encontros sem identificar nominalmente as entidades envolvidas e o Serviço Semântico apenas tem ciência das entidades envolvidas e suas relações entre si. O terceiro componente é chamado de Serviço Atendente e sua função é consultar os dois serviços, combinar os dados, e então, entregar uma informação entendível às aplicações consumidoras.

Através das consultas combinadas são oferecidas operações mais complexas, por exemplo, dado várias pessoas, retorna todos os locais em que estiveram. Dado vários locais, retorna todas as pessoas que estiveram nos mesmos.

Através de uma API REST, este componente recebe requisições *Hypertext Transfer Protocol* (HTTP) e retorna os dados solicitados. Este serviço fornece as mesmas consultas básicas do HORYS. Porém, ele também verifica o serviço semântico para substituir os identificadores por nomes. Além disso, esse serviço também introduz o conceito de encontros entre pessoas. Calculando a interseção de encontros entre pessoas e locais, é possível determinar se pessoas estiveram no mesmo local, ao mesmo tempo, e quanto tempo permaneceram juntas. A partir disso, pode-se tirar conclusões a respeito das interações entre elas. Por exemplo, a direção de um hospital pode ter ciência de todos os profissionais que interagiram com um determinado paciente internado. A Figura 6 lista os principais métodos do Serviço Atendente.

<<Interface>> <b>Serviço Atendente</b>
<pre> +getPersonByPhysicalSpace(ids) +getPersonByPhysicalSpaceAndTime(ids,timestamp,delta) +getPersonRendezvous(ids,timestamp) +getPhysicalSpaceByPerson(ids) +getPhysicalSpaceByPersonAndTime(ids,timestamp,delta) +getPersonByPhysicalSpace(ids) +getPersonByPhysicalSpaceAndTime(ids,timestamp,delta) +getPersonRendezvous(ids,timestamp,delta) </pre>

Figura 6: Principais métodos do Serviço Atendente.

### 3.1.5 Integração dos Componentes

A Figura 7 representa, através de um diagrama de sequência, como funciona a comunicação entre os componentes da arquitetura para devolver uma informação útil às aplicações consumidoras.

No caso de uso representado pelo diagrama, o usuário seleciona um espaço físico e quer saber quais pessoas estão atualmente neste espaço. A aplicação consumidora requisita a informação ao Serviço Atendente passando como parâmetro o identificador (*id*) do espaço físico em questão. O Serviço Atendente, através de duas requisições ao Serviço Semântico, obtém o nome do espaço físico e o UUID do *beacon* correspondente a esse espaço físico. A partir daí, o Serviço Atendente requisita ao HORYS uma lista contendo a sumarização dos dispositivos que estão próximos àquele *beacon*. Para finalizar, o Serviço Atendente consulta o Serviço Semântico outra vez para, através dos UUIDs retornados, saber o nome desses usuários que estão no espaço físico e retornar essa informação para a aplicação.

## 4. AVALIAÇÃO EXPERIMENTAL

### 4.1 Objetivo e Metodologia

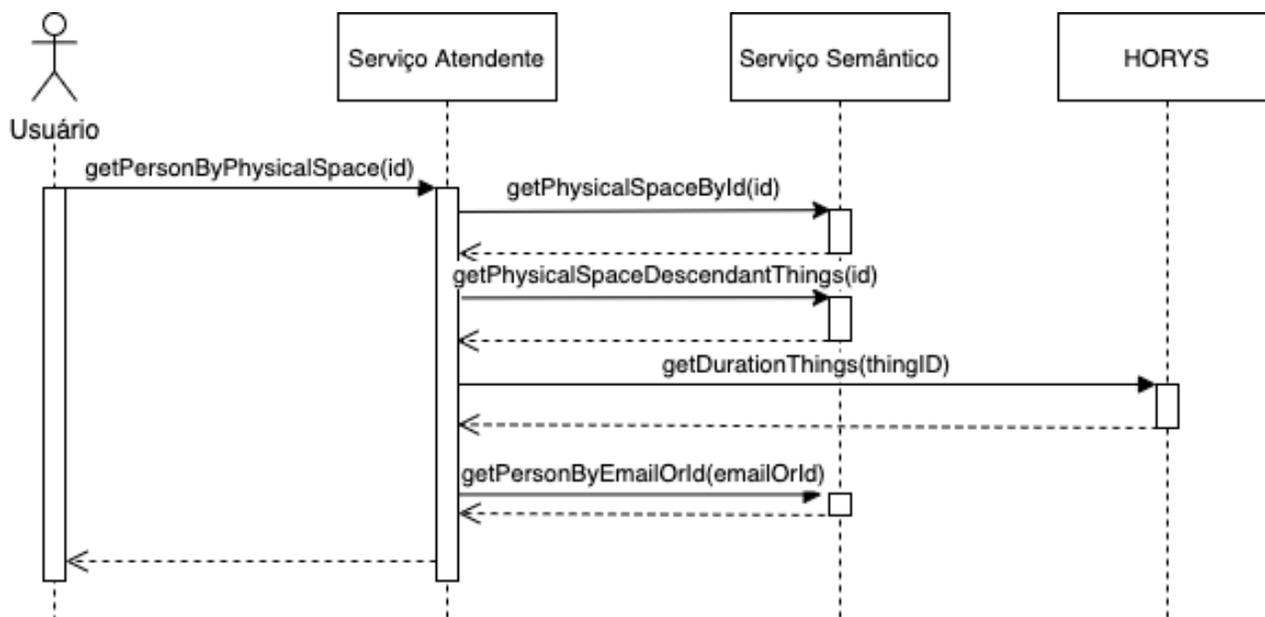


Figura 7: Diagrama de sequência da comunicação entre os componentes da arquitetura.

De forma a avaliar a solução proposta, elaborou-se um cenário de testes que compreende toda a arquitetura e possui ênfase no sistema de registro das presenças, ou seja, no HORYS. Para isso, foi desenvolvido uma aplicação Android que, utilizando o Serviço Android, notifica os encontros ao HORYS. Situada no contexto de um laboratório de pesquisa, a aplicação utiliza toda a arquitetura e dispõe de funcionalidades para visualização dos dados. A partir da mesma aplicação pode-se: enviar notificações de encontros ao HORYS, efetuar consultas no Serviço Semântico sobre pessoas e espaços físicos cadastrados, e consultar as informações de encontros através do Serviço Atendente. Portanto, essa aplicação foi utilizada como forma de demonstrar a aplicabilidade da solução em situações reais.

O experimento realizado com a solução teve como objetivo avaliar a consistência dos dados registrados, verificando se o tempo registrado pelo sistema era compatível ao tempo real de permanência da pessoa no ambiente. Além disso, o experimento também avaliou a disponibilidade do serviço Android, que tem como requisito não funcional a capacidade de operar durante horas de forma ininterrupta.

No centro de uma sala de um laboratório de pesquisa (ou seja, um ambiente interno) foi posicionado um *beacon* Bluetooth e uma pessoa utilizou este espaço físico normalmente. A pessoa utilizou o sistema para registrar o tempo passado (ou seja, a presença) no ambiente durante três dias. Além disso, ao entrar ou sair da sala, o horário era registrado manualmente pela pessoa. O objetivo deste cenário de testes foi verificar se o tempo de permanência real no ambiente, registrado manualmente, foi compatível ao tempo contabilizado pela aplicação desenvolvida com a arquitetura proposta.

Para efetuar os testes, foi utilizado um *smartphone* Android. Como *beacon*, foi utilizada uma pulseira *Mi Smart Band 4*. Ao longo da avaliação, o dispositivo Android permaneceu com a tela apagada, para demonstrar o funcionamento do serviço mesmo em períodos em que o dispositivo não está em uso. Antes da execução do teste, os relógios

do HORYS e o da pulseira utilizada pela pessoa foram sincronizados.

## 4.2 Resultados

A Figura 8 ilustra a aplicação Android desenvolvida. As duas primeiras telas consultam o Serviço Semântico, para a correta identificação de pessoas e locais. A última tela, através de consultas ao Serviço Atendente, exibe gráficos para a visualização dos dados. Este aplicativo também utiliza o Serviço Android para a notificação dos encontros ao servidor HORYS. Através desta aplicação, verificou-se que o sistema pode ser utilizado pelos mais variados tipos de problemas voltados para a gestão de presenças e encontros.

Os resultados dos três dias de experimentos foram coletados da aplicação e organizados, conforme mostra a Tabela 1. O valor registrado manualmente pela pessoa se encontra na coluna Valor Real. O Valor Computado corresponde ao tempo registrado pelo sistema, disponível em horas e em porcentagem. Subtraindo o Valor Computado do Valor Real, obtém-se o Valor Não Computado que é o tempo que o sistema deixou de registrar.

Os resultados obtidos mostram que os valores registrados pelo sistema são compatíveis com os tempos reais de permanência. O tempo total computado pelo sistema apresentou valores altos, obtendo-se aproximadamente 98.36% do valor real.

## 4.3 Discussão

Com o desenvolvimento da aplicação Android que utilizou todos os componentes, a arquitetura mostrou-se atuar de maneira transparente, se apresentando ao usuário como sendo apenas um único sistema. Os resultados obtidos com a avaliação experimental constatarem que o sistema pode ser utilizado para o registro de presença em tempo próximo ao real e determinação do tempo total de permanência de pessoas em espaços físicos. Além disso, também verificou-se que o Serviço Android consegue permanecer muitas horas

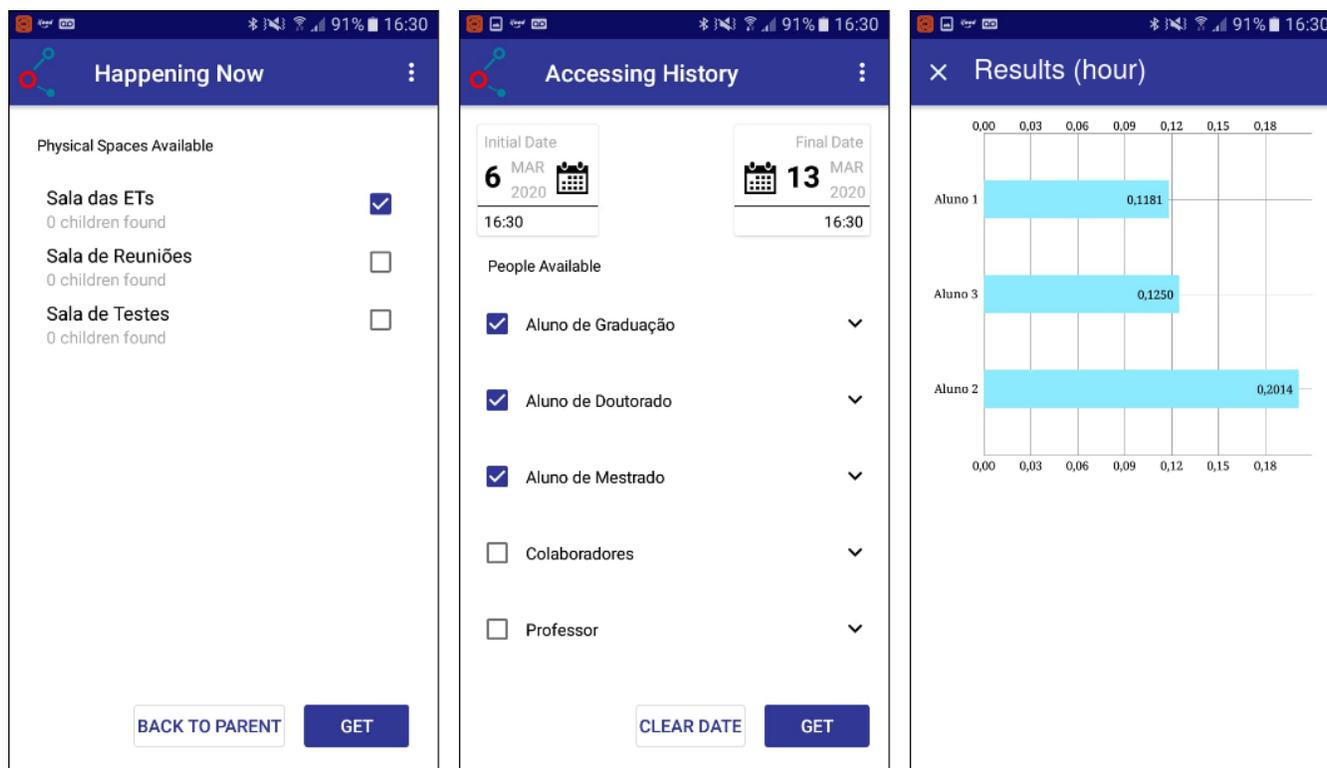


Figura 8: Exemplo de aplicação consumidora.

Tabela 1: Resultados da Avaliação - Tempo Real x Tempo Computado.

	Valor Real (hh:mm:ss)	Valor Computado (hh:mm:ss)	Valor Não Computado (hh:mm:ss)	Valor Computado (%)
14/06/2020	03:13:00	03:10:05	00:02:55	98,48
15/06/2020	03:20:00	03:16:51	00:03:09	98,42
16/06/2020	07:14:00	07:06:35	00:07:25	98,29
Total	13:47:00	13:33:31	00:13:29	98,36

seguidas em funcionamento, sendo possível seu uso em situações reais. Observa-se ainda que uma parcela de tempo que não é registrada pelo sistema. Um dos motivos para a ocorrência dessa perda é devido às otimizações de bateria do Sistema Operacional Android, as quais não permitem que um serviço fique em execução em *background* por muitas horas. Dessa forma, o sistema força o encerramento do serviço.

## 5. TRABALHOS RELACIONADOS

As pesquisas na área de localização de pessoas em ambientes internos estão se aprimorando e gerando soluções cada vez mais modernas e robustas [25]. Uma maneira de melhorar a precisão na localização de pessoas em ambientes fechados é por meio do uso de pontos de referência (do inglês, *landmarks*) [26, 27, 28]. Os *landmarks* são pontos de referência no espaço que possuem alguma informação espacial que auxilie na localização [29, 30]. Os pontos de referência são categorizados de acordo com o sensor e conhecimento prévio necessário para detectá-los. Existem duas categorias principais: pontos de referência sementes (do inglês, *seed landmarks*) e pontos de referência orgânicos (do inglês, *organic landmarks*) [31, 32]. Os pontos de referência sementes

são aqueles cujo sensor e localização física são conhecidos com antecedência, enquanto os pontos de referência orgânicos são aqueles sem conhecimento prévio do sensor e de sua localização.

Os pontos de referência utilizados na solução proposta por este trabalho estão na categoria de pontos de referência sementes, pois sua localização é conhecida e cadastrada no sistema correspondente às localizações no espaço físico. Além disso, uma categorização dos pontos mais específica utilizados neste trabalho são os pontos de referência *Bluetooth* (do inglês, *Bluetooth landmarks*) [33], uma vez que são reconhecidos ao serem detectados por uma interface *Bluetooth*.

Diferentes soluções que fazem uso de pontos de referência *Bluetooth* são propostas na literatura. Para gerenciar recursos de um prédio inteligente, Corna et al. [34] apresentaram uma solução que utiliza *iBeacons* para a detecção de pessoas. Os *beacons*, que transmitem pacotes de dados continuamente, são distribuídos em salas específicas. Paralelamente, uma aplicação captura o dado e notifica o servidor do sistema de gerenciamento. Executando algoritmos de classificação, o sistema é capaz de obter as localizações dos dispositivos com precisão e, a partir disso, é possível

automatizar as tomadas de decisões (por exemplo, desligar aparelhos de ventilação ao verificar que não existem pessoas presentes no ambiente).

A fim de determinar a localização em tempo real de funcionários de uma empresa, Syafrudin et al. [35] apresentaram uma arquitetura composta por *beacons* BLE, uma aplicação que atua como *gateway* e um servidor para armazenamento de dados. Os *beacons* são posicionados em salas e a aplicação no *smartphone* do funcionário notifica o servidor. Através de uma aplicação web, é possível consumir os dados do servidor e visualizar o movimento dos funcionários nos postos de trabalho.

Chang et al. [36] propõem uma solução para a detecção de presença através do uso de pulseiras *Bluetooth*. O usuário utiliza pulseiras de identificação, tornando desnecessário o uso de *smartphones*. O protocolo padrão do *beacon*, modificado pelos autores, permitiu a criação de uma rede, em formato de cadeia, em que foi possível a transmissão de dados entre *beacons*. No entanto, um computador com interface *Bluetooth* conectado à Internet foi requerido para atuar como *gateway* e enviar os dados a um servidor remoto. O sistema foi aplicado em um parque temático, onde os *beacons* foram distribuídos nas rotas de visitas.

O presente artigo, atuando no contexto de cidades inteligentes, especificamente em prédios inteligentes, apresenta uma solução para a gestão de presença e encontros em ambientes internos, sendo possível determinar as presenças em tempo real, calcular tempo total de permanência e verificar encontros entre pessoas. Como principal contribuição da solução proposta neste trabalho, podemos citar sua flexibilidade e capacidade de determinar intersecções de encontros. Corna et al. [34] e Syafrudin et al. [35] apresentaram configurações que *beacons* são posicionados em espaços físicos, e *smartphones* são utilizados como *gateway*. Chang et al. [36], utiliza pulseiras com *Bluetooth* para representar pessoas. A abordagem proposta por este artigo possibilita o uso das duas abordagens: pessoas portando M-Hubs e espaços físicos possuindo objetos inteligente, e vice-versa. Esse recurso aumenta a aplicabilidade da solução proposta, demonstrando sua flexibilidade, possibilitando o seu uso em diferentes contextos.

Nossa solução também destaca-se pela capacidade de calcular encontros entre pessoas, podendo informar que grupos de pessoas estiveram interagindo entre si. Esse recurso é muito útil em períodos de quarentena, como a ocasionada pelo novo coronavírus, quando há a necessidade de se monitorar pessoas infectadas pelo vírus, ou com suspeita de infecção, e identificar quais pessoas interagiram com essas pessoas infectadas. Além disso, em um hospital, por exemplo, é possível identificar os locais onde estiveram os pacientes e profissionais, bem como calcular o tempo utilizado em cada etapa de consulta, utilizando essa informação para encontrar gargalos nos atendimentos.

## 6. CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS

Este trabalho apresentou uma arquitetura distribuída que pode ser utilizada para o desenvolvimento de soluções de IoT. Através de quatro componentes independentes, é possível construir aplicações para a detecção de presença de pessoas em tempo real, determinação do tempo total de permanência em locais, e verificar a existência de interações

entre pessoas.

A flexibilidade dos componentes se revela como uma contribuição deste trabalho, em que é possível utilizar a arquitetura com as duas configurações possíveis, considerando *beacons* e *smartphones* como pessoas ou espaços físicos. Além disso, a partir da arquitetura proposta, é possível desenvolver outros tipos de soluções. Por exemplo, uma aplicação que monitora quanto tempo um aluno permanece nas dependências de seu laboratório, ou para descobrir se houve reuniões entre alunos.

Através do cenário de testes, demonstrou-se a efetividade do sistema. Os encontros foram notificados corretamente ao servidor, os dados armazenados foram consistentes, e os métodos de busca demonstraram-se ser amplos.

Como perspectivas futuras, pode-se citar um estudo direcionado para o uso de Processamento de Eventos Complexos (CEP). O HORYS, funcionando como um processador de eventos (ou seja, *rendezvous*) poderia se beneficiar do uso de regras CEP, tornando possível aumentar sua capacidade de recebimento de eventos, melhorando sua eficiência no processamento, e até disparando notificações customizadas aos usuários do sistema. Além disso, a partir desse trabalho, existe um potencial para a utilização de algoritmos de aprendizado de máquina para melhor determinar a posição de pessoas em relação aos ambientes internos.

## 7. AGRADECIMENTOS

Este trabalho foi financiado em parte pela CAPES (Código de financiamento 001 e 88887.200532/2018-00); Instituto Nacional de Ciência e Tecnologia (INCT) da Internet do Futuro para Cidades Inteligentes (CNPq 465446/2014-0, CAPES 88887.136422/2017-00, e FAPESP 14/50937-1 e 15/24485-9); Conselho Nacional de Desenvolvimento Científico e Tecnológico - CNPq (311608/2017-5, 420907/2016-5, 312324/2015-4, 154609/2019-6).

## 8. REFERÊNCIAS

- [1] U. Nations, "Urban and rural areas 2009," 2009.
- [2] J. E. P. de Farias, M. S. Alencar, Í. A. Lima, and R. T. Alencar, "Cidades inteligentes e comunicações," *Revista de Tecnologia da Informação e Comunicação*, vol. 1, no. 1, pp. 28–32, 2011.
- [3] A. Zanello, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of things for smart cities," *IEEE Internet of Things journal*, vol. 1, no. 1, pp. 22–32, 2014.
- [4] E. Borgia, "The internet of things vision: Key features, applications and open issues," *Computer Communications*, vol. 54, pp. 1–31, 2014.
- [5] F. Kon and E. F. Z. Santana, "Cidades inteligentes: Conceitos, plataformas e desafios," in *Congresso da Sociedade Brasileira de Computação*, pp. 2–49, 2016.
- [6] A. Daissaoui, A. Boulmakoul, L. Karim, and A. Lbath, "Iot and big data analytics for smart buildings: A survey," *Procedia Computer Science*, vol. 170, pp. 161–168, 2020. The 11th International Conference on Ambient Systems, Networks and Technologies (ANT) / The 3rd International Conference on Emerging Data and Industry 4.0 (EDI40) / Affiliated Workshops.
- [7] R. Faragher and R. Harle, "Location fingerprinting with bluetooth low energy beacons," *IEEE Journal on*

- Selected Areas in Communications*, vol. 33, no. 11, pp. 2418–2428, 2015.
- [8] T. Mendes, A. Cardoso, A. Silva, D. Carvalho, F. S. e Silva, A. Teles, M. Endler, and M. R. Júnior, “Uma arquitetura distribuída iot para gestão de presença e encontros de pessoas em ambientes internos,” in *Anais da VIII Escola Regional de Computação do Ceará, Maranhão e Piauí*, (Porto Alegre, RS, Brasil), pp. 236–243, SBC, 2020.
- [9] L. Atzori, A. Iera, and G. Morabito, “The internet of things: A survey,” *Computer networks*, vol. 54, no. 15, pp. 2787–2805, 2010.
- [10] D. Bandyopadhyay and J. Sen, “Internet of things: Applications and challenges in technology and standardization,” *Wireless personal communications*, vol. 58, no. 1, pp. 49–69, 2011.
- [11] H. Sundmaeker, P. Guillemin, P. Friess, and S. Woelfflé, “Vision and challenges for realising the internet of things,” *Cluster of European Research Projects on the Internet of Things, European Commission*, vol. 3, no. 3, pp. 34–36, 2010.
- [12] A. Botta, W. De Donato, V. Persico, and A. Pescapé, “Integration of cloud computing and internet of things: a survey,” *Future generation computer systems*, vol. 56, pp. 684–700, 2016.
- [13] W. Dargie and J. Wen, “A seamless handover for wsn using lms filter,” in *39th Annual IEEE Conference on Local Computer Networks*, pp. 442–445, IEEE, 2014.
- [14] K. Nahrstedt, H. Li, P. Nguyen, S. Chang, and L. Vu, “Internet of mobile things: Mobility-driven challenges, designs and implementations,” in *2016 IEEE First International Conference on Internet-of-Things Design and Implementation (IoTDI)*, pp. 25–36, IEEE, 2016.
- [15] L. E. Talavera, M. Endler, I. Vasconcelos, R. Vasconcelos, M. Cunha, and F. J. d. S. e. Silva, “The mobile hub concept: Enabling applications for the internet of mobile things,” in *2015 IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops)*, pp. 123–128, 2015.
- [16] M. Endler and F. S. e Silva, “Past, present and future of the contextnet iomt middleware,” *Open Journal of Internet Of Things (OJIOT)*, vol. 4, no. 1, pp. 7–23, 2018.
- [17] C. Schreiner, “International case studies of smart cities: Rio de janeiro, brazil,” *Inter-American Development Bank*, 2016.
- [18] B. D. T. P. Gomes, L. C. M. Muniz, F. J. Da Silva e Silva, D. V. Dos Santos, R. F. Lopes, L. R. Coutinho, F. O. Carvalho, and M. Endler, “A middleware with comprehensive quality of context support for the internet of things applications,” *Sensors*, vol. 17, no. 12, 2017.
- [19] M. Silva, A. Teles, R. Lopes, F. Silva, D. Viana, L. Coutinho, N. Gupta, and M. Endler, “Neighborhood-aware mobile hub: An edge gateway with leader election mechanism for internet of mobile things,” *Mobile Networks and Applications*, pp. 1–14, 2020.
- [20] A. Manzoor, H.-L. Truong, and S. Dustdar, “Quality of context: models and applications for context-aware systems in pervasive environments,” *The Knowledge Engineering Review*, vol. 29, no. 2, p. 154–170, 2014.
- [21] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, “Internet of things (iot): A vision, architectural elements, and future directions,” *Future generation computer systems*, vol. 29, no. 7, pp. 1645–1660, 2013.
- [22] M. Chen, S. Gonzalez, A. Vasilakos, H. Cao, and V. C. Leung, “Body area networks: A survey,” *Mobile networks and applications*, vol. 16, no. 2, pp. 171–193, 2011.
- [23] P. Kriz, F. Maly, and T. Kozel, “Improving indoor localization using bluetooth low energy beacons,” *Mobile Information Systems*, vol. 2016, 2016.
- [24] S. A. Cheraghi, V. Namboodiri, and L. Walker, “Guidebeacon: Beacon-based indoor wayfinding for the blind, visually impaired, and disoriented,” in *2017 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pp. 121–130, IEEE, 2017.
- [25] F. Gu, X. Hu, M. Ramezani, D. Acharya, K. Khoshelham, S. Valaee, and J. Shang, “Indoor localization improved by spatial context—a survey,” *ACM Computing Surveys (CSUR)*, vol. 52, no. 3, pp. 1–35, 2019.
- [26] T. Bailey and H. Durrant-Whyte, “Simultaneous localization and mapping (slam): Part ii,” *IEEE robotics & automation magazine*, vol. 13, no. 3, pp. 108–117, 2006.
- [27] J. Shang, F. Gu, X. Hu, and A. Kealy, “Apfloc: An infrastructure-free indoor localization method fusing smartphone inertial sensors, landmarks and map information,” *Sensors*, vol. 15, no. 10, pp. 27251–27272, 2015.
- [28] B. Zhou, Q. Li, Q. Mao, W. Tu, X. Zhang, and L. Chen, “Alimc: Activity landmark-based indoor mapping via crowdsourcing,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 5, pp. 2774–2785, 2015.
- [29] C. C. Presson and D. R. Montello, “Points of reference in spatial cognition: Stalking the elusive landmark,” *British Journal of Developmental Psychology*, 1988.
- [30] M. E. Sorrows and S. C. Hirtle, “The nature of landmarks for real and electronic spaces,” in *International conference on spatial information theory*, pp. 37–50, Springer, 1999.
- [31] H. Abdelnasser, R. Mohamed, A. Elgohary, M. F. Alzantot, H. Wang, S. Sen, R. R. Choudhury, and M. Youssef, “Semanticslam: Using environment landmarks for unsupervised indoor localization,” *IEEE Transactions on Mobile Computing*, vol. 15, no. 7, pp. 1770–1782, 2015.
- [32] H. Wang, S. Sen, A. Elgohary, M. Farid, M. Youssef, and R. R. Choudhury, “No need to war-drive: Unsupervised indoor localization,” in *Proceedings of the 10th international conference on Mobile systems, applications, and services*, pp. 197–210, 2012.
- [33] F. Gu, K. Khoshelham, J. Shang, and F. Yu, “Sensory landmarks for indoor localization,” in *2016 Fourth International Conference on Ubiquitous Positioning, Indoor Navigation and Location Based Services (UPINLBS)*, pp. 201–206, IEEE, 2016.
- [34] A. Corna, L. Fontana, A. Nacci, and D. Sciuto, “Occupancy detection via ibeacon on android devices

- for smart building management,” in *Design, Automation Test in Europe Conference Exhibition (DATE)*, pp. 629–632, 2015.
- [35] M. Syafrudin, K. Lee, G. Alfian, J. Lee, and J. Rhee, “Application of bluetooth low energy-based real-time location system for indoor environments,” in *Proceedings of the 2nd International Conference on Big Data and Internet of Things*, p. 167–171, ACM, 2018.
- [36] C.-m. Chang, S.-C. Li, and Y. Huang, “Building bluetooth beacon-based network for spatial-temporal data collection,” in *Proceedings of the International Conference on Communication and Information Systems*, pp. 91–95, ACM, 2016.