



UNIVERSIDADE FEDERAL DO MARANHÃO

Curso de Ciência da Computação

Pedro Arthur Freitas Dias

**Identificação de Padrões de Mobilidade Humana
Através da Fenotipagem Digital com Dados de
Localização**

São Luís

2025

Pedro Arthur Freitas Dias

Identificação de Padrões de Mobilidade Humana Através da Fenotipagem Digital com Dados de Localização

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

Freitas Dias, Pedro Arthur.

Identificação de Padrões de Mobilidade Humana Através
da Fenotipagem Digital com Dados de Localização / Pedro
Arthur Freitas Dias. - 2025.

44 f.

Orientador(a): Francisco José da Silva e Silva.

Monografia (Graduação) - Curso de Ciência da
Computação, Universidade Federal do Maranhão, São Luís-ma,
2025.

1. Cidades Inteligentes. 2. Fenotipagem Digital. 3.
Mobilidade. I. da Silva e Silva, Francisco José. II.
Título.

Pedro Arthur Freitas Dias

Identificação de Padrões de Mobilidade Humana Através da Fenotipagem Digital com Dados de Localização

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, 21 de agosto de 2025:

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

Prof. Dra. Inez Cavalcanti Dantas
Examinador

**Profa. Dr. Bruno Roberto Silva de
Moraes**
Examinador

São Luís
2025

Agradecimentos

À minha esposa, Anna Clara Vieira Pinto,

Uma das maiores pesquisadoras que já tive o privilégio de conhecer.

Gostaria de expressar minha gratidão ao professor, Dr. Francisco, pela orientação e paciência ao longo deste trabalho. Agradeço também a todos os meus colegas do LSDi, cuja colaboração, ajudas com a implementação foi fundamental para que eu conseguisse concluir este projeto.

"O Big Brother está observando você."

George Orwell., em *"1984"*

Resumo

A crescente evolução dos smartphones, equipados com sensores de localização e movimento, revolucionou a forma de capturar e analisar dados sobre o comportamento humano. A fenotipagem digital (FD) surge como uma abordagem que permite quantificar o fenótipo comportamental individual, registrando informações em tempo real a partir de dispositivos digitais pessoais. Isso possibilita a análise de hábitos, interações e, especialmente, padrões de mobilidade. A coleta de dados de GPS permite mapear deslocamentos, identificar rotinas e compreender a dinâmica espacial dos indivíduos, oferecendo *insights* sobre frequências de visita, trajetos recorrentes e mudanças de comportamento ao longo do tempo. Neste contexto, foi desenvolvido um middleware capaz de calcular as características de movimento a partir dos dados de localização, automatizando o processamento dessas informações para extrair features relevantes. Além disso, foi criada uma ferramenta para facilitar a coleta dos dados e a definição das labels, que são utilizadas e validar o middleware. O desenvolvimento dessas soluções visa proporcionar uma análise mais eficiente e precisa dos padrões de mobilidade humana, com potencial para aplicações em áreas como saúde, planejamento urbano e segurança.

Palavras-chave: Cidades Inteligentes, Fenotipagem digital, Mobilidade, Coleta de Dados, Localidade, *Middleware*.

Abstract

The rapid evolution of smartphones, equipped with location and motion sensors, has revolutionized the way human behavior data is captured and analyzed. Digital phenotyping (DP) emerges as an approach that allows quantifying the individual behavioral phenotype by recording real-time information through personal digital devices. This enables the analysis of habits, interactions, and especially mobility patterns. GPS data collection makes it possible to map displacements, identify routines, and understand individuals' spatial dynamics, providing insights into visit frequencies, recurring routes, and behavioral changes over time. In this context, a middleware was developed to calculate movement features based on location data, automating the processing of this information to extract relevant features. Additionally, a tool was created to facilitate data collection and label definition, which are used to train and validate the middleware. The development of these solutions aims to provide a more efficient and accurate analysis of human mobility patterns, with potential applications in areas such as health, urban planning, and security.

Keywords: Smart Cities, Digital phenotyping, Mobility, Data Collection, Location, Middleware.

Lista de ilustrações

Figura 1 – Arquitetura do Sistema de Coleta e Processamento de Dados	24
Figura 2 – Integração da Aplicação Feature Collector com o CDDL	26
Figura 3 – Interface da Aplicação Feature Coletor	32

Lista de tabelas

Tabela 1 – Desempenho dos modelos com base nas métricas MSE, MAE e Silhouette Score.	39
--	----

Lista de abreviaturas e siglas

CC	<i>Ciência da Computação</i>
UFMA	<i>Universidade Federal do Maranhão</i>

Sumário

1	INTRODUÇÃO	12
1.1	Justificativa	12
1.2	Objetivos	14
1.2.1	Objetivos específicos	14
2	FUNDAMENTAÇÃO TEÓRICA	15
2.1	Fenotipagem Digital	15
2.2	Computação Positiva	15
2.3	Mobilidade	16
2.4	Mineração de Padrões Frequentes	17
2.5	Eventos de Mobilidade Derivados de Dados de <i>Smartphones</i> e Sensores Vestíveis	17
2.5.0.1	Tipos de Eventos de Mobilidade	18
2.5.0.1.1	Eventos de Movimento:	18
2.5.0.1.2	Eventos de Permanência:	18
2.6	Técnicas Computacionais para a Computação de Características e Eventos Relacionados à Mobilidade	19
2.7	Métricas e Critérios de Avaliação dos Métodos Computacionais Empregados na Detecção de Eventos de Mobilidade	20
2.8	Modelo de Dados para Fenotipagem Digital	20
2.8.1	Estrutura Hierárquica de Dados	20
2.8.2	Observações Comportamentais	21
2.9	Tratamento de coleta de dados de GPS	21
2.10	Desenvolvimento de Aplicações Web com Streaming de Dados	21
2.10.1	Frameworks e Arquiteturas	21
2.10.2	Características e Vantagens	21
3	METODOLOGIA	23
3.1	Visão geral da Arquitetura	23
3.2	Desenvolvimento da Aplicação Android (Feature Collector)	24
3.2.0.1	Arquitetura da Aplicação	25
3.2.0.2	Integração com o CDDL	25
3.2.0.3	Gerenciamento de Usuários	27
3.2.0.4	Sistema de Coleta de Localização	29
3.2.0.5	Coleta de Dados	31
3.3	Sistema de Processamento de Dados (Feature Calculator)	32

3.3.1	Arquitetura	32
3.3.2	Tecnologias Empregadas no Desenvolvimento	33
3.4	Implementação do Processamento de Medições e Cálculo de	
	Features de Mobilidade	34
3.4.1	Job Spark com Padrão Strategy	34
3.4.2	Cálculo usando DBSCAN com ELKI	35
3.4.3	Cálculo do Tempo Gasto em Casa	36
3.4.4	Cálculo do Tempo Gasto em Locais Predefinidos	37
4	RESULTADOS	39
5	CONCLUSÃO	40
	REFERÊNCIAS	41

1 Introdução

Na atualidade, a presença dos *smartphones* na vida cotidiana expandiu-se consideravelmente, impulsionada pela diversidade de opções disponíveis no mercado e pela faixa de preços que os torna alcançáveis por diferentes grupos sociais. No Brasil, esse crescimento é evidente: em 2023, 72,5 milhões de domicílios utilizavam a internet (92,5%), um aumento de 1 ponto percentual em relação a 2022. Esse avanço foi ainda mais marcante em áreas urbanas, onde o uso subiu de 93,5% para 94,1%, e em zonas rurais, que passaram de 78,1% para 81,0%, refletindo uma maior penetração da tecnologia no cotidiano (MCCOOL et al., 2022).

Esse avanço dos dispositivos móveis reflete-se tanto em suas funcionalidades internas quanto em sua estrutura física. No campo dos softwares, observa-se a criação de ferramentas e plataformas cada vez mais sofisticadas, capazes de suprir uma ampla gama de necessidades dos usuários diretamente pelo aparelho. Já em termos de hardware, destacam-se os aprimoramentos em componentes como processadores mais potentes, telas táteis de alta responsividade e baterias com maior autonomia (MCCOOL et al., 2022).

A isso soma-se a incorporação de sensores variados, como os de localização, movimento e som, que caracterizam os modelos mais recentes. Ao mesmo tempo, o progresso tecnológico dos smartphones tem despertado o interesse de especialistas da área da saúde, que buscam formas de otimizar a avaliação e o acompanhamento de pacientes em contextos clínicos. Esse esforço surge, em grande parte, das restrições impostas por abordagens convencionais, como os relatos diretos dos pacientes sobre seus sintomas e vivências, que podem ser influenciados por questões emocionais, cognitivas ou sociais, gerando possíveis distorções nas informações coletadas (COHEN et al., 2023).

Nesse cenário, os dispositivos móveis têm se mostrado promissores na captura de dados de maneira passiva, oferecendo uma alternativa para mapear padrões comportamentais de forma mais precisa. Através de seus sensores, é viável registrar informações sobre deslocamentos, níveis de atividade física, qualidade do sono e interações sociais, entre outros aspectos, contribuindo para diagnósticos mais objetivos e para o acompanhamento de tratamentos com menor interferência dos vieses inerentes aos métodos tradicionais (JONGS et al., 2020).

1.1 Justificativa

Com o avanço tecnológico e a onipresença dos *smartphones*, a coleta de dados passivos tornou-se uma ferramenta poderosa para a compreensão do comportamento

humano, especialmente por meio da fenotipagem digital (F.D). Esse campo de pesquisa, que quantifica o fenótipo comportamental em nível individual a partir de dados de dispositivos digitais pessoais, tem se destacado por seu potencial de aplicação na área da saúde. Através do Monitoramento Passivo Móvel (MPM), é possível obter informações detalhadas sobre mobilidade, atividade física, sociabilidade e outros aspectos da vida cotidiana sem demandar interação ativa do usuário, o que torna essa abordagem prática, escalável e minimamente invasiva.

Nesse contexto, o cálculo de *features* (características) de mobilidade a partir de dados de localização coletados por *smartphones* emerge como uma contribuição significativa para a saúde. A mobilidade humana, enquanto componente essencial do fenótipo digital, reflete padrões de comportamento que podem estar diretamente associados ao estado de saúde física e mental de um indivíduo. Por exemplo, a análise de métricas como tempo gasto em casa, distância máxima percorrida, número de locais visitados ou variações na rotina de deslocamento pode revelar sinais precoces de condições como depressão, ansiedade, doenças neurodegenerativas ou até mesmo a adesão a tratamentos médicos. A nível individual, essas *features* permitem que profissionais de saúde monitorem pacientes de forma objetiva e contínua, identificando anomalias em padrões de mobilidade que possam indicar agravamento de sintomas ou necessidade de intervenção. A nível coletivo, o estudo dessas características pode auxiliar na modelagem de padrões epidemiológicos, na previsão de surtos de doenças infecciosas ou na alocação otimizada de recursos de saúde pública, como hospitais e campanhas de vacinação, conforme destacado por Emish (2023).

O desenvolvimento de uma ferramenta aberta e agnóstica para a coleta de dados de localização, como proposto neste trabalho, amplia ainda mais a relevância deste trabalho. Ao permitir a coleta de dados de forma flexível e independente de plataformas específicas, a ferramenta oferece uma solução acessível e adaptável para pesquisadores e profissionais da saúde. Além disso, a possibilidade de associar labels (etiquetas) aos locais visitados — como "casa", "trabalho", "hospital", etc — agrega um valor interpretativo crucial aos dados brutos. Essa funcionalidade facilita a catalogação e a contextualização das informações de mobilidade, tornando-as mais úteis para análises específicas. Por exemplo, saber que um paciente passou mais tempo em casa ou visitou um hospital com frequência pode fornecer pistas importantes sobre seu estado de saúde ou comportamento, sem depender exclusivamente de relatos subjetivos.

A natureza aberta da ferramenta também é um diferencial relevante, pois democratiza o acesso a esse tipo de tecnologia, permitindo que outros pesquisadores e desenvolvedores a utilizem, adaptem e aprimorem. Em um cenário onde a reprodutibilidade e a colaboração são pilares fundamentais da ciência, disponibilizar uma solução de código aberto contribui para o avanço coletivo do conhecimento na área de fenotipagem digital e saúde.

Portanto, este trabalho se justifica pela sua capacidade de integrar o cálculo de *features* de mobilidade ao campo da saúde, oferecendo uma abordagem inovadora para o monitoramento e a avaliação de pacientes, bem como para o estudo de padrões populacionais. A criação de uma ferramenta aberta e agnóstica para coleta de dados de localização, com suporte à rotulação de locais, não apenas potencializa a aplicabilidade prática dessas *features*, mas incentiva a pesquisa interdisciplinar e o desenvolvimento de soluções que podem contribuir para uma nova forma de entender e gerenciar a saúde na era digital.

1.2 Objetivos

O objetivo geral deste trabalho é desenvolver uma ferramenta aberta para a coleta de dados de localização por meio de *smartphones*, bem como escrever uma aplicação que realize o cálculo de *features* de mobilidade e a avaliação dos modelos gerados, contribuindo para a fenotipagem digital aplicada à área da saúde. A proposta visa aproveitar a ubiquidade desses dispositivos, como informações derivadas de sensores de GPS, para criar uma solução que facilite a análise de padrões de mobilidade humana. Além disso, a aplicação a ser desenvolvida busca não apenas extrair características quantitativas relevantes, como tempo gasto em determinados locais ou distância percorrida, mas também validar sua eficácia por meio de métricas, garantindo que os resultados possam ser utilizados de forma confiável.

1.2.1 Objetivos específicos

- Desenvolver uma arquitetura simples e robusta para coletar e armazenar dados de localização via *smartphones*. Ela usará sensores como GPS para capturar informações de mobilidade de forma contínua, resolvendo problemas comuns como falhas de conectividade, alto consumo de bateria e inconsistências de sinal. O sistema garantirá dados confiáveis e acessíveis para análises posteriores.
- Modelar uma aplicação para calcular *features* de mobilidade com uma estrutura modular que facilite adicionar novos métodos de cálculo para a mesma *feature*. Isso permitirá integrar variações de algoritmos, como diferentes formas de medir tempo em casa ou locais visitados, tornando o sistema flexível e adaptável às necessidades da fenotipagem digital.
- Desenvolver e avaliar abordagens para o cálculo incremental de padrões de mobilidade a partir de dados de localização coletados por *smartphones* em contextos definidos (por exemplo, rotinas semanais, deslocamentos diários).
- Desenvolver e avaliar abordagens para validação da eficácia e acurácia dos cálculos das características de mobilidade.

2 Fundamentação Teórica

2.1 Fenotipagem Digital

A presença constante e a versatilidade dos smartphones possibilitam a obtenção contínua de dados passivos sem demandar interação direta dos usuários. Isso permite que as pessoas mantenham suas rotinas diárias sem se preocupar com o processo de coleta, o que é essencial para construir perfis comportamentais baseados nessas informações. Esse processo é conhecido na literatura como fenotipagem digital, uma abordagem que utiliza dados gerados por dispositivos para caracterizar hábitos e padrões individuais (MOSHE *et al.*, 2021).

Nos últimos anos, esse campo tem se destacado por sua capacidade de oferecer soluções inovadoras, especialmente na área da saúde. Com essa técnica, profissionais podem monitorar pacientes de maneira mais precisa, utilizando dados objetivos para acompanhar os impactos de tratamentos ou identificar sinais de alterações clínicas. A fenotipagem digital pode ser entendida como a análise detalhada e em tempo real do comportamento humano, capturada no ambiente natural do indivíduo por meio de dispositivos digitais pessoais (TOROUS *et al.*, 2016).

Uma das formas de aplicar a fenotipagem digital é o chamado Monitoramento Passivo Móvel (MPM), que coleta informações diretamente dos smartphones sem exigir ações ativas do usuário, além de sua permissão inicial e configuração (JAGESAR *et al.*, 2021). Esse método aproveita diversos sensores dos aparelhos, como o GPS para rastrear deslocamentos, o acelerômetro para registrar movimentos, além de dados de comunicação, como ligações ou interações em redes sociais, que sugerem níveis de sociabilidade (JAGESAR *et al.*, 2021). Até mesmo o microfone pode ser usado para detectar períodos de conversa ou silêncio. Assim, a fenotipagem digital oferece um amplo conjunto de informações, permitindo aos profissionais de saúde uma compreensão mais profunda do estado e das rotinas de seus pacientes.

2.2 Computação Positiva

A Computação Positiva, conceito criado por Calvo e Peters (2014), estabelece um paradigma que direciona o desenvolvimento tecnológico explicitamente para "apoiar o bem-estar psicológico e o potencial humano" (CALVO; PETERS, 2014). Esta abordagem surge como resposta direta aos efeitos negativos do uso tecnológico intensivo, incluindo o estresse por notificações excessivas e violações de privacidade (CALVO; PETERS, 2014).

As soluções de Computação Positiva são desenvolvidas com propósito definido: promover a saúde mental e potencializar capacidades humanas, respeitando necessidades psicológicas individuais. Estas ferramentas também aumentam significativamente a eficiência dos trabalhadores do conhecimento em ambientes profissionais (CALVO; PETERS, 2014).

O avanço da computação ubíqua e tecnologias IOT impulsionou definitivamente a computação positiva inteligente (LEE et al., 2019). Estas tecnologias oferecem benefícios concretos: detectam alterações comportamentais indicativas de problemas de bem-estar, fornecem intervenções terapêuticas no momento adequado e permitem monitoramento sistemático para avaliar a eficácia das intervenções implementadas (??).

A Computação Positiva representa, portanto, não apenas uma tendência tecnológica, mas uma transformação na forma como projetamos e utilizamos ferramentas digitais para promover ativamente o bem-estar humano.

2.3 Mobilidade

Os *smartphones* modernos, equipados com GPS, acelerômetro, giroscópio e diversos outros sensores, constituem ferramentas poderosas para a coleta de dados detalhados sobre localização, movimento e ambiente do usuário (EMISH et al., 2023). Esta capacidade tecnológica permite a derivação de informações precisas sobre padrões de mobilidade, atividade física, ciclos de sono e comportamentos sociais (EMISH et al., 2023).

Os dados de mobilidade destacam-se como componentes essenciais na construção do fenótipo individual. A partir dos registros de mobilidade, extraem-se métricas objetivas e significativas que incluem o tempo de permanência na residência, distâncias máximas percorridas diariamente e a quantificação de locais visitados, fornecendo parâmetros concretos para avaliações comportamentais (EMISH et al., 2023).

A análise da mobilidade humana opera em duas dimensões complementares: individual e coletiva. Na esfera individual, possibilita a identificação precisa de padrões regulares e a detecção imediata de anomalias comportamentais. Na dimensão coletiva, esses dados revelam padrões epidemiológicos e fundamentam estratégias otimizadas para alocação de recursos em saúde pública e infraestrutura urbana (EMISH et al., 2023).

Portanto, a aplicação bidimensional destas informações—tanto no âmbito individual quanto coletivo—estabelece um novo paradigma para intervenções de saúde baseadas em evidências comportamentais objetivas.

2.4 Mineração de Padrões Frequentes

A identificação de padrões em conjuntos de dados depende de ferramentas computacionais específicas que formam a base da mineração de padrões frequentes. Quando uma estrutura aparece com frequência acima de um limiar de suporte estabelecido—seja em bancos de dados relacionais, não-relacionais, arquivos textuais ou fluxos de dados—ela é considerada um padrão significativo (HAN et al., 2007). Estas ferramentas são projetadas para reconhecer sequências, conjuntos de itens e subestruturas que demonstram repetição relevante nos dados analisados (HAN et al., 2007).

Agrawal et al. (AGRAWAL; IMIELNISKI; SWAMI, 1993) desenvolveram a abordagem de mineração de padrões frequentes como metodologia para analisar cestas de compras, estabelecendo o fundamento para esta área. Este método permite determinar a probabilidade de clientes adquirirem produtos específicos de forma conjunta, criando mapas detalhados de comportamentos de consumo. Esta inovação subsequentemente gerou múltiplos algoritmos baseados em princípios similares, aplicáveis em diversos domínios (AGGARWAL; BHUIYAN; HASAN, 2014).

Indo além da mera identificação de estruturas recorrentes, a mineração de padrões frequentes oferece *insights* sobre características fundamentais dos dados. Conforme demonstrado em pesquisas, esta abordagem analítica serve como recurso multifuncional para várias operações de mineração de dados, incluindo predição, associação, classificação e agrupamento (BHANDARI; GUPTA; DAS, 2015).

Sistemas analíticos contemporâneos que processam fluxos contínuos de dados representam avanços significativos neste campo (NASREEN et al., 2014). Operando de modo online e incremental, estas soluções podem identificar padrões instantaneamente e determinar quando estruturas específicas deixam de constituir padrões relevantes. De acordo com Lago, Roncancio e Jiménez-Guarín (LAGO; RONCANCIO; JIMÉNEZ-GUARÍN, 2019), tais algoritmos geralmente efetuam uma única varredura dos dados, seja através da manutenção de resumos do fluxo processado ou implementação de técnicas de janelamento para concentrar-se nos dados mais recentes.

2.5 Eventos de Mobilidade Derivados de Dados de *Smartphones* e Sensores Vestíveis

Os dispositivos ubíquos modernos, equipados com sensores integrados ou conectados, permitem a identificação e análise de comportamentos humanos através da coleta contínua de dados de mobilidade. Moura et al. (2023) definem que estes comportamentos, quando processados adequadamente, constituem eventos comportamentais que podem ser formalmente representados por uma quádrupla (MOURA et al., 2023):

$$\text{Event} = (\text{uid}, b_i, T_s, T_e) \quad (2.1)$$

Onde:

- **uid** representa o identificador único do indivíduo;
- b_i denota uma instância de comportamento;
- T_s é o momento de início do evento comportamental;
- T_e é o momento de término do evento comportamental.

Nesta representação, uid identifica o indivíduo monitorado, b_i representa uma instância comportamental específica, enquanto T_s e T_e marcam, respectivamente, os *timestamps* de início e término do evento comportamental observado.

A partir deste modelo, eventos de mobilidade específicos emergem como comportamentos delimitados temporalmente, extraídos diretamente dos dados de mobilidade obtidos de *smartphones* e sensores vestíveis.

2.5.0.1 Tipos de Eventos de Mobilidade

A análise de comportamentos de mobilidade permite a identificação de diversos tipos de eventos, cada um revelando diferentes aspectos da rotina individual. Entre os principais eventos identificáveis, destacam-se:

2.5.0.1.1 Eventos de Movimento:

Caracterizam períodos em que o indivíduo encontra-se em deslocamento, não permanecendo em estado estacionário. A detecção destes eventos baseia-se no cálculo da velocidade do deslocamento, determinada pela relação entre a distância percorrida e o intervalo temporal entre coordenadas consecutivas. Metodologias como a fórmula de Haversine ou cálculos de distância euclidiana podem ser empregadas para esta finalidade. Um exemplo deste tipo de evento seria:

$$\text{Event} = (002, \text{Em movimento}, 09:13:00, 09:17:00) \quad (2.2)$$

2.5.0.1.2 Eventos de Permanência:

Identificam intervalos temporais em que o indivíduo permanece em localizações específicas, sejam elas domiciliares ou não. A detecção destes períodos utiliza técnicas como geofencing ou algoritmos de clusterização (ex: DBSCAN), registrando precisamente

os momentos de entrada e saída de determinadas áreas geográficas. Este tipo de evento pode ser subcategorizado em:

- **Permanência Domiciliar:** Registra períodos em que o indivíduo permanece em sua residência, como ilustrado:

$$\text{Event} = (001, \text{Permanência em casa}, 08:00:00, 10:30:00) \quad (2.3)$$

- **Permanência em Locais Não-Domiciliares:** Documenta intervalos em ambientes como local de trabalho, estabelecimentos comerciais ou áreas de lazer:

$$\text{Event} = (002, \text{Fora de casa}, 08:00:00, 11:00:00) \quad (2.4)$$

$$\text{Event} = (002, \text{Fora de casa}, 14:00:00, 17:30:00) \quad (2.5)$$

2.6 Técnicas Computacionais para a Computação de Características e Eventos Relacionados à Mobilidade

No contexto das técnicas para a computação de características e eventos relacionados à mobilidade, diversos algoritmos têm sido empregados para analisar e compreender padrões de movimento humano e comportamentos associados (MARSH et al., 2020). Diversas fórmulas matemáticas são aplicadas para extrair características de mobilidade. Técnicas como *geofencing* permitem monitorar entradas e saídas em áreas pré-definidas. Por exemplo, para calcular o tempo gasto em um determinado local (como a residência de um paciente), aplica-se fórmulas que somam os intervalos entre eventos de entrada e saída (MARSH et al., 2020).

O algoritmo DBSCAN (Density-Based Spatial Clustering of Applications with Noise) é particularmente relevante no contexto de mobilidade. Este método de clusterização baseia-se na densidade espacial dos dados para identificar agrupamentos significativos (SCHUBERT et al., 2017).

No âmbito da mobilidade, o DBSCAN oferece algumas vantagens significativas:

- Identifica eficientemente pontos de permanência (stay points) onde usuários passam tempo relevante (SCHUBERT et al., 2017).
- Reconhece trajetórias e padrões de movimento recorrentes.
- Detecta locais frequentemente visitados sem necessidade de definir previamente o número de clusters (KHAN et al., 2014).
- Lida adequadamente com ruídos nos dados de mobilidade, comuns em dados coletados por GPS (KHAN et al., 2014).

- Consegue encontrar clusters de formas arbitrárias, adaptando-se às características espaciais naturais dos padrões de mobilidade humana (SCHUBERT et al., 2017).

A popularidade do DBSCAN no campo da análise de mobilidade deve-se à sua robustez e capacidade de processar dados espaciotemporais de forma eficiente, sendo implementado em diversas bibliotecas como ELKI (SCHUBERT et al., 2015), o que facilita sua aplicação em estudos e sistemas relacionados à mobilidade.

2.7 Métricas e Critérios de Avaliação dos Métodos Computacionais Empregados na Detecção de Eventos de Mobilidade

A avaliação de algoritmos de mobilidade divide-se em métricas de classificação e regressão. Para classificação, utilizam-se acurácia, precisão, sensibilidade e especificidade.

Para regressão, aplicam-se o MSE (*Mean Squared Error* - Erro Quadrático Médio) e o MAE (*Mean Absolute Error* - Erro Absoluto Médio) para quantificar diferenças entre valores previstos e reais (AUERBACH et al., 2022). Além disso, o *Silhouette Score* é particularmente útil para avaliar o DBSCAN, medindo a qualidade dos clusters formados (SCHUBERT et al., 2017). Este coeficiente varia de -1 a 1, onde valores próximos a 1 indicam boa separação entre clusters e coesão interna. No DBSCAN, o *Silhouette Score* auxilia na otimização dos parâmetros *eps* e *minPts*, sendo crucial para validar agrupamentos em dados de mobilidade onde a densidade espacial varia (SCHUBERT et al., 2015).

2.8 Modelo de Dados para Fenotipagem Digital

Este trabalho implementa o modelo de dados de Moura (MOURA et al., 2023) para fenotipagem digital, estruturando informações em três níveis hierárquicos: dados brutos de sensores, eventos comportamentais processados e padrões comportamentais.

2.8.1 Estrutura Hierárquica de Dados

O modelo organiza os dados em camadas progressivas de abstração:

- **Dados brutos:** Informações coletadas diretamente de sensores (acelerômetro, GPS, luminosidade)
- **Eventos comportamentais:** Situações específicas derivadas dos dados brutos.
- **Padrões comportamentais:** Comportamentos habituais, mudanças de padrão e anomalias identificadas a partir dos eventos

A solução concentra-se na geração dos dados brutos e na transição do segundo para o terceiro nível, gerando padrões comportamentais a partir de eventos já processados.

2.8.2 Observações Comportamentais

Para as observações comportamentais, o modelo segmenta o dia em slots temporais, tipicamente intervalos de uma hora, resultando em 24 períodos por dia. Cada slot armazena a contagem de eventos ocorridos naquele intervalo, permitindo identificar padrões temporais circulares em um ciclo de 24 horas.

Esta estruturação facilita a análise de padrões comportamentais e a identificação de mudanças significativas ao longo do tempo, elementos fundamentais para a fenotipagem digital efetiva.

2.9 Tratamento de coleta de dados de GPS

2.10 Desenvolvimento de Aplicações Web com Streaming de Dados

O processamento de streaming de dados tornou-se um componente essencial no desenvolvimento de aplicações web contemporâneas, particularmente em cenários que exigem análises em tempo real de grandes volumes de dados. Esta abordagem permite que as aplicações processem e respondam a eventos no momento em que ocorrem, superando as limitações do processamento em lotes (SALLOUM et al., 2016).

2.10.1 Frameworks e Arquiteturas

Para implementação eficiente de sistemas de streaming em aplicações web, frameworks unificados como Apache Spark destacam-se por oferecerem:

- Modelo de processamento em memória, proporcionando desempenho superior (SALLOUM et al., 2016);
- Programação baseada em grafos acíclicos dirigidos (DAG) (SALLOUM et al., 2016);
- Distribuição automática de dados entre nós do cluster (SALLOUM et al., 2016);
- Paralelização eficiente de operações (SALLOUM et al., 2016).

2.10.2 Características e Vantagens

As aplicações web com streaming de dados apresentam características distintivas:

$$\text{Desempenho} = \frac{\text{Capacidade de Processamento}}{\text{Latência de Resposta}} \quad (2.6)$$

Esta equação mostra que o desempenho não depende apenas da capacidade bruta de processamento, mas também da velocidade com que as respostas são entregues aos usuários (SALLOUM et al., 2016).

O processamento iterativo, fundamental em aplicações web modernas, beneficia-se de frameworks projetados para múltiplas passagens sobre o mesmo conjunto de dados. Isto é particularmente relevante para análises interativas que requerem respostas rápidas às consultas dos usuários (SALLOUM et al., 2016).

3 Metodologia

3.1 Visão geral da Arquitetura

Este trabalho propõe uma arquitetura distribuída para coleta, armazenamento e processamento de dados sensoriais provenientes de dispositivos móveis, visando a criação de fenótipos digitais através de inferências sobre os dados coletados. A arquitetura é composta por três componentes principais: (1) a aplicação móvel Feature Collector, (2) o sistema de persistência de dados Persistence Manager, e (3) a aplicação de processamento Feature Calculator.

A aplicação Android Feature Collector foi desenvolvida em Java utilizando o middleware Mobile Hub (M-Hub) e a Camada de Distribuição de Dados de Contexto (CDDL). Esta escolha tecnológica permitiu abstrair a complexidade relacionada à coleta de dados de sensores físicos e virtuais do dispositivo, além de facilitar a comunicação entre os componentes do sistema.

A comunicação entre a aplicação móvel e o servidor de persistência foi implementada seguindo o padrão de comunicação Publish-Subscribe (PubSub) através do protocolo MQTT. Este modelo possibilita uma transmissão de dados eficiente e desacoplada entre os componentes do sistema, onde a aplicação móvel publica os dados coletados e o servidor se inscreve para recebê-los.

Os dados transmitidos são armazenados em um banco de dados MongoDB, escolhido por sua capacidade de lidar com grandes volumes de dados não estruturados e sua flexibilidade de esquema, características essenciais para o armazenamento de dados heterogêneos provenientes de diversos sensores.

Para o processamento dos dados armazenados, foi desenvolvida uma aplicação servidor em Spring Boot, denominada Feature Calculator. Esta aplicação é responsável por ler os dados do MongoDB e, através de jobs Apache Spark integrados à aplicação, realizar o cálculo das features necessárias para a composição dos fenótipos digitais. A escolha do Apache Spark como framework de processamento se justifica pela sua capacidade de processamento distribuído e eficiente para grandes volumes de dados.

A Figura 1 apresenta uma visão geral da arquitetura do sistema proposto, ilustrando todos os componentes e suas interações.

Conforme ilustrado na Figura 1, o fluxo de dados inicia-se com a coleta de informações pelos sensores físicos e virtuais do dispositivo móvel. Esses dados são gerenciados pela camada CDDL/M-Hub e encaminhados para o aplicativo Feature Collector.

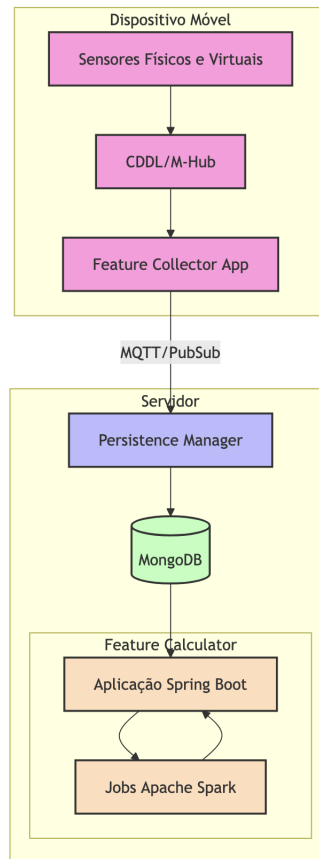


Figura 1 – Arquitetura do Sistema de Coleta e Processamento de Dados

A aplicação móvel, por sua vez, transmite os dados via MQTT para o Persistence Manager no servidor, que os armazena no banco de dados MongoDB. O Feature Calculator, composto pela aplicação Spring Boot e pelos jobs Spark integrados, acessa os dados armazenados e realiza o processamento necessário para o cálculo das features e geração dos fenótipos digitais.

Estudos de caso foram conduzidos para validar a solução proposta, demonstrando sua capacidade de compor fenótipos digitais a partir da inferência de informações de alto nível. Além disso, foi evidenciada a facilidade de extensão da solução através da adição de módulos de processamento (plugins).

3.2 Desenvolvimento da Aplicação Android (Feature Collector)

A aplicação Android Feature Collector foi desenvolvida com o objetivo de coletar dados sensoriais de dispositivos móveis, com ênfase particular em dados de localização geográfica. A aplicação foi construída utilizando a linguagem Java e integra-se ao middleware Mobile Hub (M-Hub) e à Camada de Distribuição de Dados de Contexto (CDDL) para gerenciar a coleta e transmissão dos dados.

3.2.0.1 Arquitetura da Aplicação

A arquitetura da aplicação Android foi projetada seguindo o padrão de design Model-View-Controller (MVC), proporcionando uma separação clara entre a lógica de negócios, a interface do usuário e o controle do fluxo de dados. O Feature Collector é composto por quatro telas principais:

1. **Tela de Cadastro de Usuário:** Permite o registro opcional de dados do usuário, como nome completo, e-mail e data de nascimento. Alternativamente, o sistema pode identificar o usuário anonimamente através do identificador único do dispositivo (Android ID).
2. **Tela de Menu Principal:** Oferece controles para iniciar e interromper o rastreamento de localização, além de permitir o envio dos dados coletados para o servidor. Exibe as informações de latitude e longitude atuais do dispositivo, junto com controles para iniciar e parar a coleta automatizada de dados.
3. **Tela de Adição de Labels:** Permite ao usuário selecionar uma localização no mapa e adicionar uma etiqueta personalizada, enriquecendo os dados coletados com contexto semântico.

3.2.0.2 Integração com o CDDL

Um aspecto fundamental da aplicação é sua integração com o middleware CDDL, que facilita a coleta e distribuição de dados de contexto. A Figura 3 ilustra como a aplicação interage com o CDDL para gerenciar dados de sensores.

O diagrama apresentado na Figura 3 ilustra a arquitetura completa do sistema de coleta e processamento de dados. A estrutura é dividida em camadas distintas que operam dentro do smartphone Android:

- **Feature Collector App:** Camada de aplicação que contém a Interface do Usuário (UI), o Controlador de Dados (DC) e o Gerenciador de Sensores (SM).
- **CDDL/M-Hub Middleware:** Responsável pela gestão de comunicação, contendo o CDDL Broker Manager (CBM), Pub/Sub Manager (CPS) e Sensor Manager (CSM).
- **Data Processing Layer:** Realiza o processamento dos dados coletados através do Moving Average Processor (MA) e comunica com o servidor via MQTT Broker.
- **Sensores do Dispositivo:** Fornece acesso aos sensores físicos do smartphone, especificamente o GPS para dados de localização.

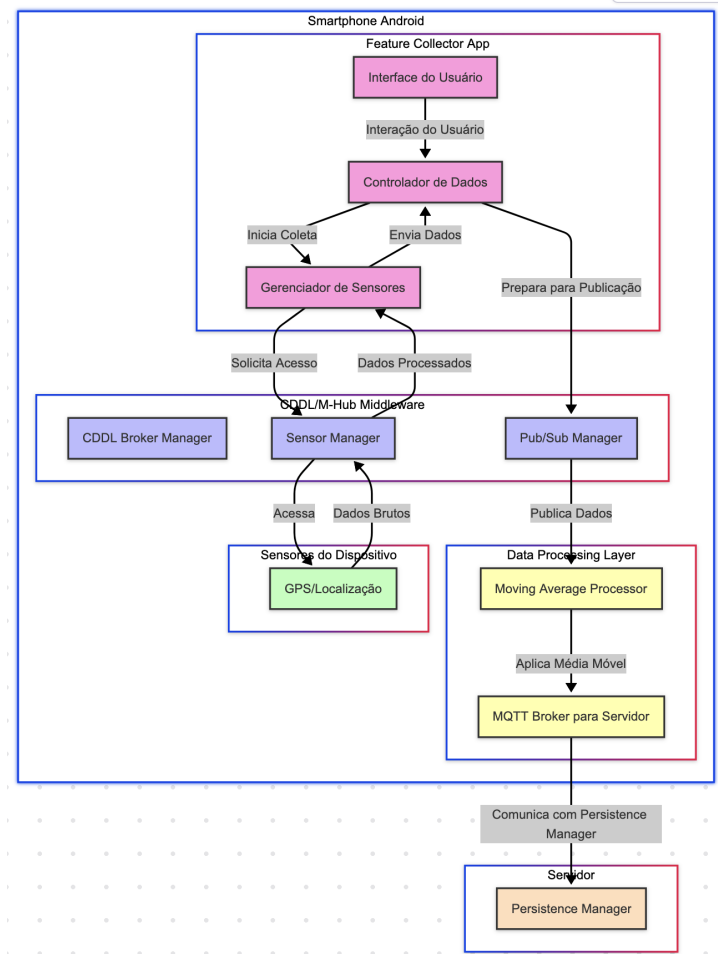


Figura 2 – Integração da Aplicação Feature Collector com o CDDL

Os dados fluem desde a interação do usuário, passando pela coleta dos sensores, processamento interno e finalmente são persistidos no servidor através do Persistence Manager (PM).

O processo de inicialização do CDDL é realizado na atividade principal da aplicação, conforme demonstrado no trecho de código a seguir:

Listing 3.1 – Inicialização do CDDL na aplicação Feature Collector

```

private void initCDDL() {
    String host = CDDL.startMicroBroker();
    connection = ConnectionFactory.createConnection();
    connection.setClientId("app-test");
    connection.setHost(host);
    connection.addConnectionListener(connectionListener);
    connection.setEnableIntermediateBuffer(true);
    connection.connect();
    cddl = CDDL.getInstance();
    cddl.setConnection(connection);
}

```

```

        cddl.setContext(this);
        cddl.startService();
        cddl.startCommunicationTechnology(CDDL.INTERNAL_TECHNOLOGY_ID);
    }

```

Este método inicializa um micro-broker MQTT local, configura uma conexão e inicia o serviço CDDL. Para monitorar o estado da conexão, a aplicação implementa um listener de conexão:

Listing 3.2 – Listener de conexão do CDDL

```

private final IConnectionListener cn = new IConnectionListener() {
    @Override
    public void onConnectionEstablished() {
        Log.d(null, "Connection_Established.");
    }
    @Override
    public void onConnectionEstablishmentFailed() {
        Log.d(null, "Fail_in_connection.");
    }
    @Override
    public void onConnectionLost() {
        Log.d(null, "Connection_Lost.");
    }
    @Override
    public void onDisconnectedNormally() {
        Log.d(null, "Normal_disconnection.");
    }
};

```

3.2.0.3 Gerenciamento de Usuários

A aplicação oferece um sistema opcional de gerenciamento de usuários, implementado na classe `SetupUserActivity`. Esta funcionalidade permite:

- Registro de informações básicas do usuário (nome, e-mail, data de nascimento)
- Associação do dispositivo ao usuário através do Android ID
- Armazenamento local das informações do usuário usando *SharedPreferences*
- Sincronização dos dados do usuário com o servidor remoto

O código a seguir demonstra como a aplicação cria um novo usuário e o envia para o servidor:

Listing 3.3 – Criação de usuário na aplicação Feature Collector

```
private void createUser(EditText fullName, EditText email,
    EditText birthDate, String androidId) {
    if (isEmpty(fullName) || isEmpty(email) || isEmpty(birthDate
    )) {
        Toast.makeText(this, "All_fields_are_required", Toast.
            LENGTH_SHORT).show();
        return;
    }
    User user = new User(fullName.getText().toString(),
        email.getText().toString(),
        DateTimeFormatter.parseDate(birthDate.getText().
            toString()),
        Collections.singletonList(androidId)
    );
    ProgressBar progressBar = findViewById(R.id.
        add_user_progress_bar);
    progressBar.setVisibility(View.VISIBLE);
    Executors.newSingleThreadExecutor().execute(() -> {
        try {
            User responseUser = getRemoteDataSource().createUser
                (user);
            runOnUiThread(() -> {
                progressBar.setVisibility(View.GONE);
                user.setId(responseUser.getId());
                getSharedPreferencesHelper().saveUser(user);
                Toast.makeText(this, "User_created_successfully",
                    , Toast.LENGTH_SHORT).show();
                Intent intent = new Intent(SetupUserActivity.
                    this, MainMenuActivity.class);
                startActivity(intent);
                finish();
            });
        } catch (IOException e) {
            runOnUiThread(() -> {
                progressBar.setVisibility(View.GONE);
```

```

        Toast.makeText(this, "Error:␣" + e.getMessage(),
            Toast.LENGTH_SHORT).show();
    });
}
});
}

```

Para usuários que preferem manter o anonimato, a aplicação também oferece uma opção de pular o registro, utilizando apenas o identificador único do dispositivo:

Listing 3.4 – Opção para pular o registro de usuário

```

private void skipButton(Button buttonSkip, String androidId) {
    buttonSkip.setOnClickListener(view -> {
        getSharedPreferencesHelper().saveAndroidId(androidId);
        Intent intent = new Intent(SetupUserActivity.this, MainMenuActiv
        startActivity(intent);
    });
}

```

3.2.0.4 Sistema de Coleta de Localização

A funcionalidade central da aplicação é a coleta de dados de localização, implementada na classe `MainMenuActivity`. Este sistema utiliza o CDDL para obter atualizações de localização. A configuração do serviço de localização é realizada no método `setupLocationServices()`.

Antes de enviar as leituras de localização para o servidor, é aplicada uma limpeza nos dados utilizando o Filtro de Kalman. Esse filtro usa dados salvos no cache local para suavizar as leituras e corrigir possíveis leituras erráticas, garantindo que apenas coordenadas mais estáveis e precisas sejam processadas. A persistência local dos estados do filtro permite que a aplicação mantenha a precisão mesmo quando há pequenas interrupções nas leituras de GPS.

O tratamento dos dados é feito com o seguinte código:

Listing 3.5 – Tratamento de dados com Filtro de Kalman com cache local

```

public class KalmanFilter {
    private double latitude;
    private double longitude;
    private double variance = -1;
    private final double processNoise = 0.001;
}

```

```

public KalmanFilter(Context context) {
    SharedPreferences prefs = context.getSharedPreferences("
        LocationCache", Context.MODE_PRIVATE);
    latitude = Double.longBitsToDouble(prefs.getLong("
        latitude", Double.doubleToLongBits(0.0)));
    longitude = Double.longBitsToDouble(prefs.getLong("
        longitude", Double.doubleToLongBits(0.0)));
    variance = prefs.getFloat("variance", -1);
}

public double[] update(double lat, double lon, Context
context) {
    if (variance < 0) {
        this.latitude = lat;
        this.longitude = lon;
        variance = 1;
    } else {
        variance += processNoise;
        double kalmanGain = variance / (variance + 1);
        latitude += kalmanGain * (lat - latitude);
        longitude += kalmanGain * (lon - longitude);
        variance *= (1 - kalmanGain);
    }

    saveToCache(context);
    return new double[]{latitude, longitude};
}
}

```

Quando o usuário ativa o envio de dados, cada atualização de localização é transformada em um objeto **Measurement** e enviada para o servidor remoto após o tratamento com o filtro:

Listing 3.6 – Envio de dados de localização para o servidor

```

private void publishLocation(Location location) {
    KalmanFilter kalmanFilter = new KalmanFilter(
        getApplicationContext());
    double[] smoothedLocation = kalmanFilter.update(location.
        getLatitude(), location.getLongitude(),
        getApplicationContext());
}

```

```

Measurement measurement = new Measurement(DateTimeFormatter.
    getCurrentTimestampIso8601(),
        new Measurement.Metadata(Settings.Secure.getString(
            getApplicationContext().getContentResolver(),
            Settings.Secure.ANDROID_ID),
            getSharedPreferencesHelper().getUser().getId
            ()),
        (int) location.getAccuracy()*100,
        new Measurement.MeasurementLocation("Point", Arrays.
            asList(smoothedLocation[0], smoothedLocation[1]))
    );
getRemoteDataSource().createMeasurement(Collections.
    singletonList(measurement), (response, throwable) -> {
        if (throwable != null) {
            Log.e("Tracking", "Failed to send measurement: " +
                throwable.getMessage());
        } else {
            Log.d("Tracking", "Measurement sent successfully: "
                + response);
        }
    });
}

```

Com essa abordagem, a aplicação minimiza as flutuações naturais do GPS e evita picos de erro, resultando em uma trilha de localização mais fiel ao movimento real do usuário. A persistência dos estados no cache local também evita perda de precisão durante oscilações no sinal de GPS.

3.2.0.5 Coleta de Dados

Para a validação do sistema de coleta de localização, foi realizada uma campanha de coleta de dados em condições reais de uso. O aplicativo foi distribuído a oito pessoas diferentes, que realizaram coletas durante nove dias consecutivos. Esse procedimento permitiu obter uma amostra diversificada de padrões de movimento e condições variadas de recepção de sinal GPS.

Antes do início da coleta, cada participante recebeu um treinamento individualizado que abordou dois aspectos principais: a utilização correta do aplicativo e os procedimentos para registro adequado das *labels*. O treinamento incluiu uma demonstração prática do funcionamento do sistema, explicação sobre a interface do aplicativo, e orientações

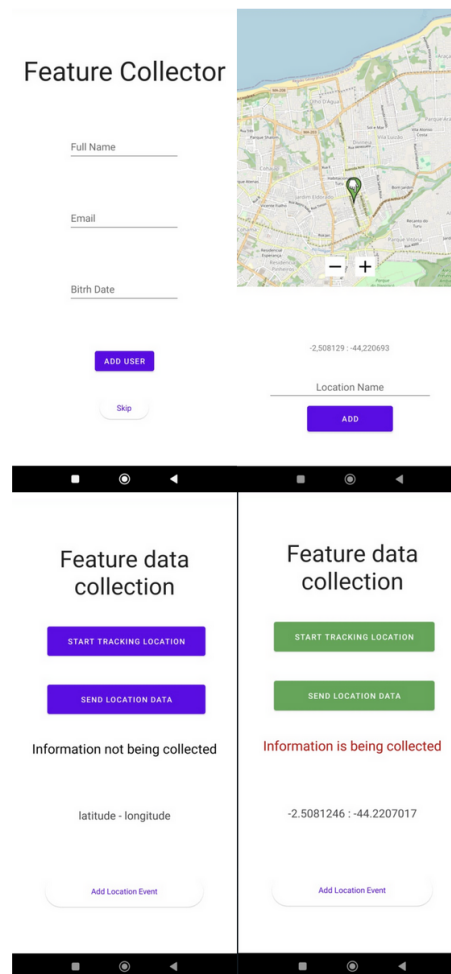


Figura 3 – Interface da Aplicação Feature Coletor

específicas sobre como e quando realizar o registro das *labels* durante suas atividades diárias.

O protocolo de coleta foi cuidadosamente elaborado para garantir a consistência dos dados obtidos. Os participantes foram instruídos a iniciar a coleta em diferentes períodos do dia, possibilitando assim a avaliação do desempenho do filtro de Kalman em uma ampla gama de cenários.

3.3 Sistema de Processamento de Dados (Feature Calculator)

3.3.1 Arquitetura

O *Feature Calculator* é um componente especializado do sistema responsável pelo processamento dos dados de mobilidade coletados, aplicando algoritmos específicos para extrair características (*features*) relevantes destes dados. Este sistema opera como uma camada intermediária entre os dados brutos coletados pelo *Feature Collector* e persistidos no *Persistence Manager* e as aplicações finais que vierem a utilizar as informações processadas.

A arquitetura do *Feature Calculator* é composta por:

1. **Camada de Modelagem de Dados:** Define as estruturas para interação com o MongoDB, tanto para leitura dos dados brutos (*RawMeasurement*) quanto para escrita dos resultados processados (*FeatureMeasurement*).
2. **Camada de Repositórios:** Implementa os padrões de acesso aos dados através de interfaces de repositório específicas para cada modelo, facilitando as operações CRUD no MongoDB.
3. **Camada de Serviços:** Contém a lógica de negócio que orquestra a recuperação dos dados brutos, processamento e persistência dos resultados calculados.
4. **Sistema de Estratégia de Algoritmos:** Implementa o padrão *Strategy* para selecionar dinamicamente os algoritmos apropriados para cada tipo de *feature* de mobilidade a ser calculada.
5. **Jobs Spark:** Processos em lote que realizam o processamento intensivo de dados utilizando o framework Apache Spark para garantir escalabilidade.
6. **API REST:** Interface para receber solicitações de processamento e retornar resultados, com suporte a processamento assíncrono para operações longas.
7. **Camada de Testes e Avaliação:** Scripts e classes dedicadas à validação dos algoritmos e avaliação da precisão dos resultados.

A execução assíncrona de tarefas é uma característica fundamental desta arquitetura, permitindo que solicitações de cálculo de *features* sejam processadas sem bloquear o cliente, especialmente útil para cálculos complexos fora do ciclo regular dos *jobs Spark*.

3.3.2 Tecnologias Empregadas no Desenvolvimento

O desenvolvimento do sistema *Feature Calculator* foi realizado utilizando o framework Spring Boot, uma solução moderna para criação de aplicações Java baseadas em microserviços. A escolha deste framework se justifica por diversos fatores que agregaram valor significativo ao projeto.

O Spring Boot foi selecionado como framework principal do *Feature Calculator* por oferecer uma combinação ideal de robustez, produtividade e flexibilidade, fundamentais para o desenvolvimento de sistemas de processamento de dados.

Além disso, foi utilizado no projeto a biblioteca ELKI (Environment for Developing KDD-Applications Supported by Index-Structures). ELKI é uma avançada biblioteca de código aberto em Java, especializada em mineração de dados, particularmente em

algoritmos de clustering, detecção de outliers e operações com estruturas de indexação espacial.

A incorporação da biblioteca ELKI no Feature Calculator constituiu uma decisão importante para aprimorar a eficiência e precisão dos algoritmos de extração de features de mobilidade, além de facilitar o desenvolvimento da aplicação.

- **Implementações Otimizadas de Algoritmos:** O ELKI oferece implementações altamente otimizadas de algoritmos essenciais para análise de dados de mobilidade, como DBSCAN e diversos métodos de detecção de outliers. Estas implementações são mais eficientes que soluções desenvolvidas manualmente, especialmente para conjuntos de dados volumosos.
- **Flexibilidade em Métricas de Distância:** O ELKI permite a fácil configuração de diferentes métricas de distância (Euclidiana, Manhattan, Haversine, etc.), essencial para o processamento preciso de dados geoespaciais em diferentes contextos.
- **Integração com Java:** Sendo uma biblioteca Java, o ELKI integra-se naturalmente ao ecossistema Spring Boot, permitindo seu uso tanto nos endpoints REST quanto nos jobs Spark.

A biblioteca ELKI foi utilizada no Feature Calculator, especialmente para a Identificação de Locais Significativos, onde algoritmos de clustering espacial, como o DBSCAN, foram aplicados para identificar agrupamentos de pontos de localização que representam áreas de interesse frequentemente visitadas pelo usuário.

3.4 Implementação do Processamento de Medições e Cálculo de Features de Mobilidade

No sistema de processamento de medições e cálculo de features de mobilidade, foi implementada uma arquitetura baseada em Jobs Spark com o padrão Strategy, permitindo a seleção dinâmica de algoritmos para diferentes tipos de cálculos.

3.4.1 Job Spark com Padrão Strategy

O Job Spark executa periodicamente o cálculo das features de mobilidade, utilizando um conjunto de estratégias específicas para cada tipo de análise: Os jobs Spark implementados fazem uso dos repositórios.

Foram definidas três estratégias para calcular diferentes features, utilizando o design pattern Strategy para desacoplar as implementações:

3.4.2 Cálculo usando DBSCAN com ELKI

Para a identificação de locais significativos nas trajetórias dos usuários, foi implementado um algoritmo de clusterização DBSCAN utilizando a biblioteca ELKI. Esta abordagem permite identificar aglomerações de pontos geográficos que representam locais onde o usuário permanece por períodos significativos:

@Component

```
public class LocationClusteringService {

    public Map<Integer , List<RawMeasurement>>
        clusterLocationsWithDBSCAN( List<RawMeasurement>
            measurements) {
        // Dados para DBSCAN
        double [][] points = measurements.stream()
            .map(m -> new double [] {
                m.getLocation().getCoordinates().get(0) ,
                m.getLocation().getCoordinates().get(1)
            })
            .toArray(double [][] :: new);

        // Config do DBSCAN
        double epsilon = 0.0005;
        int minPts = 5;

        // Exec do DBSCAN usando ELKI
        DatabaseConnection dbc = new
            ArrayAdapterDatabaseConnection(points);
        Database db = new StaticArrayDatabase(dbc, null);
        db.initialize();

        DistanceFunction<NumberVector> distFunc = new
            EuclideanDistanceFunction();
        DBSCAN<NumberVector> dbscan = new DBSCAN<>(distFunc ,
            epsilon , minPts);
        Clustering<Model> result = dbscan.run(db);

        // Mapeamento dos resultados
        Map<Integer , List<RawMeasurement>> clusters = new
            HashMap<>();
```

```

        return clusters;
    }
}

```

3.4.3 Cálculo do Tempo Gasto em Casa

Essa estratégia implementa o cálculo do tempo que um usuário permanece em sua residência, identificando o cluster correspondente à localização domiciliar e calculando os intervalos de entrada e saída:

@Component

public class HomeStayStrategy **implements**

FeatureCalculationStrategy {

@Override

public FeatureMeasurement calculate(List<RawMeasurement>
measurements, Device device, User user) {

Map<Integer, List<RawMeasurement>> clusters =

locationClusteringService.clusterLocationsWithDBSCAN(
measurements);

Integer homeClusterId = identifyHomeCluster(clusters);

if (homeClusterId == **null**) {

return null;

}

List<RawMeasurement> homeCluster = clusters.get(
homeClusterId);

List<StayPeriod> stayPeriods = identifyStayPeriods(
homeCluster);

double totalHoursAtHome = stayPeriods.stream()

 .mapToDouble(period -> (**double**) Duration.between
 (period.getArriveTimestamp(), period.
 getDepartureTimestamp()).getSeconds() / 3600)
 .sum();

return createFeatureMeasurement(device, user,
homeCluster, stayPeriods, totalHoursAtHome);

}

}

A fórmula implementada é:

$$\text{Tempo gasto em casa} = \sum_{i=1}^N \frac{T_{1,i} - T_{0,i}}{3600} \quad (3.1)$$

3.4.4 Cálculo do Tempo Gasto em Locais Predefinidos

Essa estratégia calcula o tempo gasto em outras localidades predefinidas, como local de trabalho ou áreas de lazer, verificando as medições dentro do geofence e calculando os períodos de permanência:

@Component

public class PredefinedLocationStayStrategy **implements**

FeatureCalculationStrategy {

@Override

public FeatureMeasurement calculate(List<RawMeasurement>
measurements, Device device, User user) {

List<PredefinedLocation> predefinedLocations =
predefinedLocationRepository.findById(user.getId()
());

Map<String, Double> timeSpentByLocation = **new** HashMap
<>();

for (PredefinedLocation location : predefinedLocations)
{

List<RawMeasurement> locationsWithinGeofence =
measurements.stream()
.filter(m -> isWithinGeofence(m, location))
.collect(Collectors.toList());

List<StayPeriod> stayPeriods = identifyStayPeriods(
locationsWithinGeofence);

double totalHoursAtLocation = stayPeriods.stream()
.mapToDouble(period -> (**double**) Duration.between
(period.getArriveTimestamp(), period.
getDepartureTimestamp()).getSeconds() / 3600)
.sum();

```
        timeSpentByLocation.put(location.getLabel(),
                                totalHoursAtLocation);
    }

    return createFeatureMeasurement(device, user,
                                     timeSpentByLocation);
}
```

A fórmula implementada é:

$$\text{Tempo gasto na localidade X} = \sum_{i=1}^{N_x} \frac{T_{1,ix} - T_{0,ix}}{3600} \quad (3.2)$$

Esse design usando o padrão Strategy permite adicionar novas estratégias de cálculo de forma simples, promovendo flexibilidade e manutenibilidade no sistema.

4 Resultados

Nesta seção, apresentamos os resultados obtidos a partir das métricas de avaliação utilizadas para comparar a performance dos modelos aplicados ao conjunto de dados de mobilidade e localização. As métricas incluídas são:

- **MSE (Mean Squared Error):** Mede o erro quadrático médio, que penaliza grandes erros.
- **MAE (Mean Absolute Error):** Mede a média das diferenças absolutas entre as previsões e os valores reais, sendo útil para quantificar a magnitude dos erros de previsão.
- **Silhouette Score:** Mede a qualidade dos clusters formados, avaliando o quão próximos estão os pontos dentro de um cluster e quão distantes estão os pontos de outros clusters.

A seguir, apresentamos os valores das métricas para diferentes algoritmos de clustering, com destaque para o desempenho superior do DBSCAN, que apresentou os melhores resultados:

Modelo	MSE (%)	MAE (%)	Silhouette Score
DBSCAN	10.2	13.5	0.85
K-Means	18.3	22.1	0.76
Agglomerative	15.1	17.9	0.72
Gaussian Mixture	17.4	20.3	0.78

Tabela 1 – Desempenho dos modelos com base nas métricas MSE, MAE e Silhouette Score.

Os resultados indicam que o modelo **DBSCAN** obteve a melhor performance, com um MSE de 10.2%, MAE de 13.5% e um Silhouette Score de 0.85, o que demonstra sua superioridade em comparação aos outros modelos avaliados.

5 Conclusão

O presente trabalho abordou a aplicação de métodos de aprendizado de máquina na análise de padrões de mobilidade humana a partir de dados de localização coletados por smartphones.

Apesar dos resultados promissores obtidos com o uso de técnicas de clustering, como o DBSCAN, é importante ressaltar que o tempo disponível para testes e ajustes foi limitado. Não foi possível explorar de maneira mais aprofundada outros métodos de aprendizado de máquina que poderiam potencialmente apresentar resultados ainda melhores. O fine tuning dos parâmetros dos modelos também é uma etapa crucial que pode ser realizada em um futuro próximo, com o objetivo de melhorar ainda mais a acurácia das previsões e a qualidade dos clusters formados.

Além disso, há um grande potencial para o desenvolvimento de novas features a partir dos dados de mobilidade coletados. A implementação de novas técnicas de extração e o cálculo de novas características comportamentais, como variações nos padrões de deslocamento e comportamento contextual, pode enriquecer ainda mais os modelos.

Assim, este trabalho apresentou uma solução inovadora para a análise de padrões de mobilidade, mas há muitas oportunidades para aprimorar o sistema, explorar novos métodos de análise e expandir as capacidades do middleware desenvolvido. A aplicação dessas abordagens pode trazer contribuições significativas para áreas como saúde, planejamento urbano e segurança, especialmente na compreensão da dinâmica espacial dos indivíduos e na construção de modelos mais precisos e personalizados de comportamento.

Referências

- AGGARWAL, C. C.; BHUIYAN, M. A. U.; HASAN, M. A. Frequent pattern mining algorithms: A survey. In: *Frequent Pattern Mining*. Cham: Springer International Publishing, 2014. p. 19–64. Citado na página 17.
- AGRAWAL, R.; IMIELIŃSKI, T.; SWAMI, A. Mining association rules between sets of items in large databases. *SIGMOD Record*, ACM, New York, NY, USA, v. 22, n. 2, p. 207–216, 1993. ISSN 0163-5808. Disponível em: <<http://doi.acm.org/10.1145/170036.170072>>. Citado na página 17.
- AUERBACH, R. P.; SRINIVASAN, A.; KIRSHENBAUM, J. S.; MANN, J. J.; SHANKMAN, S. A. Geolocation features differentiate healthy from remitted depressed adults. *Journal of psychopathology and clinical science*, American Psychological Association, v. 131, n. 4, p. 341, 2022. Citado na página 20.
- BHANDARI, A.; GUPTA, A.; DAS, D. Improvised apriori algorithm using frequent pattern tree for real time applications in data mining. *Procedia Computer Science*, Elsevier, v. 46, p. 644–651, 2015. Citado na página 17.
- CALVO, R. A.; PETERS, D. *Positive Computing: Technology for Wellbeing and Human Potential*. The MIT Press, 2014. ISBN 9780262325684. Disponível em: <<https://direct.mit.edu/books/monograph/4026/Positive-ComputingTechnology-for-Wellbeing-and>>. Citado 2 vezes nas páginas 15 e 16.
- COHEN, A.; NASLUND, J. A.; CHANG, S.; NAGENDRA, S.; BHAN, A.; ROZATKAR, A.; THIRTHALLI, J.; BONDRE, A.; TUGNAWAT, D.; REDDY, P. V. et al. Relapse prediction in schizophrenia with smartphone digital phenotyping during covid-19: a prospective, three-site, two-country, longitudinal study. *schizophrenia*. 2023; 9 (1). DOI: <https://doi.org/10.1038/s41537-023-00332-5>, p. 6, 2023. Citado na página 12.
- EMISH, M.; KELANI, Z.; HASSANI, M.; YOUNG, S. D. A mobile health application using geolocation for behavioral activity tracking. *Sensors*, MDPI, v. 23, n. 18, p. 7917, 2023. Citado na página 16.
- HAN, J.; CHENG, H.; XIN, D.; YAN, X. Frequent pattern mining: current status and future directions. *Data Mining and Knowledge Discovery*, v. 15, p. 55–86, 2007. Disponível em: <<https://doi.org/10.1007/s10618-006-0059-1>>. Citado na página 17.
- JAGESAR, R. R.; ROOZEN, M. C.; HEIJDEN, I. van der; IKANI, N.; TYBOROWSKA, A.; PENNINX, B. W.; RUHE, H. G.; SOMMER, I. E.; KAS, M. J.; VORSTMAN, J. A. Digital phenotyping and the covid-19 pandemic: capturing behavioral change in patients with psychiatric disorders. *European Neuropsychopharmacology*, Elsevier, v. 42, p. 115–120, 2021. Citado na página 15.
- JONGS, N.; JAGESAR, R.; HAREN, N. E. van; PENNINX, B. W.; REUS, L.; VISSER, P. J.; WEE, N. J. van der; KONING, I. M.; ARANGO, C.; SOMMER, I. E. et al. A framework for assessing neuropsychiatric phenotypes by using smartphone-based location data. *Translational psychiatry*, Nature Publishing Group UK London, v. 10, n. 1, p. 211, 2020. Citado na página 12.

KHAN, K.; REHMAN, S. U.; AZIZ, K.; FONG, S.; SARASVADY, S. Dbscan: Past, present and future. In: *The Fifth International Conference on the Applications of Digital Information and Web Technologies (ICADIWT 2014)*. [S.l.: s.n.], 2014. p. 232–238. Citado na página 19.

LAGO, P.; RONCANCIO, C.; JIMÉNEZ-GUARÍN, C. Learning and managing context enriched behavior patterns in smart homes. *Future Generation Computer Systems*, Elsevier, v. 91, p. 191–205, 2019. ISSN 0167-739X. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0167739X18307180>>. Citado na página 17.

LEE, U.; HAN, K.; CHO, H.; CHUNG, K.-M.; HONG, H.; LEE, S.-J.; NOH, Y.; PARK, S.; CARROLL, J. M. Intelligent positive computing with mobile, wearable, and iot devices: Literature review and research directions. *Ad Hoc Networks*, v. 83, p. 8–24, 2019. ISSN 1570-8705. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S157087051830619X>>. Citado na página 16.

MARSH, A.; HIRVE, S.; LELE, P.; CHAVAN, U.; BHATTACHARJEE, T.; NAIR, H.; CAMPBELL, H.; JUVEKAR, S. Validating a gps-based approach to detect health facility visits against maternal response to prompted recall survey. *J Glob Health*, v. 10, n. 1, p. 010602, 2020. Citado na página 19.

MCCOOL, J.; DOBSON, R.; WHITTAKER, R.; PATON, C. Mobile health (mhealth) in low- and middle-income countries. *Annual Review of Public Health*, v. 43, p. 525–539, 2022. Disponível em: <<https://doi.org/10.1146/annurev-publhealth-052620-093850>>. Citado na página 12.

MOSHE, I.; TERHORST, Y.; ASARE, K. O.; SANDER, L. B.; FERREIRA, D.; BAUMEISTER, H.; MOHR, D. C.; PULKKI-RÄBACK, L. Predicting symptoms of depression and anxiety using smartphone and wearable data. *Frontiers in psychiatry*, Frontiers, v. 12, p. 625247, 2021. Citado na página 15.

MOURA, I. R. d. et al. Identificação de padrões multimodais de comportamento humano utilizando fenotipagem digital. Universidade Federal do Maranhão, 2023. Citado 2 vezes nas páginas 17 e 20.

NASREEN, S.; AZAM, M. A.; SHEHZAD, K.; NAEEM, U.; GHAZANFAR, M. A. Frequent pattern mining algorithms for finding associated frequent patterns for data streams: a survey. *Procedia Computer Science*, Elsevier, v. 37, p. 109–116, 2014. Citado na página 17.

SALLOUM, S.; DAUTOV, R.; CHEN, X.; AL. et. Big data analytics on apache spark. *Int J Data Sci Anal*, v. 1, p. 145–164, 2016. Citado 2 vezes nas páginas 21 e 22.

SCHUBERT, E.; KOOS, A.; EMRICH, T.; ZÜFLE, A.; SCHMID, K. A.; ZIMEK, A. A framework for clustering uncertain data. In: *Proceedings of the VLDB Endowment*. [s.n.], 2015. v. 8, n. 12, p. 1976–1979. Disponível em: <<https://elki-project.github.io/>>. Citado na página 20.

SCHUBERT, E.; SANDER, J.; ESTER, M.; KRIEGEL, H. P.; XU, X. Dbscan revisited, revisited: Why and how you should (still) use dbscan. *ACM Trans. Database Syst.*, Association for Computing Machinery, New York, NY, USA, v. 42, n. 3, jul. 2017. ISSN

0362-5915. Disponível em: <<https://doi.org/10.1145/3068335>>. Citado 2 vezes nas páginas 19 e 20.

TOROUS, J.; KIANG, M. V.; LORME, J.; ONNELA, J.-P. et al. New tools for new research in psychiatry: a scalable and customizable platform to empower data driven smartphone research. *JMIR mental health*, JMIR Publications Inc., Toronto, Canada, v. 3, n. 2, p. e5165, 2016. Citado na página 15.