

**Universidade Federal do ABC**  
**Bacharelado em Ciência e Computação**



**Sistema de mapeamento e localização *indoor* baseado em Wi-Fi  
fingerprinting e redes neurais artificiais**

Anderson Lima de Araújo

**Santo André – SP**  
**2026**



**MINISTÉRIO DA EDUCAÇÃO**  
**Fundação Universidade Federal do ABC**  
**Centro de matemática, computação e cognição**  
Divisão acadêmica  
Av. dos Estados, 5001 - Bairro Santa Teresinha - Santo André, SP  
CEP 09210-580 - Fone: (11) 4996.7950/52  
pgc.cmcc@ufabc.edu.br

## Ata nº 9/2026/CMCC

Na data de 22 de Abril de 2026, às 14h30, online: <https://conferenciaweb.rnp.br/sala/hugo-81> da Universidade Federal do ABC em Santo André foi realizada a defesa de trabalho intitulado

### **Sistema de mapeamento e localização indoor baseado em Wi-Fi fingerprinting e redes neurais artificiais**

\_\_\_\_\_, para conclusão do Projeto de Graduação em Computação III do bacharelado em Ciência da Computação do discente Anderson Lima de Araújo, RA 11028713. Esteve presente avaliando o trabalho a banca examinadora composta pelos professores: Hugo Puertas de Araújo (UFABC), Mario Alexandre Gazziro (UFABC), Francisco de Assis Zampirolli (UFABC). Após apresentação do trabalho pelo discente e arguição da banca examinadora, seu trabalho foi considerado:

Aprovado com conceito A .

Aprovado com conceito \_\_\_\_\_, sob a condição de apresentação de revisões sugeridas pela banca até o dia \_\_\_\_/\_\_\_\_/\_\_\_\_.

Reprovado

Santo André, 22 de Abril de 2026

Documento assinado digitalmente  
**gov.br** HUGO PUERTAS DE ARAUJO  
Data: 23/04/2026 11:34:11-0300  
Verifique em <https://validar.iti.gov.br>

Documento assinado digitalmente  
**gov.br** MARIO ALEXANDRE GAZZIRO  
Data: 22/04/2026 15:45:21-0300  
Verifique em <https://validar.iti.gov.br>

Hugo Puertas de Araújo  
orientador(a)

Documento assinado digitalmente  
**gov.br** FRANCISCO DE ASSIS ZAMPIROLLI  
Data: 23/04/2026 07:29:12-0300  
Verifique em <https://validar.iti.gov.br>

Mario Alexandre Gazziro  
examinador(a) titular

Francisco de Assis Zampirolli  
examinador(a) titular

Anderson Lima de Araújo

**Sistema de mapeamento e localização *indoor* baseado em Wi-Fi  
fingerprinting e redes neurais artificiais**

Projeto de Graduação em Computação III  
(PGC III) apresentado ao Bacharelado em  
Ciência da Computação, como parte dos  
requisitos necessários para a obtenção do  
Título de Bacharel em Ciência da  
Computação.

Universidade Federal do ABC – UFABC  
Centro de Matemática, Computação e Cognição  
Bacharelado em Ciência da Computação

Orientador: Prof. Dr. Hugo Puertas de Araújo

**Santo André – SP  
2026**

## Resumo

Para sistemas de localização *indoor*, soluções tradicionais como o *Global Positioning System* (GPS) não apresentam desempenho satisfatório. Uma solução que se destaca é a utilização dos sinais de Wi-Fi para estimar a localização. Neste trabalho foi desenvolvido um sistema completo, desde o banco de dados até o aplicativo para *smartphone*, para coleta de sinais de Wi-Fi e para predição de localização no piso térreo da UFABC *campus* Santo André utilizando o método de impressão digital, consistindo de uma etapa de classificação para estimar a torre e de uma etapa de regressão para estimar as coordenadas. Como algoritmo de aprendizado de máquina para predição, foi utilizado um autocodificador variacional (VAE – variational autoencoder), conjuntamente com um perceptron multicamadas (*Multilayer Perceptron* – MLP), e os resultados obtidos são comparados com outros algoritmos pertinentes encontrados na literatura, avaliados com e sem Análise dos Componentes principais (*Principal Component Analysis* – PCA). A precisão do sistema mostrou-se satisfatória e o VAE apresentou bom desempenho para regressão, melhorando o desempenho do MLP, embora com desempenho próximo ao do MLP combinado com PCA. Para o problema de classificação, o VAE não apresentou melhoria de desempenho para o sistema.

**Palavras-chave:** Localização *Indoor*, Autocodificador Variacional, Aprendizado de Máquina, Rede Perceptron Multicamadas, Wi-Fi.

## **Abstract**

*For indoor localization systems, traditional solutions such as the Global Positioning System (GPS) do not provide satisfactory performance. A prominent alternative is the use of Wi-Fi signals to estimate location. In this work, a complete system was developed, from the database to the smartphone application, to be used for location prediction on the ground floor of the UFABC Santo André campus using the fingerprinting method, consisting of a classification stage to estimate the tower and a regression stage to estimate coordinates. As machine learning algorithms for prediction, a variational autoencoder (VAE) was used in conjunction with a multilayer perceptron (MLP), and the obtained results are compared with other relevant algorithms from the literature, evaluated with and without Principal Component Analysis (PCA). The system's accuracy proved to be satisfactory, and the VAE demonstrated good performance for regression, improving the performance of the MLP, although with performance close to MLP combined with PCA. For the classification problem, the VAE did not show performance improvement for the system.*

**Keywords:** *Indoor Localization, Variational Autoencoder, Machine Learning, Multilayer Perceptron, Wi-Fi.*

## Sumário

<b>1.Introdução</b> .....	1
<b>2.Trabalhos relacionados</b> .....	2
<b>3. Referencial Teórico</b> .....	3
3.1 Wi-Fi .....	3
3.2 Técnicas de Localização .....	3
3.3 Métodos de Localização .....	5
3.4 Aprendizado de Máquina .....	7
3.5 K Vizinhos Mais Próximos .....	7
3.6 Floresta Aleatória .....	8
3.7 Redes Neurais Artificiais .....	8
3.8 Análise de Componentes Principais .....	9
3.9 Autocodificador Variacional .....	10
<b>4. Materiais e Métodos</b> .....	13
<b>5. Resultados e Discussão</b> .....	18
<b>6. Conclusão e Trabalhos Futuros</b> .....	21
<b>Referências Bibliográficas</b> .....	22

## 1. Introdução

Quando nos referimos a localização, a tecnologia que mais se popularizou foi a do GPS, revolucionando essa área e sendo usada nos mais diversos contextos, como localização de pessoas e de veículos, jogos, entre outras (Mittelstadt e Pretto, 2020). Porém, o GPS não possui precisão adequada em lugares subterrâneos e fechados (*indoor*) devido à atenuação e reflexão dos sinais (Rodrigues e Carvalho, 2019).

Nesse cenário, os sistemas de localização *indoor* desempenham um papel importante e, dentre as tecnologias de transmissão sem fio, se destacam o padrão IEEE802.11 (popularmente conhecida como Wi-Fi) e o *Bluetooth Low Energy* (BLE), que inclusive são comumente encontradas em *smartphones* (Castro, Silva e Sampaio, 2018). As técnicas baseadas nas tecnologias sem fio mais empregadas são a *Time of Arrival* (ToA), *Angle of Arrival* (AoA) e *Received Signal Strength Indication* (RSSI), que utilizam o tempo de chegada, o ângulo de chegada, ambos em relação a uma estação base, e a intensidade do sinal, respectivamente (Bellecieri, Jabour e Jabour, 2016). Pode-se destacar a utilização de aprendizado de máquina como uma tendência notável, que consiste na utilização de algoritmos sobre uma base de dados para reconhecer padrões e estimar a localização (Oliveira, 2017).

O objetivo deste trabalho consiste em desenvolver um sistema completo de localização *indoor* baseado em Wi-Fi, como ilustrado na Figura 1, que engloba um aplicativo móvel para coleta e mapeamento dos sinais (RSSI), a criação e gerenciamento de um banco de dados, o desenvolvimento de um modelo de redes neurais com métodos de redução de dimensionalidade treinado para realizar inferências de posição e comparativo de resultados com outros algoritmos de aprendizado de máquina. Também haverá uma funcionalidade destinada ao usuário final para estimar e visualizar sua localização.

Este trabalho foi dividido em seção 2, que apresenta alguns trabalhos relacionados com o tema; referencial teórico na seção 3; na seção 4 são apresentados os materiais e métodos utilizados; na seção 5 são apresentados os resultados obtidos; por fim, na seção 6, é apresentada a conclusão e trabalhos futuros.

Figura 1: Representação esquemática do sistema de localização *indoor*



Fonte: Elaborado pelo autor com auxílio de ferramenta de IA generativa (OpenAI, 2026)

## 2. Trabalhos Relacionados

Existem, na literatura, diversos trabalhos de pesquisa abordando localização *indoor*. Em Bellecieri, Jabour e Jabour (2016), a localização *indoor* é realizada através da leitura bidirecional do RSSI, usando a média dos valores dos dispositivos envolvidos, comparando-se também com o desempenho das leituras unidirecionais. O experimento se limita a utilizar apenas 2 dispositivos, um fixo e outro móvel, que interagem através de uma rede ad-hoc. A leitura bidirecional se mostrou mais precisa. Em Castro, Silva e Sampaio (2018), *smartphones* são utilizados para localização *indoor*, onde são combinados os sinais de Wi-Fi, de BLE e de sensores inerciais através do filtro de Kalman para se obter uma melhor precisão. Os experimentos ocorreram utilizando apenas um *smartphone* e em um ambiente limitado de 300 m<sup>2</sup>, com 2 estações de Wi-Fi e 4 estações de BLE e, apesar de não possuir necessidade de calibração, esse método apresentou precisão inferior em alguns casos. Em Issa *et al.* (2025), com o uso de impressão digital, foi estimada a localização *indoor* através do algoritmo K Vizinhos Mais Próximos (K-Nearest Neighbors — KNN). Para o experimento foi utilizado um ambiente controlado, no qual as medições foram espacialmente distribuídas de maneira homogênea. Foram testados diversos valores de K e previsões precisas foram alcançadas, mostrando que o KNN possui desempenho satisfatório para esse caso de uso, mesmo com sua simplicidade. Contudo, no artigo não são testados outros algoritmos.

Em Gorjan e Jiménez (2024), a localização *indoor* é estimada através de dados de sinais de Wi-Fi, disponibilizados de forma pública na *Internet*, em um conjunto de edifícios universitários, estimando-se, além das coordenadas X e Y, o edifício e o andar em questão, dividindo a predição em etapas, de classificação e regressão. Para essa tarefa o desempenho de diversos algoritmos é avaliado, inclusive, após o uso de uma Busca em Grade, que realiza uma busca completa sobre um subconjunto de hiperparâmetros, em um intervalo especificado, no algoritmo com dados de treinamento para se obter os melhores daquele intervalo (Liashchynskiy e Liashchynskiy, 2019). Nosso trabalho se diferencia deste por avaliar também a redução de dimensionalidade em redes neurais e por ter desenvolvido o sistema de coleta de pontos de Wi-Fi.

### 3. Referencial Teórico

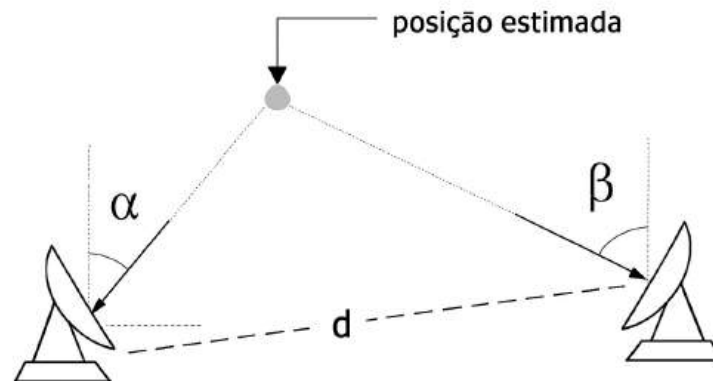
#### 3.1 Wi-Fi

O *Wireless Fidelity*, conhecido como Wi-Fi, refere-se a redes que estão na especificação IEEE 802.11. Lançado em 1997, esse padrão não exige licença para operação por operar em frequências de ISM (*Industrial, Scientific and Medical*) (Mittelstadt e Pretto, 2020). A transmissão de dados é especificada na camada física, sendo que a principal função nessa camada está relacionada com modulação do sinal, condicionamento da informação a ser enviada e o espalhamento no espectro (Castro, 2018). As redes sem fio, apesar de serem utilizadas para fins externos, têm sido muito úteis em outras aplicações, como sistemas de localização e cálculo de distância, que exploram propriedades do meio para realizar essas estimativas. Nesse contexto, destacam-se técnicas como Time of Arrival (ToA), Angle of Arrival (AoA) e Received Signal Strength Indicator (RSSI), que são detalhadas a seguir (Bellecieri, Jabour e Jabour, 2016).

#### 3.2 Técnicas de Localização

No método *Angle of Arrival* (AoA) é obtido o ângulo do sinal que chega aos receptores, exigindo-se para isso receptores e antenas especiais. Para inferir a localização do dispositivo que emite o sinal, são necessárias pelo menos duas antenas, conforme ilustrado na Figura 2. Esse método é vulnerável ao fenômeno de espalhamento devido à interferência que o sistema de localização pode sofrer de outros sinais (Bellecieri, Jabour e Jabour, 2016).

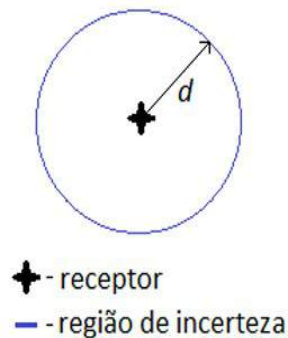
Figura 2: Localização de um dispositivo através da técnica AoA



Fonte: (Bellecieri; Jabour; Jabour, 2016)

O *Time of Arrival* (ToA) indica o tempo que o sinal leva para chegar ao receptor e, baseado no fato de que o sinal viaja na velocidade da luz, obtêm-se a distância entre os dispositivos. Para que haja uma precisão na estimativa é fundamental que os relógios dos dispositivos estejam sincronizados (Oliveira, 2017). Através de uma fonte de sinal, obtém-se uma região de incerteza circular conforme a Figura 3.

Figura 3: Região de incerteza obtido pela técnica ToA



Fonte: (Oliveira, 2017)

A sincronização entre nós é difícil de alcançar na prática e, como alternativa, há o Time Difference of Arrival (TDOA), que é obtido a partir da diferença entre o tempo de chegada entre dois pontos distintos. O TDOA, portanto, não exige sincronização de tempo, mas precisa de computação adicional (Ohashi, 2023).

O *Received Signal Strength Indicator* (RSSI) é um indicativo da potência do sinal recebido. Como o sinal perde potência proporcionalmente à distância entre o receptor e o transmissor, torna-se possível estimar a distância entre eles (Oliveira, 2017). É medido em decibéis (dBm) em uma escala negativa, ou seja, quanto mais próximo o valor for de zero maior será a intensidade do sinal (Bellecieri, Jabour e Jabour, 2016) e, em um cenário ideal, pode ser descrito conforme a Equação 1. Como resultado, obtemos uma região circular de incerteza da mesma forma como ilustrada na Figura 3, do ToA (Oliveira, 2017).

$$P(d) = P_o - 10 * n * \log(d/d_o) \quad (1)$$

$P(d)$ : potência na distância  $d$

$n$ : fator de perda

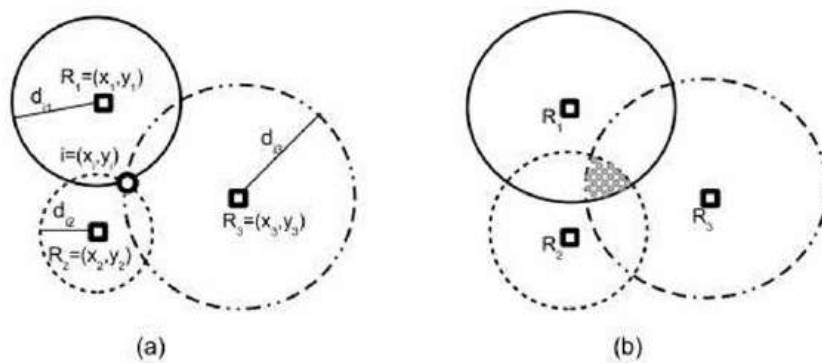
$P_o$ : potência recebida a uma distância de referência  $d_o$

O RSSI pode sofrer interferência de diversas maneiras, como a sensibilidade do receptor, influência de outros sinais no meio, a atenuação no meio, entre outros (Bellecieri, Jabour e Jabour, 2016).

### 3.3 Métodos de Localização

Como métodos de localização podemos citar a triangulação que, através do uso de senos e cossenos, calcula a localização com base nos ângulos dos sinais que chegam à base, utilizando portanto a técnica AoA. Outra técnica seria a trilateração, que determina a localização do dispositivo ao obter a informação através de qualquer técnica de, pelo menos, três nós que possuem localização previamente conhecida. A interseção da área de cobertura dos nós representa a localização do dispositivo e a Figura 4 ilustra a diferença entre uma situação ideal e hipotética e outra real, onde a localização é representada por uma área em vez de um ponto (Bellecieri, Jabour e Jabour, 2016). As técnicas baseadas em alcance mais populares para serem usadas neste caso são o AoA, o TDOA, o TOA e o RSSI (Ohashi, 2023).

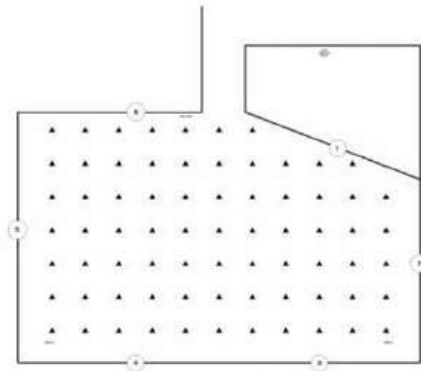
Figura 4: Trilateração, no qual a) representa uma situação ideal e b) uma situação real



Fonte: (Bellecieri; Jabour; Jabour, 2016)

Além desses métodos, há também a impressão digital, ou *fingerprinting*, no qual são colhidas medições de RSSI de várias fontes para vários pontos em uma planta, como ilustrado na Figura 5, e os valores são armazenados em um banco de dados. Sendo assim, quando se deseja saber a localização de um dispositivo, são coletados os valores de RSSI das estações naquela posição e um algoritmo comparará com os valores armazenados para indicar a localização (Bellecieri, Jabour e Jabour, 2016).

Figura 5: Planta que possui pontos de referência representados por triângulo e pontos de acesso representados por círculos



Fonte: (Bellecieri; Jabour; Jabour, 2016)

Por utilizar dados reais que refletem a natureza do local, o *fingerprinting* apresenta precisão elevada, se comparada com a técnica de triangulação (Lee, 2019), e é uma das técnicas mais utilizadas devido ao seu baixo custo e compatibilidade com infraestrutura já existente (Issa *et al.*, 2025), pois o Wi-Fi já é empregado massivamente tanto em áreas residenciais quanto em escritórios e distritos comerciais (Wang *et al.*, 2018). Contudo, o *fingerprinting* exige calibração e é sensível a mudanças no ambiente (Issa *et al.*, 2025).

### 3.4 Aprendizado de Máquina

O aprendizado de máquina, uma subárea da inteligência artificial conforme divisão feita pela *Association for the Advancement of Artificial Intelligence*, considerada referência na área (IPEA, 2020), consiste no reconhecimento de padrões através de algoritmos. Isso pode ocorrer de maneira supervisionada, que é quando o conjunto de dados consiste em entradas com saídas conhecidas, ou seja, o algoritmo possui um gabarito do resultado esperado no momento de treinamento. O reconhecimento de padrões também pode ocorrer de forma não supervisionada, onde são realizadas inferências apenas com os dados de entrada. Também é possível dividir o aprendizado supervisionado em classificação ou regressão, que é quando a variável de saída é uma categoria no primeiro caso e, no segundo caso, a variável de saída é um valor numérico contínuo (Oliveira, 2017).

### 3.5 K Vizinhos Mais Próximos

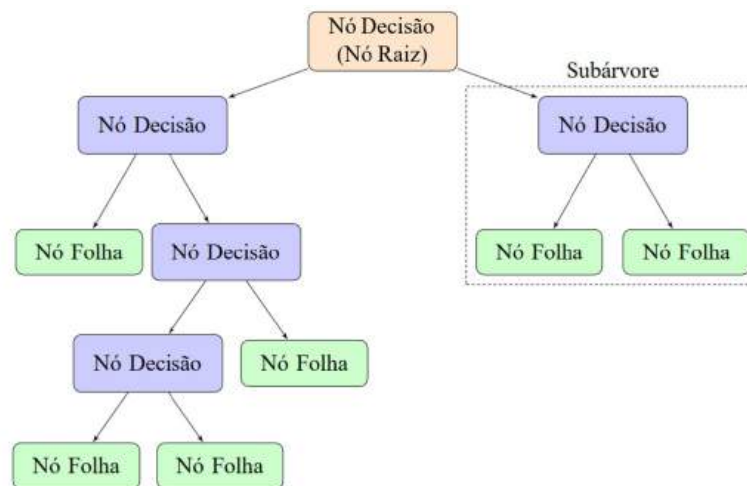
O KNN é um algoritmo de aprendizado de máquina supervisionado, usado principalmente para problemas de classificação, mas podendo ser usado também para regressão. O KNN pode classificar um objeto sem rótulo com base na contagem dos rótulos dos K objetos vizinhos do conjunto de treinamento mais próximos dele (Uddin, 2022). Essa distância entre os objetos pode ser estimada a partir de várias métricas, como a Distância Euclidiana ou a Distância de Manhattan (Semaan *et al.*, 2019). Para problemas de regressão, o KNN pode utilizar a média dos K objetos vizinhos para estimar o valor de dados novos (Pinecone, s.d.).

É um algoritmo que se destaca pela sua simplicidade por conta de sua arquitetura fácil de entender e que permite alta adaptabilidade (Uddin, 2022), porém, também possui desvantagens como a dependência da qualidade do conjunto de treinamento, uma vez que ruídos ou instâncias irrelevantes podem aumentar consideravelmente a taxa de erros (Max, 2016).

### 3.6 Floresta Aleatória

A floresta aleatória (*Random Forest* – RF) é uma técnica de aprendizado supervisionado que pode ser utilizado tanto para classificação como para regressão. Sua estrutura é composta de nós terminais, ou folhas, e de nós de decisão, sendo conectados por ramificações conforme ilustra a Figura 6. Cada nó folha representa uma classe e, partindo do nó raiz (de decisão), se chega a ele ao final do trajeto após serem direcionados nos nós de decisão (Paiva *et al.*, 2024). Para tarefas de regressão, será obtida a média das predições das árvores de decisão individuais e, para classificação, a variável categórica mais frequente determinará a classe prevista (IBM, s.d.a).

Figura 6: Representação esquemática de uma floresta aleatória



Fonte: (Paiva *et al.*, 2024).

A RF é capaz de atingir boa capacidade de generalização sem precisar de poda, tornando-a mais eficiente. Também se destaca em acurácia preditiva quando comparada a outras técnicas de aprendizado supervisionado (Riqueti, Ribeiro e Zárte, 2018).

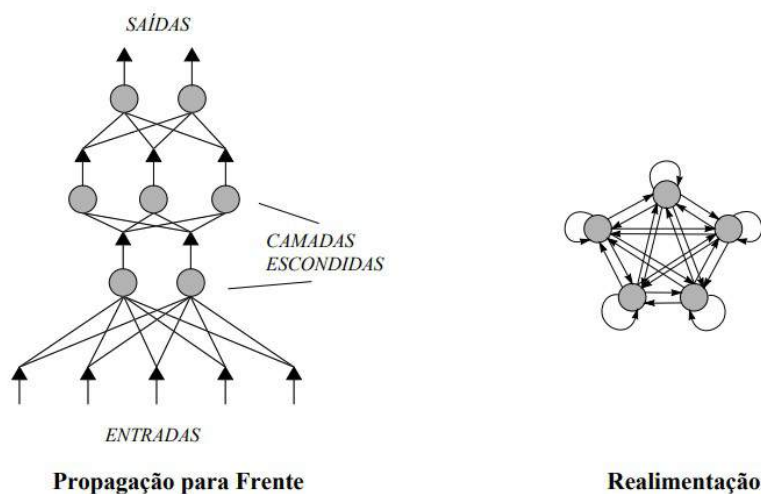
### 3.7 Redes Neurais Artificiais

Uma rede neural artificial (RNA) é um modelo composto de neurônios artificiais, sendo estes elementos de processamento simples que aplicam uma função de ativação aos dados, estando organizados em camadas e ligados entre si por conexões associadas a pesos. O conjunto de pesos é atualizado em um processo chamado treinamento e é a partir dele que são extraídas as características dos dados e armazenado o conhecimento na rede (Binoti, Binoti e Leite, 2014).

A propagação da informação pela rede pode ser unidirecional para frente (*feedforward*) ou com realimentação (*recurrent*), conforme Figura 7 (Rauber, 2005).

O MLP é a rede neural mais conhecida e frequentemente a mais utilizada (Popescu *et al*, 2009). Nela, os neurônios são organizados em camadas, sendo uma de entrada, uma de saída e uma ou mais camadas ocultas entre essas duas e a propagação de informação é do tipo *feedforward*. A maneira mais utilizada para a atualização dos pesos das conexões no treinamento é a retropropagação, que se baseia em técnicas de gradiente descendente (Ramchoun *et al.*, 2016).

Figura 7: Esquema das principais topologias de uma RNA



Fonte: (Rauber, 2005)

### 3.8 Análise de Componentes Principais

A PCA é uma técnica de estatística multivariada que combina todos os componentes em outros componentes linearmente independentes, onde cada um é uma combinação linear de todas as variáveis originais. O propósito da PCA é reduzir a quantidade de dados, perdendo o mínimo possível de informação e, apesar da dimensão do conjunto de variáveis ser mantida, as novas variáveis estão ordenadas segundo sua variância (Varella, 2008), permitindo derivar um número menor de componentes que representem tanta informação quanto as originais quanto possível (Sarkar, 2023).

Seja a matriz  $X$ , onde cada coluna representa uma característica observada e cada linha representa um indivíduo. Devido à complexidade de analisar a estrutura diretamente de  $X$ , pode-se utilizar a matriz de covariância  $S$  ou a matriz de correlação  $R$  para a análise.

$$X = \begin{bmatrix} X_{11} & X_{12} & \dots & X_{1p} \\ X_{21} & X_{22} & \dots & X_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ X_{n1} & X_{n2} & \dots & X_{np} \end{bmatrix} \quad S = \begin{bmatrix} \hat{V}ar(X_1) & \hat{C}ov(X_1 X_2) & \dots & \hat{C}ov(X_1 X_p) \\ \hat{C}ov(X_2 X_1) & \hat{V}ar(X_2) & \dots & \hat{C}ov(X_2 X_p) \\ \vdots & \vdots & \ddots & \vdots \\ \hat{C}ov(X_p X_1) & \hat{C}ov(X_p X_2) & \dots & \hat{V}ar(X_p) \end{bmatrix}$$

$$R = \begin{bmatrix} 1 & r(X_1 X_2) & \dots & r(X_1 X_p) \\ r(X_2 X_1) & 1 & \dots & r(X_2 X_p) \\ \vdots & \vdots & \ddots & \vdots \\ r(X_p X_1) & r(X_p X_2) & \dots & 1 \end{bmatrix}$$

Os componentes principais podem ser determinados através a Equação 2, para S ou R, onde as raízes  $\lambda_1, \lambda_2, \dots, \lambda_n$  são os autovetores, e para cada autovetor  $\lambda_i$ , existe um autovalor  $\tilde{a}_i$ .

$$\tilde{a}_i = \begin{bmatrix} a_{i1} \\ a_{i2} \\ \vdots \\ a_{ip} \end{bmatrix}$$

Então, temos que o i-ésimo componente principal é determinado conforme a Equação 3.

$$\det[R - \lambda I] = 0 \text{ ou } \det[S - \lambda I] = 0 \quad (2)$$

$$Y_i = a_{i1}X_1 + a_{i2}X_2 + \dots + X_{ip}X_p \quad (3)$$

Através da Equação 4, é possível determinar qual a contribuição  $C_i$ , em porcentagem, de cada componente principal  $Y_i$  (Varella, 2008).

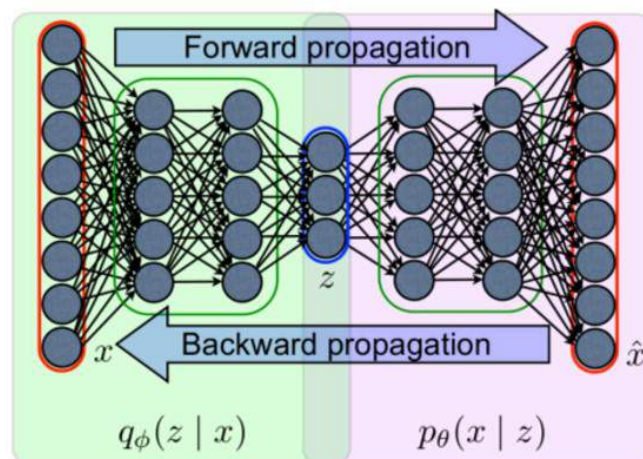
$$C_i = \frac{\hat{V}ar(Y_i)}{\sum_{i=1}^p \hat{V}ar(Y_i)} 100 = \frac{\lambda_i}{\sum_{i=1}^p \lambda_i} 100 = \frac{\lambda_i}{\text{traço}(S)} 100 \quad (4)$$

### 3.9 Autocodificador Variacional

Um autocodificador é uma RNA que codifica uma entrada para uma representação comprimida e expressiva e que decodifica para o espaço original da entrada, de forma que fique o mais similar possível (Bank, Koenigstein e Girytes, 2020). Durante seu treinamento não supervisionado o autocodificador aprende, portanto, quais variáveis latentes são mais importantes para reconstruir a entrada (IBM, s.d.b).

É chamado de VAE o autocodificador que reconstrói a entrada e também modela a distribuição dos dados na forma probabilística. O codificador possui uma função de densidade probabilística  $q_\phi(z|x)$ , com pesos representados por  $\phi$ , e o decodificador possui uma função de densidade probabilística representado por  $p_\theta(x|z)$ , com pesos  $\theta$ . As variáveis latentes  $z$  são extraídas da entrada  $x$  através de uma distribuição de probabilidade e a reconstrução  $\hat{X}$  é gerada a partir de  $z$ , conforme Figura 8.

Figura 8: Arquitetura de um VAE



Fonte: (Dias Júnior, 2024)

Caso a densidade de probabilidade para o codificador seja uma Gaussiana,  $q_\phi(z|x)$  e  $p_\theta(x|z)$  fornecem as matrizes com os parâmetros da distribuição do modelo, geralmente a média e a covariância, e os pesos  $\phi$  e  $\theta$  precisam ser aprendidos por suas respectivas redes de forma a configurar uma função de perda, no primeiro caso para gerar o espaço latente e no segundo caso para gerar o sinal reconstruído (Dias Júnior, 2024).

O objetivo de otimização do VAE é chamado de *Evidence Lower Bound* (ELBO), sendo representado pela Equação 5 (Kingma e Welling, 2019). A função de perda do VAE necessita de um termo a mais para forçar  $q_\phi(z|x_i)$ , para cada entrada  $x_i$ , a se aproximar de uma distribuição de densidade simples (geralmente a distribuição normal), sendo este termo a Divergência de Kullback-Leibler (Divergência KL), que mede a diferença entre duas distribuições de probabilidade, conforme a Equação 6, onde  $x_i$  são as entradas individuais e  $p(z)$  é uma distribuição fixa para qualquer entrada. Sem o termo, a saída pode ficar desorganizada e também pode haver memorização (Dias Júnior, 2024). Caso  $q_\phi(z|x_i)$  e  $p_\theta(z)$  sejam gaussianas, a Divergência KL pode ser expressa de forma simplificada como na Equação 7, com gaussiana de variância  $\sigma$  e média  $\mu$  (Kingma e Welling, 2022).

$$l_{\theta, \phi}(\mathbf{x}) = E_{q_{\phi}(z|\mathbf{x})}[\log p_{\theta}(\mathbf{x}|z) - \log q_{\phi}(z|\mathbf{x})] \quad (5)$$

$$D_{kl}(q_{\phi}(z|x_i)||p_{\theta}(z)) = \sum_i ((q_{\phi}(z|x_i) \cdot \log(\frac{q_{\phi}(z|x_i)}{p_{\theta}(z)})) \quad (6)$$

$$-D_{kl}(q_{\phi}(z)||p_{\theta}(z)) = \frac{1}{2} \sum_1^j (1 + \log(\sigma_j^2) - \mu_j^2 - \sigma_j^2) \quad (7)$$

O critério para modelos de probabilidade mais comum é a maximização da log-verossimilhança dos dados que, no contexto de VAEs, é equivalente a minimização da Divergência KL entre as distribuições dos dados e do modelo. Como mostrado na Equação 8 é possível, maximizando a ELBO, tornar o modelo melhor pela maximização da probabilidade marginal  $p_{\theta}(\mathbf{x})$  e pela minimização da Divergência KL da aproximação  $q_{\phi}(z|\mathbf{x})$  e do verdadeiro posterior  $p_{\theta}(z|\mathbf{x})$ , tornando o primeiro termo melhor (Kingma e Welling, 2019).

$$l_{\theta, \phi}(\mathbf{x}) = \log p_{\theta}(\mathbf{x}) - D_{kl}(q_{\phi}(z|\mathbf{x})||p_{\theta}(z|\mathbf{x})) \quad (8)$$

Uma extensão do VAE original é conhecida como  $\beta$ -VAE, proposto por Higgins et al. (2017), no qual a Divergência KL é multiplicada pelo fator  $\beta$  que, em valores elevados, proporciona mais organização na aprendizagem de representações latentes, isto é, melhor generalização e, no caso de valores reduzidos, incentivam a fidelidade de reconstrução, ou seja, reconstruções mais próximas do real. No caso do  $\beta$ -VAE, a ELBO fica representada como na equação 9 (Higgins et al., 2017).

$$l(\theta, \phi; \mathbf{x}, z, \beta) = E_{q_{\phi}(z|\mathbf{x})}[\log p_{\theta}(\mathbf{x}|z)] - \beta D_{kl}(q_{\phi}(z|\mathbf{x})||p_{\theta}(z)) \quad (9)$$

No VAE, embora a propagação do erro seja simples, a retropropagação do erro possui uma série de desafios, pois ela não é diretamente diferenciável. Uma técnica possível para lidar com problema é chamada de truque de reparametrização que, com o auxílio de uma variável não determinística, assume que a distribuição pode ser redefinida como uma função diferenciável. Para VAEs clássicos, onde  $z \sim q_{\phi}(z|\mathbf{x})$  segue uma distribuição normal  $N(\mu, \sigma^2)$ , a função de reparametrização é  $z = \mu + \sigma \cdot \epsilon$ , onde  $\epsilon$  é uma função auxiliar de ruído amostrada de uma distribuição normal padrão. De modo geral, são possíveis 3 abordagens para lidar com a reparametrização, podendo ser por uma função de distribuição acumulada inversa, no qual  $\epsilon$ , amostrada de uma distribuição uniforme no intervalo  $(0,1)$ , é transformada pela inversa da função de distribuição acumulada da distribuição desejada. Esse tipo de reparametrização pode ser utilizada com Exponencial, Logística e Pareto, por exemplo.

Outra forma de reparametrização é por composição, no qual as variáveis aleatórias são expressas como transformações aplicadas a variáveis auxiliares como, por exemplo, a Normal-Logística e a Gamma.

A reparametrização também é possível por uma família de distribuições do tipo localização-escala, onde há um parâmetro para a localização, como a média, e outro parâmetro para a escala, como o desvio padrão. Nesta abordagem,  $\epsilon$  vem da distribuição padrão e quando a localização vale 0 e a escala vale 1, pode-se reparametrizar na forma padrão. Algumas distribuições que permitem essa reparametrização incluem distribuição Gaussiana, de Laplace e T de Student (Silveira, 2018).

#### 4. Materiais e Métodos

Neste trabalho foi desenvolvido um sistema *Full Stack*, isto é, um sistema completo composto do lado do cliente (*front-end*) e do lado do servidor (*back-end*) de aplicações *web*, que engloba desde a interface do usuário até o manuseio do banco de dados (Geeksforgeeks, 2025). Ferramentas de inteligência artificial generativa foram utilizadas como apoio na elaboração inicial de alguns trechos de códigos, sendo a validação, integração e implementação realizadas pelo autor.

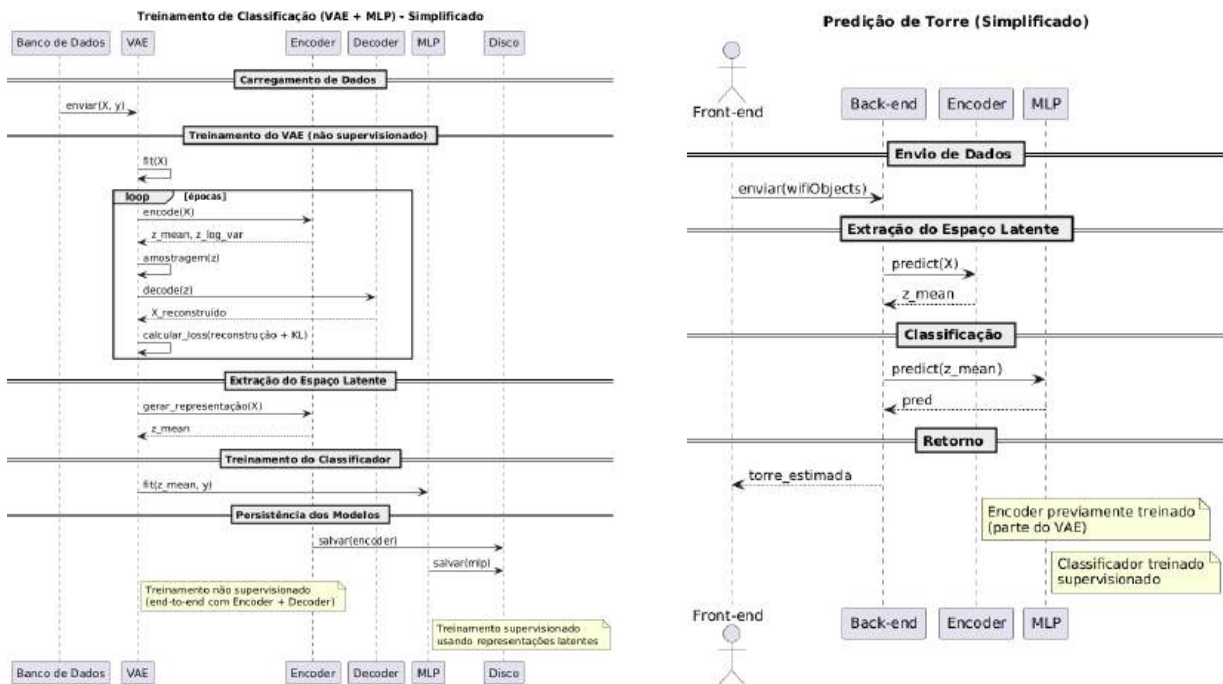
Na parte do *front-end*, foi utilizado o *framework* React-Native para desenvolver um aplicativo móvel, que será responsável por popular o banco de dados com os valores de varredura dos sinais de Wi-Fi recebidos pelo dispositivo, sendo composto pelos valores de RSSI, frequência, *Media Access Control* (MAC) e o *Basic Service Set Identifier* (BSSI), entre outros, e as coordenadas espaciais do local obtidas através da visualização e apontamento na planta, além da torre em que se encontra no momento, conforme exemplo ilustrativo da Figura 9-a). As varreduras foram realizadas com a biblioteca *react-native-wifi-reborn*. Esses dados serão utilizados para o treinamento supervisionado dos modelos de aprendizado de máquina e, para cada ponto de extração dos sinais de Wi-Fi, serão extraídas 4 medidas com intervalo curto de tempo entre elas. Para caso de pontos de acesso reincidentes por varredura, é tirada a média de seus sinais. O aplicativo também permite a estimativa da localização no momento através do envio dos dados de varredura do Wi-Fi naquele instante, tendo como resultado a torre e as coordenadas em que se encontra no momento, conforme ilustrado em 9-b) e em 9-c). Neste caso é extraída a média dos sinais de Wi-Fi das 4 varreduras para um mesmo ponto de acesso.

Figura 9: Interface gráfica do aplicativo móvel



Fonte: arquivos do autor

O *back-end* foi desenvolvido utilizando a linguagem de programação Python e o *framework* FastAPI, onde foi implantado em um *Droplet* da Digital Ocean, ou seja, em um servidor de sistema operacional Linux que se encontra na nuvem. O banco de dados utilizado é o PostgreSQL, também implantado no *Droplet*. Na API, um VAE foi implementado para ser utilizado tanto na classificação da torre quanto na estimativa das coordenadas, sendo que cada uma destas etapas possui seu próprio MLP. A lógica de implementação do treinamento de classificação de torre e a lógica de implementação da estimativa de localização do *back-end* do sistema podem ser observadas, com algumas simplificações, nos diagramas de sequência da *Unified Modeling Language* (UML) apresentados na Figura 10-a) e 10-b), respectivamente.

Figura 10: Diagrama de seqüência UML do *back-end* do sistema de *localização indoor*

a) treinamento

b) predição

Fonte: Arquivos do autor

O experimento foi realizado no campus de Santo André da Fundação Universidade Federal do ABC (UFABC) que é composto de 9 blocos, conforme a Figura 11. Foram extraídas medidas de Wi-Fi de pontos localizados nas três torres do bloco A, no piso térreo, representados pelos pontos de 7 a 9 da Figura 11. Foi levado em conta que a quantidade de pontos coletados não pode ser muito baixa, o que reduz a acurácia da localização, e nem muito alta, o que demandaria muito trabalho para coleta (Gao, 2015). Porém, foi coletada uma maior quantidade de pontos para a torre 3 e uma menor quantidade para a torre 1, conforme Tabela 1, que também mostra a distância média entre os pontos coletados e seus vizinhos mais próximos, para fins comparativos.

Tabela 1: Métricas dos pontos coletados das torres do térreo da UFABC

Torre	Quantidade de pontos	Média das distâncias entre o ponto e seu vizinho mais próximo (m)
1	233	2,88
2	244	1,68
3	301	1,67

Fonte: Elaborado pelo autor

A estimativa de posição foi dividida em duas etapas onde, na primeira etapa, utiliza-se um classificador para determinar a torre em que o ponto se encontra e, na segunda, aplica-se um modelo de regressão para estimar as coordenadas bidimensionais. Na etapa de regressão, cada torre foi considerada individualmente no treinamento e na avaliação dos algoritmos. Em ambas as etapas, foram testados o KNN, RF e o MLP, e estes algoritmos tiveram os hiperparâmetros ajustados através de uma Busca em Grade, com e sem PCA. Esses resultados foram comparados com o modelo de MLP baseado em VAE, que utiliza a família de distribuições do tipo localização-escala.

A avaliação do desempenho dos modelos foi realizada através da plataforma Google Colab, um ambiente de computação em nuvem semelhante aos cadernos Jupyter, e a separação entre treinamento e teste foi realizada de forma que não houvesse uma mesma coordenada nos dois conjuntos de dados.

Figura 11: *Campus* de Santo André da Universidade Federal do ABC



Fonte: (UFABC, s.d.)

Na etapa de classificação, são utilizadas como métricas de desempenho a precisão, que é a taxa de predições positivas corretas em relação ao total de positivas estimadas, como mostrado na Equação 10; a acurácia, que informa a taxa de observações corretas relativas ao total de observações, sendo calculada conforme Equação 11; o *recall*, que é a taxa de verdadeiros positivos em relação aos todos os positivos, conforme Equação 12; e o F1-Score, que é uma média harmônica entre precisão e *recall*, como mostrado na Equação 13 (Gorjan e Jiménez, 2024).

$$\text{Precisão} = \frac{TP}{TP + FP} \quad (10)$$

$$\text{Acurácia} = \frac{TP + TN}{TP + TN + FP + FN} \quad (11)$$

$$\text{Recall} = \frac{TP}{TP + FN} \quad (12)$$

$$\text{F1-Score} = 2 * \frac{\text{Precisão} * \text{Recall}}{\text{Precisão} + \text{Recall}} \quad (13)$$

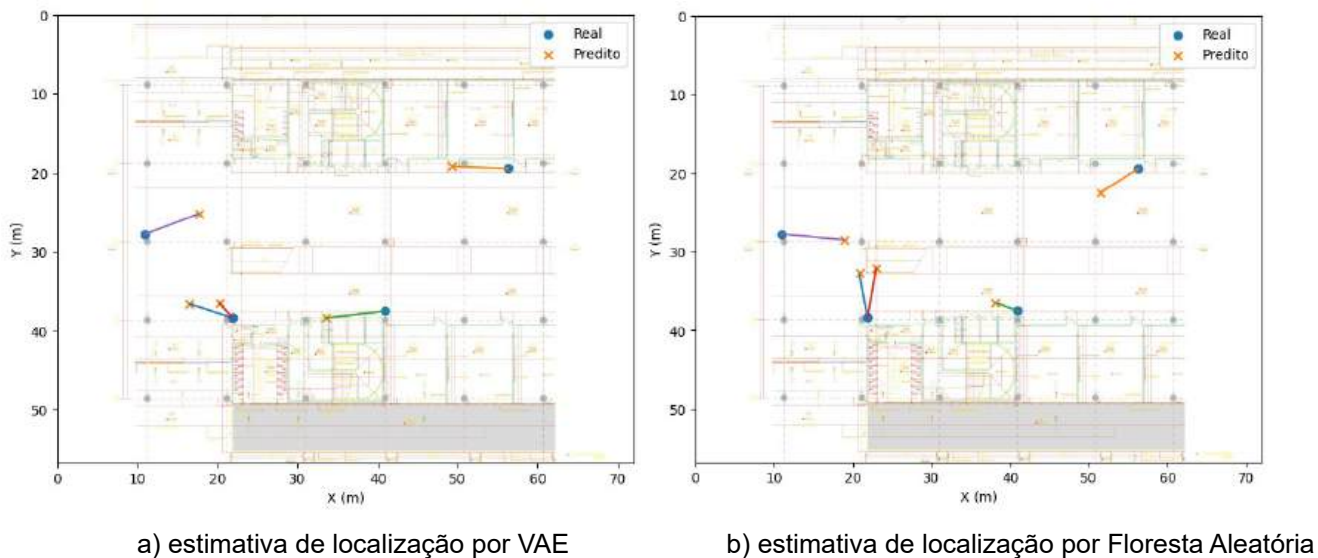
Também é utilizada a matriz de confusão para avaliar os modelos na etapa de classificação, que é uma tabela que compara os valores reais com os valores preditos de um modelo. Na matriz de confusão, cada coluna representa a quantidade de instâncias previstas de uma classe e cada linha representa as instâncias reais dessa mesma uma classe (IBM, s.d.c).

Foi utilizado como métrica de avaliação de desempenho dos modelos de aprendizado de máquina na etapa de regressão o Erro Absoluto Médio (EAM), obtido conforme a Equação 14, onde  $E_i$  são os valores estimados,  $O_i$  os valores observados e N é o número de observações (Rubio, 2018):

$$\text{EAM}(\%) = \frac{\sum_{i=1}^n |E_i - O_i|}{N} \quad (14)$$

A Figura 12 apresenta 5 pontos de coleta de sinais de Wi-Fi da torre 2 do térreo da UFABC, sendo 2 pontos repetidos, e suas respectivas posições estimadas por meio da RF e do VAE.

Figura 12: Pontos de coleta de Wi-Fi na torre 3 do térreo da UFABC e suas respectivas localizações estimadas



Fonte: Arquivos do autor

## 5. Resultados e Discussão

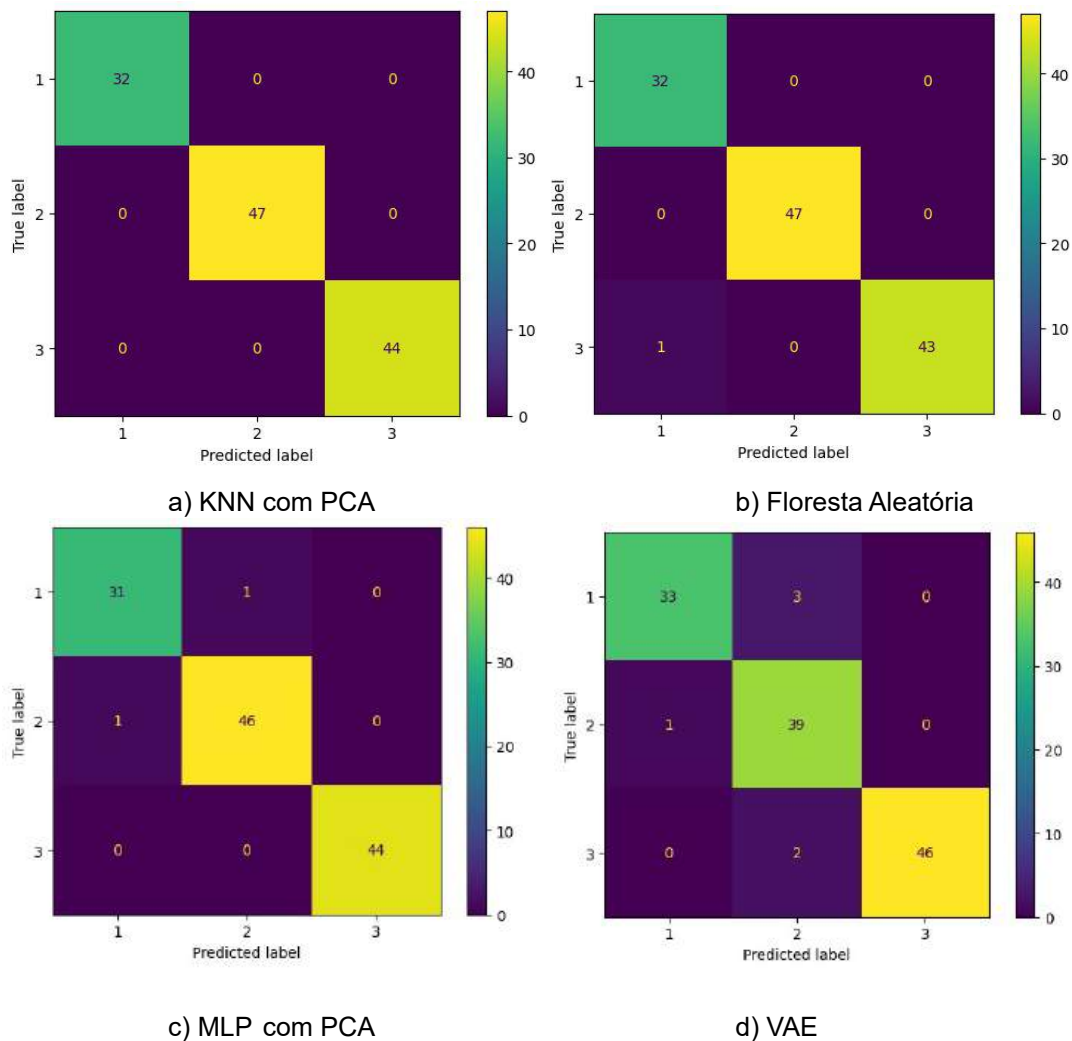
Após a coleta de dados, treinamento e teste dos 4 algoritmos de aprendizado de máquina, cada um com os melhores hiperparâmetros obtidos via Busca em Grade, com e sem PCA, foi obtido na etapa de classificação o desempenho das métricas acurácia, precisão, *F1-Score* e *Recall* conforme Tabela 2. A matriz de confusão mostrada na Figura 13 foi obtida dos algoritmos com o melhor caso possível, com PCA se este for o caso. É possível observar que o KNN obteve o melhor desempenho, sem cometer nenhum erro, com uso de PCA. Em seguida, com desempenho próximo ao do KNN, temos a RF sem PCA e, em terceiro lugar, o MLP com PCA. No caso do RF, a utilização de PCA resultou em uma piora no desempenho do modelo. Dada a contiguidade do piso térreo da UFABC, já se esperava a ocorrência de erros na estimativa entre torres próximas. Na Figura 13, observa-se a predominância desse comportamento, com 8 erros entre torres adjacentes e apenas 1 erro envolvendo torres mais distantes nas matrizes de confusão. Dentre os quatro algoritmos o VAE apresentou um desempenho inferior aos demais, indicando que a redução de dimensionalidade de forma não-linear não contribuiu significativamente para a melhoria da classificação. No VAE os valores de  $\beta$  que trouxeram melhor desempenho foram próximos ao padrão de 1 ou valores reduzidos, o que indica que, para este problema, priorizar a reconstrução foi mais eficaz do que impor uma forte regularização no espaço latente.

Tabela 2: Métricas de desempenho obtidas na estimativa das torres do térreo na UFABC

Métrica	KNN	RF	MLP	KNN + RF + PCA	MLP + PCA	VAE
Acurácia	0,984	0,992	0,967	1,000	0,959	0,952
Precisão	0,986	0,989	0,974	1,000	0,958	0,952
F1-Score	0,982	0,991	0,964	1,000	0,956	0,950
Recall	0,979	0,992	0,958	1,000	0,954	0,950

Fonte: Elaborado pelo autor

Figura 13: Matriz de confusão para classificação da torre do térreo da UFABC



Fonte: Arquivos do autor

Na etapa de estimativa de coordenadas, o VAE apresentou menor EAM, conforme Tabela 3, e com erro médio na distância entre o real e predito de cerca de 5 metros para as torres 2 e 3, como mostrado na Tabela 4. Porém, o desempenho foi próximo, na média, ao do MLP com PCA. Em terceiro lugar o aparece o RF com PCA.

Convém destacar que o desempenho de todos os algoritmos foi superior na torre 3, seguida da torre 2, evidenciando como uma maior quantidade e densidade de pontos de coleta de sinais de Wi-Fi contribuem para melhorar a precisão das estimativas. Além disso, o uso de PCA melhorou o desempenho de todos os algoritmos. Para este experimento, o  $\beta$  com valores reduzidos resultou em melhor desempenho. A Figura 14 apresenta, de forma gráfica, os resultados de EAM mostrados na tabela 3.

Tabela 3: EAM globais obtidos das estimativas de localização *indoor* do térreo na UFABC

Torre	KNN (m)	KNN + PCA (m)	RF (m)	RF + PCA (m)	MLP (m)	MLP + PCA (m)	VAE (m)
1	7,11	5,73	7,19	5,18	5,89	5,71	4,97
2	4,33	4,27	3,49	4,43	3,87	2,97	3,22
3	2,97	3,38	3,35	3,14	3,34	3,04	3,28
<b>Média</b>	<b>4,80</b>	<b>4,46</b>	<b>4,68</b>	<b>4,25</b>	<b>4,37</b>	<b>3,91</b>	<b>3,82</b>

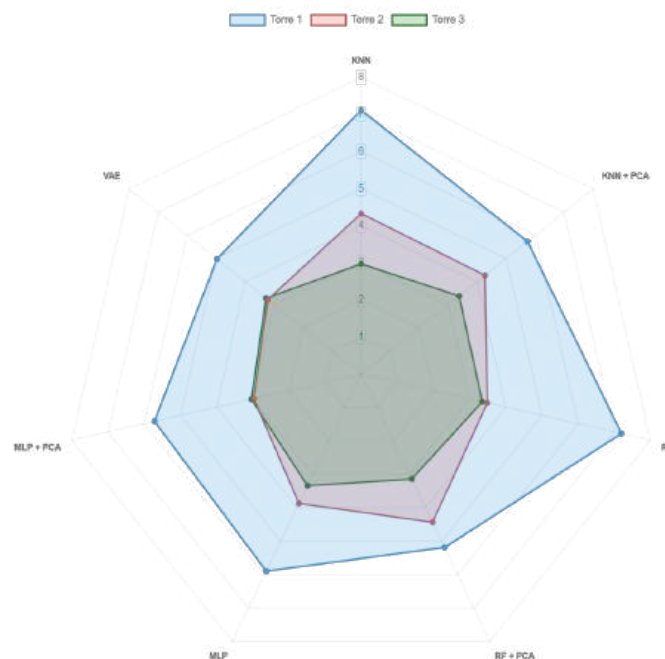
Fonte: Elaborado pelo autor

Tabela 4: Erro médio obtido das estimativas de localização *indoor* do térreo na UFABC

Torre	KNN (m)	KNN + PCA (m)	RF (m)	RF + PCA (m)	MLP (m)	MLP + PCA (m)	VAE (m)
1	11,25	9,49	10,85	7,96	9,13	9,72	8,41
2	7,31	7,20	5,99	7,52	6,32	4,75	5,13
3	4,86	5,33	5,33	5,17	5,38	5,00	5,04
<b>Média</b>	<b>7,81</b>	<b>7,34</b>	<b>7,39</b>	<b>6,88</b>	<b>6,94</b>	<b>6,49</b>	<b>6,19</b>

Fonte: Elaborado pelo autor

Figura 14: Gráfico radar do EAM das predições de coordenadas



Fonte: Elaborado pelo autor

## 6. Conclusão e Trabalhos Futuros

Neste trabalho, o emprego de um VAE foi realizado para se avaliar a melhoria de desempenho na estimativa de localização com base nos sinais de Wi-Fi, e houve a comparação de seu desempenho com um MLP isolado, com um KNN e com uma RF. Também foi avaliado o desempenho do PCA nestes algoritmos. A utilização do piso térreo da UFABC se mostrou possível com o desenvolvimento de um sistema completo, desde a interface do usuário, passando pela API até chegar ao banco de dados, com *back-end* implementados na nuvem e *front-end* no *smartphone*, que atendeu suas funções de armazenar um conjunto de treinamento e mostrar visualmente para o usuário sua posição estimada.

Com o VAE, observou-se melhoria no desempenho do MLP na estimativa de coordenadas, embora por pequena margem em relação ao MLP com PCA. O melhor desempenho com valores reduzidos de  $\beta$  indica que a preservação de informações relevantes impacta positivamente o modelo. Na classificação de torres, o KNN com redução de dimensionalidade através da PCA apresentou desempenho superior aos demais.

Foi possível observar que a acurácia das estimativas aumenta com maior densidade de pontos de varredura de Wi-Fi: todos os algoritmos tiveram melhor desempenho na torre 3, com pontos mais próximos entre si, e pior desempenho na torre 1, caracterizada por maior dispersão.

A aplicação de PCA resultou, em média, em ganhos de desempenho para os três algoritmos testados, exceto na tarefa de classificação com floresta aleatória, tanto na classificação de torres quanto na estimativa de coordenadas, sugerindo a presença de ruído e redundância nos dados.

Como trabalho futuro, propõe-se a investigação do uso do modelo generativo para criação de amostras sintéticas, no qual o decodificador, a partir da amostragem latente  $z$ , geraria novos dados no espaço original com o intuito de aumentar o conjunto de treinamento para os algoritmos supervisionados. Propõe-se também a automatização da coleta de dados por meio de drones, o que poderia permitir maior padronização e incremento na cobertura espacial nas medições de Wi-Fi.

## Referências Bibliográficas

Bank, Dor; Koenigstein, Noam; Giryes, Raja. **Autoencoders**. arXiv, 2020. Disponível em: <<https://arxiv.org/abs/2003.05991>>. Acesso em: 30 mar. 2026.

Bellecieri, Y.; Jabour, F. C.; Jabour, E. G. **Localização Indoor Baseada na Leitura Bidimensional do RSSI**. Seminários de Trabalho de Conclusão de Curso do Bacharelado em Sistemas de Informação, v. 1, n. 1, 2016.

Binoti, D. H. B.; Binoti, M. L. M. da S.; Leite, H. G. **Configuração de redes neurais artificiais para estimação do volume de árvores**. Ciência da Madeira, Pelotas, v. 5, n. 1, p. 58-67, 2014.

Castro, H. J. S. **Uma metodologia de localização Indoor híbrida para sistemas móveis**. Dissertação (Mestrado em Engenharia Elétrica e de Computação) – Universidade Federal do Rio Grande do Norte, Centro de Tecnologia, Programa de Pós-Graduação em Engenharia Elétrica e de Computação, Natal, 2018.

Castro, H. J. S.; Silva, I. M. D. da; Sampaio, S. C. **Uma metodologia de localização Indoor para smartphones em ambientes de Cidades Inteligentes**. In: WORKSHOP BRASILEIRO DE CIDADES INTELIGENTES (WBCI), 1., 2018, Natal. Anais [...]. Porto Alegre: Sociedade Brasileira de Computação, 2018. DOI: <https://doi.org/10.5753/wbci.2018.3224>

Dias Júnior, L. T. **Uso De Autocodificadores Variacionais Para a Detecção De Danos Estruturais**. Dissertação de mestrado – Universidade Federal de Juiz de Fora, Juiz de Fora, 2024.

Gao, L. **Channel state information fingerprinting based indoor localization: a deep learning approach**. Dissertação (Master of Science) — Auburn University, Auburn, 2015.

Geeksforgeeks. **What is Full Stack Development?** 20 nov. 2025. Disponível em: <<https://www.geeksforgeeks.org/blogs/what-is-full-stack-development/>>. Acesso em: 10 fev. 2026.

Gorjan, H. E.; Jiménez, V. P. G. **Improving Indoor WiFi Localization by Using Machine Learning Techniques**. *Sensors (Basel)*, v. 24, n. 19, p. 6293, 28 set. 2024. DOI: 10.3390/s24196293.

Higgins, I. et al.  **$\beta$ -VAE: Learning Basic Visual Concepts with a Constrained Variational Framework**. *International Conference on Learning Representations (ICLR)*, 2017.

IBM. **What is random forest?** Disponível em: <<https://www.ibm.com/think/topics/random-forest>> Acesso em: 24 abr. 2026. (s.d.a)

IBM. **Autoencoder**. Disponível em: <<https://www.ibm.com/think/topics/autoencoder>>. Acesso em: 30 mar. 2026. (s.d.b)

IBM. **O que é uma matriz de confusão?** Disponível em: <<https://www.ibm.com/br-pt/think/topics/confusion-matrix>>. Acesso em: 21 mar. 2026. (s.d.c)

IPEA – Instituto de Pesquisa Econômica Aplicada. **Inteligência artificial e redes neurais**. 2020. Disponível em: <<https://www.ipea.gov.br/cts/pt/central-de-conteudo/artigos/artigos/106-inteligencia-artificial-e-redes-neurais>>. Acesso em: 22 jul. 2025.

Issa, A.; Boukari, S.; Gital, A. Y.; Usman, A. **Wireless RSSI-Fingerprint and KNN Algorithm for Improved Accuracy Indoor Positioning System**. *Journal of Pure and Applied Sciences (Science Forum)*, v. 25, n. 3, p. 169–176, 15 out. 2025. Disponível em: <<https://atbuscienceforum.com.ng/index.php/jpas/article/view/199>>. Acesso em: 16 dez. 2025.

Kingma, D. P.; Welling, M. **An introduction to variational autoencoders**. 2019. Disponível em: <<https://arxiv.org/abs/1906.0269>> . Acesso em: 30 mar. 2026.

Kingma, D. P.; Welling, M. **Auto-Encoding Variational Bayes**. 2022. Disponível em: <<https://arxiv.org/abs/1906.0269>> . Acesso em: 01 abr. 2026.

Lee, S.; Kim, J.; Moon, N. **Random forest and WiFi fingerprint-based indoor location recognition system using smart watch**. Human-centric Computing and Information Sciences, [S.l.], v. 9, n. 1, art. 6, 2019. DOI: 10.1186/s13673-019-0168-7. Disponível em: <<https://doi.org/10.1186/s13673-019-0168-7>>. Acesso em: 17 dez. 2025.

Liashchynskiy, P.; Liashchynskiy, P. **Grid Search, Random Search, Genetic Algorithm: A Big Comparison for NAS**. Preprint arXiv – 1912.06059 [cs.LG], 2019. Disponível em: <<https://arxiv.org/abs/1912.06059>>. Acesso em: 10 nov. 2025.

Max, E. Z. G. **Seleção de instâncias baseado em aprendizado de métricas para kvizinhos mais próximos**. Dissertação de mestrado – Universidade Federal do Mato Grosso do Sul, Campo Grande, 2016.

Mittelstadt, R. S.; Pretto, F. **BLUEPATH: SISTEMA DE LOCALIZAÇÃO INDOOR**. Revista Destaques acadêmicos, [S. l.], v. 12, n. 4, 2020. DOI: 10.22410/issn.2176-3070.v12i4a2020.2758. Disponível em: <<https://www.univates.br/revistas/index.php/destaques/article/view/2758>>. Acesso em: 22 jul. 2025.

Ohashi, F. T. **Avaliação de quatro algoritmos de trilateração baseados em rssi considerando a variação do número de nós fixos**. Trabalho de Conclusão de Curso (Graduação em Engenharia Mecânica) Universidade Federal de São Carlos, São Carlos, 2023.

Oliveira, G. C. de. **Localização indoor utilizando a tecnologia LoRaWAN e aprendizado de máquina**. Monografia (Graduação em Engenharia de Telecomunicações) – Instituto Federal de Santa Catarina, São José, 2017.

Paiva, R. V. de O.; Ciribelli Jr, R. V.; Saraiva, E. M.; Oliveira, C. H. de; Mesquita, S. C. **Análise de crédito baseada em floresta aleatória**. Revista EAD em Debate: Uma Visão Integrada entre TIC, I.A. e Educação, [S. l.], v. 9, n. 2, 21 mar. 2024. Disponível em: <<https://periodicos.uniateneu.edu.br/index.php/revista-ead-em-debate/article/view/573>>. Acesso em: 09 nov. 2025.

Pinecone. ***K-Nearest Neighbor (KNN) Explained***. Disponível em: <<https://www.pinecone.io/learn/k-nearest-neighbor/>>. Acesso em: 29 mar. 2026.

Popescu, M-C.; Balas, V. E.; Perescu-Popescu, L.; Mastorakis, N. ***Multilayer Perceptron and Neural Networks***. *WSEAS Transactions on Circuits and Systems*, v. 15, p. 303-311, 2009. Disponível em: <<https://d1wqtxts1xzle7.cloudfront.net/69679997/29-485-libre.pdf?1631693012>>. Acesso em: 19 fev. 2026.

Ramchoun, H.; Ghanou, Y.; Ettaouil, M.; Idrissi, M. A. J. ***Multilayer Perceptron: Architecture Optimization and Training***. *International Journal of Interactive Multimedia and Artificial Intelligence*, v. 4, n. 1, set. 2016. DOI: 10.9781/ijimai.2016.415. Disponível em: <<https://reunir.unir.net/handle/123456789/11569>>. Acesso em: 19 fev. 2026

Rauber, T. W. ***Redes Neurais Artificiais***. Departamento de Informática – Universidade Federal do Espírito Santo, Vitória, 2005.

Riqueti, G. A.; Ribeiro, C. E.; Zárata, L. E. ***Classificando perfis de longevidade de bases de dados longitudinais usando Floresta Aleatória***. In: SYMPOSIUM ON KNOWLEDGE DISCOVERY, MINING AND LEARNING (KDMILE), 6. , 2018, São Paulo/SP. Anais [...]. Porto Alegre: Sociedade Brasileira de Computação, 2018 . p. 33-40. ISSN 2763-8944. DOI: <https://doi.org/10.5753/kdmile.2018.27382>.

Rodrigues, M. da S.; Carvalho, S. T. de. ***Sistemas de localização indoor utilizando Bluetooth Low Energy: Uma revisão sistemática***. In: WORKSHOP DE INICIAÇÃO CIENTÍFICA EM SISTEMAS DE INFORMAÇÃO – SIMPÓSIO BRASILEIRO DE SISTEMAS DE INFORMAÇÃO (SBSI), 15. 2019, Aracaju. Anais [...]. Porto Alegre: Sociedade Brasileira de Computação, 2019. p. 5-8. DOI: <https://doi.org/10.5753/sbsi.2019.7428>.

Rubio, R. J. H. ***Otimização de parâmetros de krigagem baseada na minimização do erro absoluto e do erro quadrático***. 2018. Tese (Doutorado em Engenharia de Minas, Metalúrgica e de Materiais) – Universidade Federal do Rio Grande do Sul, Porto Alegre, 2018.

Sarkar, S. K. **Principal Component Analysis**. In S. Rathod, B. Sailaja, N. Bandumula, S. Arun Kumar, P. A. Lakshmi Prasanna, P. Jeyakumar, A. Waris, P. Muthuraman, and R. M. Sundaram (Eds.), *Statistical Procedures for Analysing Agricultural Data using R* (pp. 139-156). ICAR - Indian Institute of Rice Research, Hyderabad, 2023.

Semaan, G. *et al.* **Um método para classificação de dados baseado nos k-vizinhos mais próximos para o reconhecimento de caracteres**. XIX Simpósio de Pesquisa Operacional e Logística da Marinha, 2019.

Silveira, D. D. B. **Modelos de tópicos baseados em autocodificadores variacionais utilizando as distribuições gumbel-softmax e mistura de normais-logísticas**. 2018. Dissertação (Mestrado em Informática) - Instituto de Computação. Universidade Federal do Amazonas. Manaus, 2018.

Uddin, S.; Haque, I.; Lu, H.; Moni, M. A.; Gide, E. **Comparative performance analysis of K-nearest neighbour (KNN) algorithm and its different variants for disease prediction**. *Scientific Reports*, v. 12, n. 1, p. 6256, 2022. Disponível em: <<https://www.nature.com/articles/s41598-022-10358-x>>. Acesso em: 29 mar. 2026.

Universidade Federal do ABC – UFABC. **Campi**. Disponível em: <<https://www.ufabc.edu.br/a-ufabc/campi>>. Acesso em: 10 fev. 2026.

Varella, C. A. A. **Análise de Componentes Principais**. Seropédica: Editora da UFRRJ, 2008.

Wang, Y.; Xiu, C.; Zhang, X.; Yang, D. **WiFi indoor localization with CSI fingerprinting-based random forest**. *Sensors*, Basel, v. 18, n. 9, art. 2869, 2018. DOI: 10.3390/s18092869. Disponível em: <<https://www.mdpi.com/1424-8220/18/9/2869>>. Acesso em: 17 dez. 2025.