

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

MIZUEL SIMÃO BORGOMOREIRA NISA

USO DE IA PARA ESTIMAR DESEMPENHO DE UMA REDE Wi-Fi

PONTA GROSSA

2026

MIZAEI SIMÃO BORGU MOREIRA NISA

USO DE IA PARA ESTIMAR DESEMPENHO DE UMA REDE Wi-Fi

Using AI To Estimate the Performance of a Wi-Fi Network

Trabalho de Conclusão de Curso de Graduação
apresentado como requisito para obtenção do
título de Bacharel em Ciência da Computação, da
Universidade Tecnológica Federal do Paraná
(UTFPR).

Orientador: Prof. Dr. Augusto Foronda

PONTA GROSSA

2026



[4.0 Internacional](https://creativecommons.org/licenses/by/4.0/)

Esta licença permite compartilhamento, remixe, adaptação e criação a partir do trabalho, mesmo para fins comerciais, desde que sejam atribuídos créditos ao(s) autor(es). Conteúdos elaborados por terceiros, citados e referenciados nesta obra não são cobertos pela licença.

MIZAEEL SIMÃO BORG0 MOREIRA NISA

USO DE IA PARA ESTIMAR DESEMPENHO DE UMA REDE Wi-Fi

Trabalho de Conclusão de Curso de Graduação
apresentado como requisito para obtenção do
título de Bacharel em Ciência da Computação, da
Universidade Tecnológica Federal do Paraná
(UTFPR).

Data de aprovação: 19 / Maio /2026

Augusto Foronda
Doutorado
Universidade Tecnológica Federal do Paraná

Geraldo Ranthum
Doutorado
Universidade Tecnológica Federal do Paraná

Richard Ribeiro
Doutorado
Universidade Tecnológica Federal do Paraná

PONTA GROSSA

2026

RESUMO

Este trabalho apresenta o desenvolvimento e a avaliação modelo de IA para predições voltados à estimativa de desempenho de redes Wi-Fi em ambientes domésticos e institucionais. A justificativa da pesquisa está na crescente demanda por conexões sem fio eficientes, de baixa latência e custo reduzido, essenciais para aplicações como videoconferências, jogos online e dispositivos da Internet das Coisas. O objetivo geral foi construir um sistema capaz de prever a vazão da rede (Mbps), considerando variáveis como distância do ponto de acesso e barreiras físicas. A metodologia envolveu três etapas principais: coleta automatizada de dados em ambiente controlado, tratamento e organização das medições com auxílio da linguagem Python e bibliotecas estatísticas. Os resultados demonstraram que o modelo é capaz de representar o comportamento real da atenuação do sinal, mesmo em cenários com diferentes condições ambientais, destacando-se pela simplicidade de implementação e baixo custo computacional. A análise técnica evidenciou que o tratamento de *outliers* foi decisivo para alcançar desempenho próximo ao ideal ($R^2 \approx 0,91$). Conclui-se que a abordagem adotada contribui para a otimização de redes sem fio, oferecendo suporte à tomada de decisão sobre posicionamento de equipamentos e infraestrutura, além de abrir perspectivas para pesquisas futuras com modelos híbridos e técnicas mais avançadas de aprendizado de máquina.

Palavras-chave: rede wi-fi; inteligência artificial; atenuação de sinal; predição estatística; latência; conectividade sem fio.

ABSTRACT

This study presents the development and evaluation of a predictive model of IA aimed at estimating the performance of Wi-Fi networks in domestic and institutional environments. The research is justified by the growing demand for efficient, low-latency, and cost-effective wireless connections, which are essential for applications such as video conferencing, online gaming, and Internet of Things devices. The main objective was to build a system capable of predicting network throughput (Mbps), considering variables such as the distance from the access point and physical barriers. The methodology comprised three main stages: automated data collection in a controlled environment, processing and organization of measurements using the Python programming language and statistical libraries. The results showed that the model can represent the real behavior of signal attenuation even under different environmental conditions, standing out for its simplicity of implementation and low computational cost. The technical analysis revealed that outlier treatment was decisive for achieving near-ideal performance ($R^2 \approx 0.91$), and that the Power Law model reproduced the asymptotic behavior of the signal more realistically at greater distances. It is concluded that the proposed approach contributes to the optimization of wireless networks, supporting decision-making regarding equipment positioning and infrastructure, while opening perspectives for future research involving hybrid models and more advanced machine learning techniques.

Keywords: wi-fi network; artificial intelligence; linear regression; signal attenuation; machine learning ; latency ; wireless connectivity

SUMÁRIO

1	INTRODUÇÃO.....	8
1.1	Objetivos.....	9
1.1.1	Objetivos Geral.....	9
1.1.2	Objetivos Específicos.....	9
2	FUNDAMENTAÇÃO TEÓRICA.....	9
2.1	Tecnologia Wi-Fi.....	10
2.2	Parâmetros de Desempenho.....	10
2.2.1	Nível de Sinal (RSSI/dbm).....	11
2.2.2	Velocidade de Transmissão (Throughput).....	11
2.2.3	Latência e Tempo de Resposta.....	12
2.3	Fontes de Atenuação e Interferência.....	12
2.3.1	Atenuação por Barreiras Físicas.....	13
2.4	Modelos Preditivos em Redes.....	14
2.4.1	Regressão Linear.....	14
2.4.2	Regressão Logarítmica (ou Modelo Log-Linear).....	14
2.4.3	Random Forest Regressor.....	15
2.4.4	Regressão Ridge.....	16
2.4.5	Regressão Polinomial.....	16
3	DESENVOLVIMENTO.....	18
3.1	Planejamento e Configuração do Ambiente.....	18
3.2	Bibliotecas Usadas.....	19
3.3	Descrição do módulo de monitoramento.....	20
3.4	Ferramenta de Coleta de Dados.....	21
3.5	Bibliotecas e Preparação do Ambiente.....	26
3.6	Processamento e Divisão dos Dados.....	28
3.7	Treinamento dos Modelos de Predição.....	30
4	RESULTADOS.....	32
4.1	Resultado Final da IA.....	35
4.2	Análise Técnica das Predições Nulas.....	36
5	CONCLUSÃO.....	38
	REFERÊNCIAS.....	39

1 INTRODUÇÃO

A rede Wi-Fi, sigla para (*Wireless Fidelity*), Fidelidade Sem Fio, é uma tecnologia de comunicação sem fio que permite a transmissão de dados por meio de ondas de rádio, conectando dispositivos a uma rede local ou à internet. Sua popularização transformou a forma como interagimos com o ambiente digital, oferecendo mobilidade, praticidade e conectividade em tempo real. De acordo com o padrão IEEE 802.11 (IEEE, 2021), o Wi-Fi opera em faixas de frequência específicas e utiliza protocolos de comunicação padronizados, tornando-se uma solução versátil para residências, empresas e instituições públicas. Entretanto, seu desempenho está sujeito a múltiplas variáveis técnicas e estruturais que demandam constante monitoramento e otimização — sobretudo em cenários nos quais a qualidade da conexão é essencial para o funcionamento de aplicações digitais.

Com o avanço do uso de dispositivos móveis e da *Internet of Things* (IoT) Internet das coisas, tornou-se essencial dispor de redes Wi-Fi eficientes e com baixa latência. No entanto, essas redes são suscetíveis a interferências físicas — como paredes, móveis e eletrodomésticos — e a fatores ambientais, como a quantidade de usuários conectados e a intensidade do sinal. Tais interferências impactam diretamente o nível e velocidade do sinal, dificultando a previsibilidade do desempenho, especialmente em aplicações sensíveis à latência. Essa relação entre interferência e desempenho de redes sem fio é amplamente discutida por Tanenbaum e Wetherall (2011), que destacam a complexidade da comunicação em ambientes reais.

Diante da complexidade das relações entre distância, obstáculos e velocidade do Wi-Fi, este trabalho explora diferentes modelos preditivos para estimar a velocidade de download em função da distância e do número de paredes. Mesmo abordagens mais simples, como a regressão linear, sejam frequentemente consideradas por possuir uma baixa complexidade de implementação e baixo custo de recursos computacionais, avaliamos seu desempenho em comparação com modelos mais robustos, como a regressão logarítmica e o *Random Forest*. O objetivo é identificar qual abordagem oferece a melhor precisão na previsão da

velocidade de download, considerando a natureza não-linear da propagação do sinal Wi-Fi em ambientes com barreiras físicas.

1.1 Objetivos

1.1.1 Objetivos Geral

Desenvolver um modelo de IA capaz de estimar a velocidade de uma rede Wi-Fi.

1.1.2 Objetivos Específicos

- Automatizar a coleta de dados de intensidade de sinal e velocidade da rede Wi-Fi em diferentes distâncias e condições ambientais;
- Selecionar as variáveis mais relevantes para a predição
- Converter coordenadas cartesianas em distância euclidiana e filtrar valores de obstrução física para garantir consistência dos dados;
- Aplicar técnicas de tratamento de *outliers* para eliminar medições anômalas e melhorar o ajuste dos modelos;
- Testar diversos tipos de regressão no modelo para representar o comportamento real da atenuação;
- Avaliar o desempenho dos modelos propostos e sua aplicabilidade na otimização de redes sem fio.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo são apresentados os conceitos fundamentais que sustentam o desenvolvimento deste trabalho. São abordados os princípios da tecnologia Wi-Fi, suas principais características operacionais e os parâmetros que influenciam diretamente o desempenho das redes sem fio, como intensidade do sinal (RSSI), atenuação, interferências e *throughput*. Também são discutidos fatores estruturais que afetam a propagação das ondas de rádio em ambientes internos, além de estudos e pesquisas relacionados à estabilidade de redes Wi-Fi e aos métodos de análise utilizados na literatura. Essa base teórica é essencial para contextualizar as medições realizadas, justificar a metodologia adotada e oferecer suporte à interpretação dos resultados experimentais apresentados nos capítulos seguintes.

2.1 Tecnologia Wi-Fi

A tecnologia Wi-Fi, ou *Wireless Fidelity*, é o padrão global para redes locais sem fio (WLANs), regida primariamente pela família de especificações IEEE 802.11. Esse conjunto de normas padroniza os protocolos da camada de acesso ao meio (MAC) e da camada física (PHY) para comunicação por radiofrequência, sendo a base para a conectividade onipresente em ambientes domésticos e corporativos (Shaaban & Badawy, 2017).

Desde o lançamento dos padrões iniciais (como 802.11b e 802.11g) que operam na banda de 2,4 GHz, as revisões subsequentes (802.11ac, 802.11ax/Wi-Fi 6) introduziram o uso mais amplo das bandas de 5 GHz e 6 GHz, elevando o *throughput* e a eficiência espectral por meio de técnicas como MU-MIMO (Múltiplos Usuários com Múltiplas Entradas e Múltiplas Saídas) e OFDMA (Acesso Múltiplo por Divisão Ortogonal de Frequência) (IEEE, 2021).

2.2 Parâmetros de Desempenho

A avaliação do desempenho de redes Wi-Fi requer a análise de parâmetros que influenciam diretamente a qualidade da comunicação sem fio. Esses indicadores permitem compreender o comportamento do sinal em diferentes condições de operação, tanto em ambientes domésticos quanto institucionais. Entre os principais parâmetros estão o nível de sinal (RSSI), que reflete a potência recebida e a atenuação no meio propagou (Gupta & Verma, 2019), a velocidade de transmissão

(*throughput*), que determina a taxa real de dados transferidos (Kurose & Ross, 2021), e a latência, fundamental para aplicações em tempo real (Tanenbaum & Wetherall, 2019). Juntas, essas métricas fornecem uma visão abrangente sobre a eficiência, a capacidade de vazão e a estabilidade da rede. A correta interpretação e correlação dessas variáveis é essencial para o desenvolvimento de modelos de predição de cobertura e ferramentas de diagnóstico voltadas à otimização do desempenho das WLANs, conforme as especificações e evoluções da família IEEE 802.11 (Ieee, 2021; Shaaban & Badawy, 2017; Peraha & Stacey, 2014).

2.2.1 Nível de Sinal (RSSI/dbm)

O Nível de Sinal Recebido (*Received Signal Strength Indicator – RSSI*) é a medida da potência do sinal de rádio captada pelo dispositivo cliente a partir do Ponto de Acesso (AP). Esta potência é expressa em dBm (decibel-milliwatts), uma unidade logarítmica que se correlaciona diretamente com a distância entre transmissor e receptor e a atenuação causada pelo ambiente (RIFKI, 2020).

Estudos têm demonstrado que o RSSI é uma métrica confiável para estimar a capacidade de transmissão (*throughput*) e é fundamental em algoritmos de localização e planejamento de topologia (NAUVALDI *et al.*, 2023; MDPI, 2021). Contudo, o RSSI é vulnerável a flutuações e à influência de eventos *multipath* ou seja o sinal Wi-Fi se propaga por múltiplos caminhos devido a reflexões em superfícies como paredes e objetos. Essas múltiplas trajetórias fazem com que o receptor receba o sinal em diferentes instantes, causando interferências que podem aumentar ou reduzir a potência medida, e seu valor tipicamente se enfraquece exponencialmente à medida que a distância aumenta – um princípio que fundamenta a aplicação da regressão linear na modelagem de propagação em ambientes internos (IFIP Digital Library, 2020).

2.2.2 Velocidade de Transmissão (*Throughput*)

O *throughput*, medido em megabits por segundo (Mbps), representa a quantidade real de dados que é transferida com sucesso do ponto de origem para o destino em um determinado intervalo de tempo. Diferente das taxas nominais teóricas estabelecidas pelos padrões 802.11, o *throughput* é a métrica de

desempenho real, sendo diretamente afetada pelo RSSI, pela taxa de erro de pacotes (PER) e pela sobrecarga de protocolo (*overhead*) (Researchgate, 2017).

A relação entre RSSI e *throughput* é bem estabelecida na literatura, sendo que a queda no nível de sinal (valores de dBm mais negativos) está diretamente associada a uma redução na taxa de transferência. Isso ocorre porque o hardware da rede é obrigado a selecionar um Esquema de Modulação e Codificação (MCS) menos eficiente para compensar o sinal mais fraco, o que, por sua vez, diminui a velocidade máxima de transmissão (Tauber; Bhatti, 2013).

2.2.3 Latência e Tempo de Resposta

A Latência é o atraso temporal no envio de pacotes através da rede, geralmente medida em milissegundos (ms) pelo comando Ping. Nas redes sem fio, a latência é um indicador para a qualidade de serviços em tempo real, como Voz sobre IP (VoIP) e aplicações interativas.

O padrão 802.11 utiliza o protocolo CSMA/CA (*Carrier-Sense Multiple Access with Collision Avoidance*) para gerenciar o acesso ao meio, o que introduz um tempo de espera (*backoff*) que contribui para o atraso total (IEEE, 2021). A evolução dos padrões (como o 802.11ax) tem buscado ativamente reduzir a latência e aumentar a previsibilidade do tempo de resposta por meio de técnicas de acesso mais eficientes, como o OFDMA.

A correlação entre essas três métricas — Nível de Sinal, Velocidade e Latência — e as características do ambiente (distância e barreiras) é o foco central deste trabalho e foi modelada por meio de técnicas de regressão.

2.3 Fontes de Atenuação e Interferência

A propagação de um sinal Wi-Fi em ambientes internos, regulada pelos padrões IEEE 802.11, não ocorre em um vácuo ideal, mas sim em um meio complexo onde a potência do sinal (RSSI) e, conseqüentemente, o *throughput* são degradados por uma série de fenômenos eletromagnéticos como a iluminação, eletrônicos, eletrodomésticos. A degradação da qualidade do sinal é causada fundamentalmente por dois fatores: atenuação e interferência.

2.3.1 Atenuação por Barreiras Físicas

A atenuação refere-se à redução da intensidade de potência do sinal de rádio à medida que ele se propaga. O modelo de propagação em ambientes internos é um dos desafios centrais na otimização de redes, pois ele se desvia significativamente do modelo de espaço livre, sendo dominado por fenômenos como reflexão, difração e dispersão (Marçal, N. A, 2023).

A principal causa da atenuação em ambientes residenciais e corporativos são as barreiras físicas. Quando as ondas eletromagnéticas encontram materiais de construção, parte da energia é absorvida ou refletida, resultando em uma perda de sinal mensurável em dB. O nível de perda é diretamente dependente da composição e espessura do material, bem como da frequência de operação conforme pode ser visto no Quadro 1. (Ferreira, L.; Montanha, G. K., 2017).

Quadro 1 - Influência das Barreiras Físicas na Atenuação do Sinal Wi-Fi

Material	Potencial de Atenuação	Perda Média de Sinal (dB)	Efeito na Propagação
Metal	Alto (Maior Fator)	20 a 30 dB	Causa alta reflexão e absorção, atuando como uma blindagem de sinal.
Concreto/Tijolo	Médio a Alto	10 a 20 dB	Absorve o sinal, sendo a principal barreira estrutural em edificações.
Água/Parede com Tubulação	Médio a Alto	12 a 14 dB	A água tem alta capacidade de absorção de radiofrequência, atenuando o sinal.
Madeira/Gesso/Vidro	Baixo a Médio	3 a 8 dB	Causa pouca absorção e maior refração.

Fonte: Network Layer (Kazaz et al., 2021)

2.4 Modelos Preditivos em Redes

O desempenho de uma rede Wi-Fi, intrinsecamente ligado ao valor do (RSSI) e ao *throughput*, é determinado por um conjunto complexo de fatores, incluindo a distância do transmissor, o tipo e número de barreiras físicas e a interferência eletromagnética. Em ambientes internos, a dificuldade em modelar com precisão o caminho de propagação do sinal levou ao desenvolvimento de Modelos Preditivos Estatísticos para estimar o desempenho de forma mais acurada do que os modelos puramente determinísticos baseados em fórmulas físicas (Shaaban; Badawy, 2017).

A utilização de técnicas de modelagem baseada em dados oferece uma solução ao permitir que o sistema aprenda os padrões de atenuação a partir de dados coletados em campo, resultando em modelos de propagação mais precisos e adaptados ao ambiente específico (Barbosa, B.S.S, 2024).

2.4.1 Regressão Linear

A Regressão Linear é um dos modelos estatísticos mais fundamentais, que assume uma relação linear entre uma variável dependente (neste caso, a velocidade da rede) e uma ou mais variáveis independentes (distância, barreiras). (Tanenbaum; Wetherall, 2011). Sua simplicidade e interpretabilidade a tornam uma escolha comum para análises iniciais (Montgomery; Peck; Vining, 2021). No entanto, sua eficácia é limitada em cenários onde as relações são inerentemente não-lineares, como é frequentemente o caso na propagação de sinais de rádio. A atenuação do sinal Wi-Fi, por exemplo, não se comporta de maneira estritamente linear com o aumento da distância e a presença de obstáculos, o que pode levar a um desempenho preditivo subótimo.

2.4.2 Regressão Logarítmica (ou Modelo Log-Linear)

A utilização da Regressão Polinomial de segundo grau fundamenta-se na necessidade de modelar comportamentos não lineares inerentes à propagação de sinais de radiofrequência, uma vez que a atenuação da potência e a consequente perda de velocidade não ocorrem de forma puramente linear em relação à distância (Rappaport, 2002). Ao transformar o espaço de características original em um plano de maior dimensão, o modelo permite capturar a curvatura da degradação do sinal e

as interações complexas entre a distância e os obstáculos físicos, oferecendo flexibilidade para ajustar a parábola de decaimento sem a necessidade de assumir a rigidez de um modelo linear estrito (Montgomery; Peck; Vining, 2021; Géron, 2012). Esta abordagem transforma uma ou mais variáveis independentes (geralmente a distância) usando uma função logarítmica antes de aplicar o modelo linear. Essa transformação busca linearizar a relação, permitindo que a regressão linear capture melhor o comportamento do sinal em relação à distância (Rappaport, 2002). Reconhecendo a natureza não linear da propagação do sinal Wi-Fi, onde a potência do sinal decai de forma inversamente proporcional ao logaritmo da distância — fenômeno formalizado pelo Modelo de Perda de Percurso Log-Distância (Log-Distance Path Loss Model) (Rappaport, 2010; Goldsmith, 2005) —, a Regressão Logarítmica surge como uma adaptação direta da regressão linear matemática. Essa abordagem permite linearizar a relação entre as variáveis por meio de uma transformação logarítmica na variável preditora, tornando o ajuste por mínimos quadrados aplicável a fenômenos de decaimento exponencial ou logarítmico (Montgomery; Peck; Vining, 2021).

2.4.3 *Random Forest Regressor*

O *Random Forest* é um algoritmo de aprendizado de máquina do tipo ensemble, o que significa que combina diversos modelos individuais para produzir uma previsão final mais precisa e robusta. Esse método baseia-se na construção de múltiplas árvores de decisão durante o treinamento e na agregação de suas previsões, resultando em um desempenho superior e menor suscetibilidade ao sobreajuste (*overfitting*) e melhora a capacidade de generalização do modelo (Breiman, 2001). Sua capacidade de modelar relações não lineares complexas e capturar interações entre variáveis independentes o torna particularmente adequado para problemas onde as dinâmicas subjacentes são intrincadas, como a predição do desempenho de redes Wi-Fi em ambientes com múltiplos obstáculos e variações de distância (ALAKAŞ et al., 2022; ZHOU et al., 2020). Esse modelo é capaz de aprender padrões mais sofisticados nos dados, superando as limitações dos modelos lineares tradicionais em cenários de severa atenuação e sombreamento de sinal (SAMPATH et al., 2018).

2.4.4 Regressão Ridge

A aplicação da Regressão Ridge possui uma capacidade de mitigar problemas de multicolinearidade e *overfitting*, comuns em modelos de predição de sinais onde variáveis como distância e obstáculos apresentam correlações complexas. Ao contrário da regressão linear convencional, a técnica Ridge consegue estabilizar os coeficientes, resultando em um modelo com maior poder de generalização. Conforme discutido por Hastie, Tibshirani e Friedman (2009), essa abordagem reduz a variância das estimativas em troca de um pequeno acréscimo no viés, o que frequentemente minimiza o erro quadrático médio em dados não vistos. No contexto específico de propagação de sinal, que é inerentemente ruidoso, Hoerl e Kennard (1970) demonstram que o estimador Ridge oferece uma solução significativamente mais estável. Complementarmente, Pedregosa et al. (2011) ressaltam que tal regularização é fundamental em aplicações de engenharia para evitar que o modelo reaja de forma desproporcional a variações ruidosas nos dados de entrada, garantindo uma estimativa de velocidade em Mbps mais robusta e confiável.

2.4.5 Regressão Polinomial

A utilização da Regressão Polinomial de segundo grau fundamenta-se na necessidade de modelar comportamentos não lineares inerentes à propagação de sinais de radiofrequência, uma vez que a atenuação da potência e a consequente perda de velocidade não ocorrem de forma puramente linear em relação à distância (Rappaport, 2010). Ao transformar o espaço de características original em um plano de maior dimensão, o modelo permite capturar a curvatura da degradação do sinal e as interações complexas entre a distância e os obstáculos físicos, oferecendo flexibilidade para ajustar a parábola de decaimento sem a necessidade de assumir a rigidez de um modelo linear estrito (Montgomery; Peck 2021). De acordo com Montgomery, Peck e Vining (2021), a regressão polinomial é uma ferramenta poderosa para aproximar funções desconhecidas em regiões limitadas, permitindo que o modelo linear se ajuste a dados com padrões curvilíneos sem perder a simplicidade matemática do estimador. Sob a ótica de Kuhn e Johnson (2013), a

inclusão de termos quadráticos é uma forma eficaz de engenharia de atributos que ajuda a reduzir o viés do modelo, capturando nuances que modelos lineares simples ignoraram. No contexto deste trabalho, essa flexibilidade permitiu que o modelo Polinomial G2 obtivesse um dos melhores índices de R^2 , demonstrando uma aderência superior às flutuações reais observadas nas medições de Mbps conforme a distância aumenta.

3 DESENVOLVIMENTO

Neste capítulo, são apresentadas as etapas de desenvolvimento da aplicação proposta, incluindo as ferramentas utilizadas, bibliotecas empregadas e estrutura geral do sistema. O objetivo é detalhar os recursos técnicos e metodológicos que possibilitaram a construção da ferramenta de medição e análise de desempenho de redes Wi-Fi.

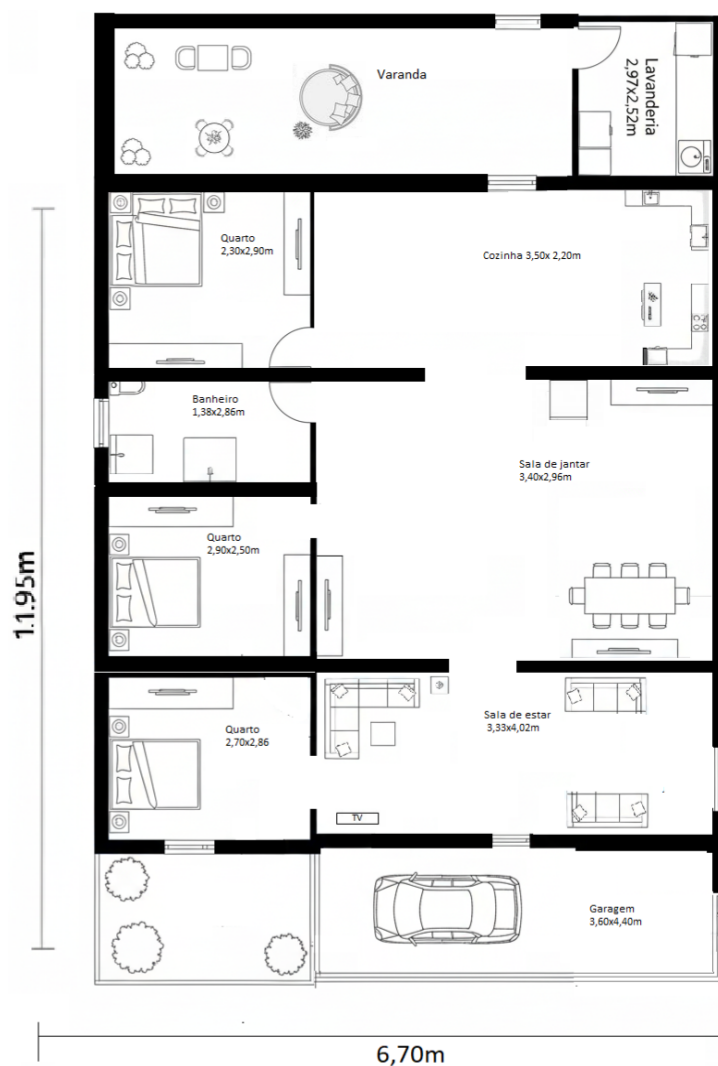
Inicialmente, são descritas as principais ferramentas de software e bibliotecas utilizadas para o desenvolvimento da aplicação, justificando suas escolhas de acordo com as necessidades do projeto. Em seguida, apresenta-se a arquitetura da ferramenta, contemplando a estrutura do código, o fluxo de execução e as funções implementadas para coleta de dados e tratamento das medições.

Por fim, são detalhados os procedimentos de testes aplicados nos diferentes ambientes residenciais, assim como os critérios adotados para armazenamento, organização e análise dos resultados coletados.

3.1 Planejamento e Configuração do Ambiente

A Figura 1 apresenta a planta da residência utilizada no projeto, composta por 9 cômodos, além de um jardim e uma garagem. O imóvel possui 3 quartos, cozinha, banheiro, sala de jantar, lavanderia, sala de estar, jardim e garagem, com dimensões de 6,70m x 11,95m. O ponto de acesso (AP) está localizado no primeiro quarto, próximo à cozinha. O segundo quarto fica entre o primeiro e o banheiro, em frente à sala de jantar, enquanto o terceiro quarto está ao lado do segundo e da sala de estar. A varanda, lavanderia e sala de jantar apresentam melhor sinal Wi-Fi devido à proximidade com o AP. Já o jardim e a garagem, por estarem mais afastados e separados por barreiras físicas e interferências eletrônicas, sofrem maior perda de sinal

Figura 1 - Planta da casa



Fonte: Autoria Própria (2026)

3.2 Bibliotecas Usadas

Nesta seção são apresentadas as principais bibliotecas empregadas no desenvolvimento do sistema. Cada uma desempenha um papel essencial na coleta, tratamento e análise dos dados de desempenho da rede Wi-Fi, bem como na interface e automação das medições.

- **Pandas:** utilizada para manipulação, filtragem e análise dos conjuntos de dados coletados, permitindo a organização das medições em tabelas estruturadas.
- **NumPy:** responsável por operações matemáticas e cálculos numéricos eficientes, especialmente no tratamento de matrizes e vetores de dados.
- **Matplotlib:** aplicada na geração de gráficos e visualizações, possibilitando a interpretação visual dos resultados obtidos nas medições.
- **Speedtest (API):** biblioteca utilizada para realizar medições automáticas de velocidade de download, upload e latência, servindo como base de comparação com os dados da ferramenta desenvolvida.
- **Subprocess:** execução de comandos do sistema operacional, integrando dados externos — como medições via Speedtest CLI — à aplicação.
- **Re (Regular Expressions):** extração e filtragem de informações textuais a partir de saídas de APIs ou comandos de terminal.
- **JSON:** manipulação e leitura de dados em formato JSON, utilizados para estruturar e armazenar resultados de medições.
- **CSV:** leitura e gravação de arquivos no formato CSV, permitindo exportar resultados e tabelas para análise estatística posterior.

3.3 Descrição do módulo de monitoramento

O funcionamento do código baseia-se em um processo contínuo que envolve a coleta, o processamento e a visualização de dados de intensidade do sinal Wi-Fi. Inicialmente, o programa executa um comando do sistema operacional Windows (`netsh wlan show interfaces`) por meio da biblioteca `subprocess`, obtendo as informações da interface de rede sem fio ativa. A partir desse resultado, a biblioteca `Re` utiliza uma expressão regular para extrair o valor percentual que representa a intensidade do sinal Wi-Fi, retornando-o como um número inteiro entre 0 e 100. Em seguida, a planta baixa do ambiente é carregada através da função `mpimg.imread()`, que transforma a imagem em uma matriz numérica, cujas dimensões (altura e largura) são utilizadas como base para o mapeamento espacial dos pontos medidos.

O usuário interage com a aplicação clicando sobre a planta exibida na tela. Cada clique é capturado pelo evento `button_press_event`, que aciona uma função

responsável por registrar as coordenadas (x, y) do ponto clicado e medir o sinal Wi-Fi naquele instante. Esses dados são armazenados em uma lista, formando um conjunto de medições que relacionam posição e intensidade do sinal. Após coletar os pontos desejados, o código cria uma matriz Numpy do mesmo tamanho da imagem da planta, inicializada com zeros, e insere os valores de intensidade nas posições correspondentes aos pontos medidos, formando assim um mapa inicial (*heatmap*) com valores discretos.

Para tornar o resultado mais representativo e simular a difusão natural do sinal de rádio, o programa aplica um filtro gaussiano, fornecido pela função `gaussian_filter` da biblioteca `scipy.ndimage`. Essa suavização interpola os valores entre os pontos medidos, criando uma transição contínua que se assemelha à variação real da intensidade do sinal em um ambiente físico. Por fim, a imagem da planta é exibida novamente como plano de fundo, e o mapa suavizado é sobreposto com uma coloração gradual (`colormap "jet"`) e um nível de transparência ajustado. A função `plt.colorbar()` adiciona uma barra de cores indicando a escala percentual de intensidade, resultando em um mapa de calor que ilustra visualmente as regiões de maior e menor cobertura do sinal Wi-Fi dentro do ambiente analisado.

3.4 Ferramenta de Coleta de Dados

Enquanto o *Speedtest* na web emprega uma versão completa que contempla uma ampla rede de servidores distribuídos globalmente, com mecanismos de geolocalização para selecionar automaticamente o servidor mais adequado e garantir medições precisas e consistentes, a API disponibilizada para integração apresenta limitações significativas.

A API empregada neste projeto realiza a rotação automática de servidores durante os testes, o que pode introduzir variações e margens de erro mais elevadas nos resultados de latência e velocidade. Para mitigar esse problema, foi desenvolvido um mecanismo que fixa o servidor de teste usando id e nome do servidor que é salvo em um arquivo json é carregado pelo programa como mostrado na figura 2 abaixo, permitindo maior estabilidade nas medições e possibilitando comparações consistentes entre diferentes execuções.

Essa solução foi essencial para garantir a confiabilidade dos dados obtidos, além de viabilizar futuras etapas de compilação e testes de campo com maior precisão e continuidade.

Figura 2- Salvando ID em um JSON



```
1 def salvar_servidor_id(server_info):
2     """Salva o ID e nome do servidor em um arquivo JSON."""
3     try:
4         with open(ARQUIVO_CONFIG, "w") as f:
5             json.dump(server_info, f)
6     except Exception as e:
7         print(f"Erro ao salvar ID do servidor: {e}")
8
9 def carregar_servidor_id():
10    """Carrega o ID e nome do servidor do arquivo JSON."""
11    try:
12        with open(ARQUIVO_CONFIG, "r") as f:
13            return json.load(f)
14    except FileNotFoundError:
15        return None
16    except Exception as e:
17        print(f"Erro ao carregar ID do servidor: {e}")
18        return None
```

Fonte: Autoria Própria (2026)

Outro aspecto importante implementado nesta versão foi a inclusão das dimensões reais da planta da residência, considerando altura e largura compatíveis com as medidas físicas do ambiente. Essa abordagem permite uma representação mais fiel do espaço e possibilita a realização de testes em diferentes tipos de plantas, ampliando a aplicabilidade da ferramenta.

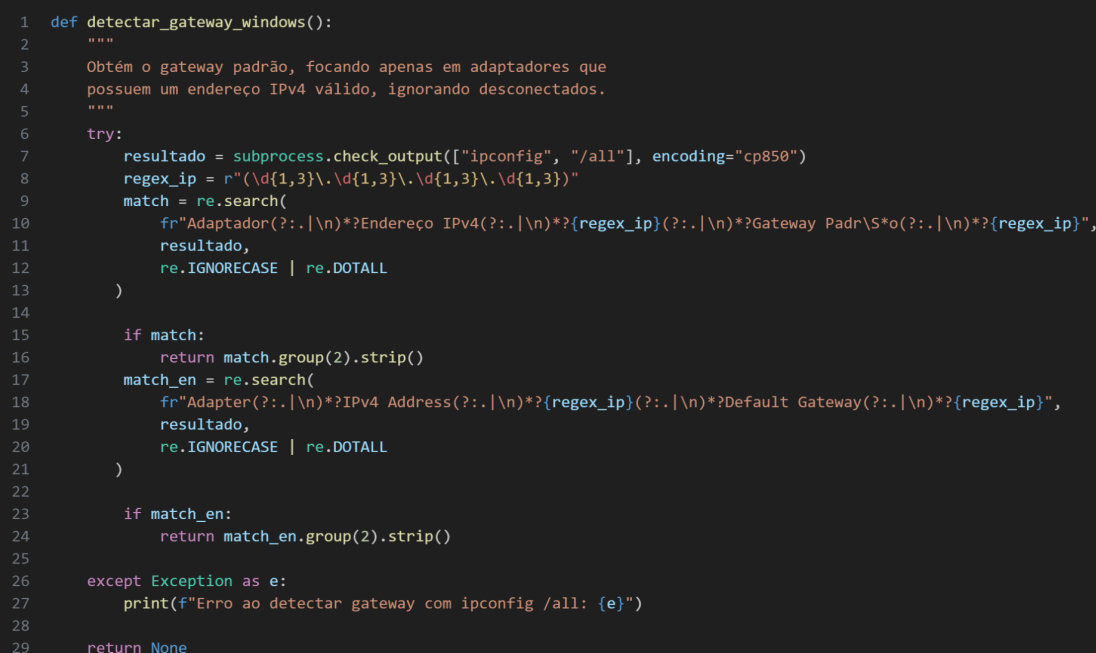
Entretanto, a principal evolução foi a implementação de testes utilizando o *Access Point* (AP) — o ponto de acesso responsável pela emissão do sinal Wi-Fi. Essa adição foi fundamental porque tanto a API quanto o site do Speedtest realizam medições de latência (em milissegundos) com base na comunicação entre o

dispositivo testado e os servidores externos da rede principal, ou seja, entre o gateway do provedor e o servidor de teste.

Ao introduzir o AP como referência de origem do sinal, o trajeto de medição passa a incluir o gateway do modem até o servidor de teste, refletindo de forma mais realista o comportamento da rede local. Essa abordagem reduz discrepâncias observadas nas medições realizadas apenas via API, proporcionando resultados mais consistentes e próximos da condição real de uso.

Para facilitar o uso da ferramenta em nível teste, um tipo de mecanismo que detecta o gateway automático do windows também foi desenvolvido.

Figura 3- Detectando gateway automático



```

1 def detectar_gateway_windows():
2     """
3     Obtém o gateway padrão, focando apenas em adaptadores que
4     possuem um endereço IPv4 válido, ignorando desconectados.
5     """
6     try:
7         resultado = subprocess.check_output(["ipconfig", "/all"], encoding="cp850")
8         regex_ip = r"(\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3})"
9         match = re.search(
10             fr"Adaptador(?:\n)*?Endereço IPv4(?:\n)*?{regex_ip}(?:\n)*?Gateway Padr\o(?:\n)*?{regex_ip}",
11             resultado,
12             re.IGNORECASE | re.DOTALL
13         )
14
15         if match:
16             return match.group(2).strip()
17         match_en = re.search(
18             fr"Adapter(?:\n)*?IPv4 Address(?:\n)*?{regex_ip}(?:\n)*?Default Gateway(?:\n)*?{regex_ip}",
19             resultado,
20             re.IGNORECASE | re.DOTALL
21         )
22
23         if match_en:
24             return match_en.group(2).strip()
25
26     except Exception as e:
27         print(f"Erro ao detectar gateway com ipconfig /all: {e}")
28
29     return None

```

Fonte: Autoria Própria (2026)

A Figura 3 mostra as bibliotecas “subprocess” tanto quanto “re”, para criar um regex que usa os comandos internos no windows como ipconfig para acessar informações em relação ao gateway e ao ipconfig, em caso de sucesso esse ipv4 é usado em outra parte do programa para conseguir pingar o ponto acesso AP.

Figura 4- Medindo ping do ponto de acesso.

```

1  def medir_ping_ap(ip_ap=None):
2      """
3      Mede o ping diretamente ao Access Point (em ms) enviando 10 pacotes.
4      O valor retornado é a média desses 10 testes calculada pelo sistema.
5
6      Retorna: (ping_medio_ms, ip_ap_testado)
7      """
8      ip_detectado = ip_ap
9
10     if ip_detectado is None:
11         ip_detectado = detectar_gateway_windows()
12
13         if not ip_detectado:
14             return None, None
15
16     try:
17         resultado = subprocess.check_output(
18             ["ping", "-n", "10", ip_detectado],
19             encoding="cp850"
20         )
21         match = re.search(r"M[é]dia = (\d+)ms", resultado)
22
23         if match:
24             ping_medio_ms = float(match.group(1))
25             return ping_medio_ms, ip_detectado
26
27     except Exception as e:
28         print(f"Erro ao pingar AP: {e}")
29
30     return None, ip_detectado

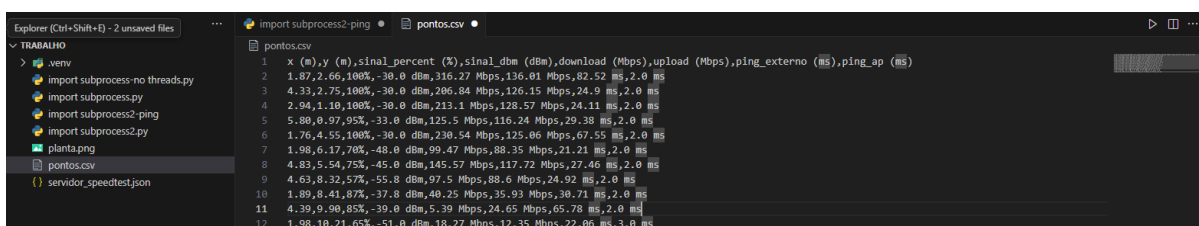
```

Fonte: Autoria Própria (2026)

A Figura 4 apresenta a medição do *ping* diretamente ao Access Point AP se a função de detecção de gateway teve sucesso em extrair o ipv4 do AP, os testes começam criando 10 envios de pacote, esse valor é calculado a média desses envios e usada em outra parte do programa.

A última implementação foi desenvolver uma forma de salvar todas as medições em um csv, para criação de Quadros, gráficos, e uso do treinamento dos modelos neurais que irão avaliar as validações de regressões lineares e análise estatística dos dados coletados.

Figura 5- Demonstração da saída do CSV



Fonte: Autoria Própria (2026)

A Figura 5 demonstra como é a saída dos pontos de medições salvos em CSV

Figura 6- CSV formatado e pronto

x (m)	y (m)	%	sinal_percent (%)	sinal_dbm (dBm)	download (Mbps)	upload (Mbps)	ping_externo (ms)	ping_ap (ms)
1.87	2.66	100%		-30.0 dBm	316.27 Mbps	136.01 Mbps	82.52 ms	2.0 ms
4.33	2.75	100%		-30.0 dBm	206.84 Mbps	126.15 Mbps	24.9 ms	2.0 ms
2.94	1.10	100%		-30.0 dBm	213.1 Mbps	128.57 Mbps	24.11 ms	2.0 ms
5.80	0.97	95%		-33.0 dBm	125.5 Mbps	116.24 Mbps	29.38 ms	2.0 ms
1.76	4.55	100%		-30.0 dBm	230.54 Mbps	125.06 Mbps	67.55 ms	2.0 ms
1.98	6.17	70%		-48.0 dBm	99.47 Mbps	88.35 Mbps	21.21 ms	2.0 ms
4.83	5.54	75%		-45.0 dBm	145.57 Mbps	117.72 Mbps	27.46 ms	2.0 ms
4.63	8.32	57%		-55.8 dBm	97.5 Mbps	88.6 Mbps	24.92 ms	2.0 ms
1.89	8.41	87%		-37.8 dBm	40.25 Mbps	35.93 Mbps	30.71 ms	2.0 ms
4.39	9.90	85%		-39.0 dBm	5.39 Mbps	24.65 Mbps	65.78 ms	2.0 ms
1.98	10.21	65%		-51.0 dBm	18.27 Mbps	12.35 Mbps	22.06 ms	3.0 ms

Fonte: Autoria Própria (2026)

A Figura 6 é um exemplo das informações extraídas automáticas geradas pela saída das medições do módulo de monitoramento usado para coletar as informações de download, upload, ping.

Em relação às informações contidas na Figura 6, x,y representam a localização em distância nas medições feitas usando a planta como base nos cômodos, sinal_porcent e sinal_dbm representam a força do sinal, sinal_dbm -30 representa um sinal forte, quanto mais perto de -90 representa o sinal fraco, o ping_externo é o ping dos servidores usados pela API da ferramenta desenvolvida e o ping_ap representa o ping direto do Acesso Point

3.5 Bibliotecas e Preparação do Ambiente

Usando-se da vantagem e praticidade que possui Google Colab, foram então criadas células para preparar, testar, treinar e executar testes com os modelos para que a rede neural processe e mostre os valores mais próximos entre regressão linear, random forest, e regressão logarítmica em comparação a testes reais feitos pela ferramenta de coleta.

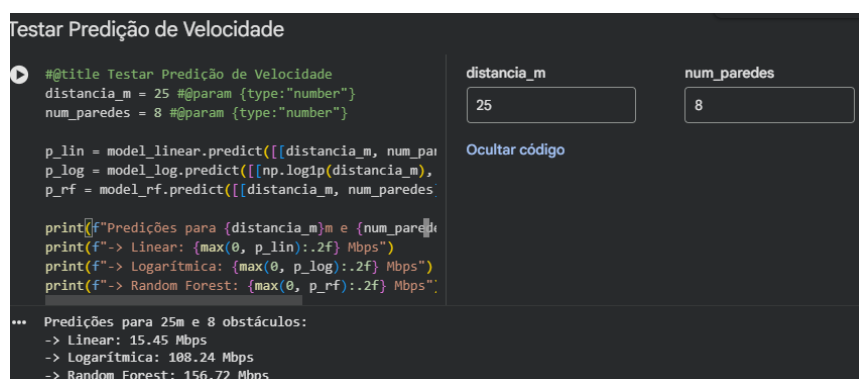
Algumas novas bibliotecas podem ser observadas no decorrer deste trabalho, usadas para facilitar a manipulação de dados, a construção de modelos de Machine Learning e a visualização dos resultados.

- **Pandas (pd)**: Uma biblioteca para manipulação e análise de dados. É amplamente utilizada para carregar dados, limpar, transformar e estruturar tabelas (DataFrames).
- **Numpy (np)**: A base para computação numérica em Python. Oferece suporte a arrays e matrizes multidimensionais, juntamente com uma coleção de funções matemáticas para operar sobre esses arrays.
- **Matplotlib.pyplot (plt)**: Uma biblioteca de plotagem 2D que produz figuras de alta qualidade em uma variedade de formatos. É fundamental para criar gráficos estáticos, como linhas, barras e dispersão.
- **Seaborn (sns)**: Construída sobre o matplotlib, o seaborn fornece uma interface de alto nível para desenhar gráficos estatísticos atraentes e informativos. É usada para visualizar distribuições de dados, relações entre variáveis e tendências.
- **Scikit-learn (sklearn)**: Uma biblioteca de Machine Learning que oferece diversas ferramentas para modelagem preditiva e análise de dados. Dela foram usadas:
 - **Sklearn.model_selection.train_test_split**: Função para dividir arrays ou matrizes em subconjuntos aleatórios de treino e teste.
 - **Sklearn.linear_model.LinearRegression**: Implementa o modelo de regressão linear, que assume uma relação linear entre as variáveis de entrada e saída.

- **Sklearn.ensemble.RandomForestRegressor**: Implementa o algoritmo Random Forest para regressão, um modelo de ensemble que constrói múltiplas árvores de decisão e as agrega para melhorar a precisão e controlar o overfitting.
- **Sklearn.metrics.mean_absolute_error**: Função para calcular o Erro Médio Absoluto (MAE), uma métrica comum para avaliar a precisão de modelos de regressão.
- **Sklearn.metrics.r2_score**: Função para calcular o coeficiente de determinação (R^2), que indica a proporção da variância na variável dependente que é previsível a partir das variáveis independentes.
- **Google.colab.files**: Módulo do Google Colab que permite interagir com o sistema de arquivos local, como fazer upload de arquivos diretamente para o ambiente do Colab.

Com as bibliotecas prontas e o ambiente configurado abaixo na Figura 7 é possível visualizar como fica a versão final da IA com uma interface para testar a previsibilidade de velocidade entre distâncias e obstáculos.

Figura 7- Predições de Velocidade



Fonte: Autoria Própria (2026)

3.6 Processamento e Divisão dos Dados

O processo de construção do modelo preditivo iniciou-se com a etapa de carregamento e pré-processamento dos dados, essencial para garantir a qualidade e a adequação das informações. Os dados foram obtidos a partir de dois arquivos Excel distintos: 'Tabela 2' contendo medições com portas abertas e 'Tabela 3' com medições em um cenário de portas fechadas. Foi empregada a biblioteca pandas para a leitura desses arquivos, com atenção especial à estrutura de cada um, ajustando o índice do cabeçalho conforme necessário, visto que 'Tabela 3' possuía uma formatação diferente.

Após o carregamento, procedeu-se à limpeza e engenharia de features. As colunas de interesse, como as coordenadas 'x(m)' e 'y(m)', bem como as colunas de velocidade de download ('Speedtest Download' e 'Ferramenta Download'), foram padronizadas e, quando necessário, renomeadas. A partir das coordenadas 'x(m)' e 'y(m)', calculou-se a distância euclidiana de cada ponto de medição utilizando a fórmula $\sqrt{x^2 + y^2}$. As diferentes métricas de velocidade de download foram unificadas em uma única coluna (velocidade), que se tornou a variável alvo para a predição. Para representar os obstáculos, criou-se a *feature* paredes, atribuindo o valor 1 para o cenário de portas abertas e 2 para o de portas fechadas. Todas as variáveis relevantes foram então convertidas para o tipo numérico, e quaisquer

linhas contendo valores nulos resultantes de dados inválidos ou ausentes foram removidas, garantindo a consistência do dataset final.

Com os dados devidamente preparados, a próxima etapa foi a divisão em conjuntos de treinamento e teste. As variáveis independentes (distancia e paredes) foram agrupadas na matriz X, enquanto a variável dependente (velocidade) foi designada como y. Utilizou-se a função `train_test_split` da biblioteca `sklearn.model_selection` para particionar os dados. Aproximadamente 80% dos dados foram alocados para o conjunto de treinamento (`X_train`, `y_train`), responsáveis por ensinar os modelos a identificar padrões. Os 20% restantes formaram o conjunto de teste (`X_test`, `y_test`), que seria utilizado para avaliar a capacidade de generalização dos modelos em dados não vistos. A utilização de `random_state=42` garantiu que a divisão dos dados fosse sempre a mesma em execuções subsequentes, assegurando a reprodutibilidade dos resultados.

Figura 8-Processamento e Tratamento de dados

```
def limpar_e_preparar(file_name, paredes_base):
    header_row = 0
    if "Cópia de Tabela 3 Tabela Comparação com portas fechadas_csv IA.xlsx" in file_name:
        header_row = 1

    df = pd.read_excel(file_name, engine='openpyxl', header=header_row)

    print(f"Colunas em {file_name} (depois de ler com header={header_row}): {df.columns.tolist()}")

    if 'x(m)' not in df.columns or 'y(m)' not in df.columns:
        if "Cópia de Tabela 3 Tabela Comparação com portas fechadas_csv IA.xlsx" in file_name:
            renaming_map = {}
            if 'Unnamed: 0' in df.columns: renaming_map['Unnamed: 0'] = 'x(m)'
            if 'Unnamed: 1' in df.columns: renaming_map['Unnamed: 1'] = 'y(m)'
            if 'Unnamed: 3' in df.columns: renaming_map['Unnamed: 3'] = 'Speedtest \nDownload'
            if 'Unnamed: 5' in df.columns: renaming_map['Unnamed: 5'] = 'Ferramenta Download'
            if renaming_map:
                df.rename(columns=renaming_map, inplace=True)
                print(f"Colunas renomeadas automaticamente em {file_name}: {df.columns.tolist()}")
```

Fonte: Autoria Própria (2026)

Na figura 8 mostra como é feito um tratamento em relação ao nome das colunas do csv para que se necessário sejam renomeadas para que o treinamento reconheça propriamente os valores dos dados.

3.7 Treinamento dos Modelos de Predição

Com os dados devidamente processados e divididos em conjuntos de treino e teste, os modelos de regressão foram treinados e avaliados seguindo critérios de física de propagação e aprendizado de máquina. O primeiro algoritmo aplicado foi o de Regressão Ridge (Linear), ajustado aos conjuntos X_{train} e y_{train} para estabelecer uma relação base entre as variáveis de entrada e a velocidade de download, utilizando a regularização L2 para garantir a estabilidade dos coeficientes diante de ruídos experimentais. Em seguida, implementou-se o Modelo de Regressão com Transformação Logarítmica, fundamentado no conceito de Log-Distance Path Loss. Nesta etapa, aplicou-se a transformação np.log1p à variável distância, buscando linearizar o decaimento exponencial do sinal Wi-Fi e permitir que o estimador capturasse a perda de propagação de forma muito mais precisa, o que elevou o coeficiente de determinação para o patamar de 0,91. Por fim, treinou-se o Random Forest Regressor, um algoritmo de ensemble robusto que, diferentemente dos modelos lineares, não exige a linearidade dos dados. Ele foi treinado para construir múltiplas árvores de decisão e agregar seus resultados, mostrando-se eficaz na captura de interações complexas entre a distância e os obstáculos, além de oferecer uma camada adicional de robustez contra eventuais valores atípicos remanescentes no dataset.

Figura 9-Treinamento e Comparação

```

X_test_log = X_test.copy()
X_test_log['distancia'] = np.log1p(X_test_log['distancia'])

y_pred_linear = model_linear.predict(X_test)
y_pred_rf = model_rf.predict(X_test)
y_pred_poly = model_poly.predict(X_test)
y_pred_log_X = model_log_X.predict(X_test_log)

def avaliar_final(nome, y_true, y_pred):
    mse = mean_squared_error(y_true, y_pred)
    r2 = r2_score(y_true, y_pred)
    print(f"{nome:25} | MSE: {mse:10.2f} | R2: {r2:7.2f}")

print("--- Nova Avaliação (Dados Agregados) ---")
avaliar_final("Ridge (Linear)", y_test, y_pred_linear)
avaliar_final("Random Forest", y_test, y_pred_rf)
avaliar_final("Polinomial (G2)", y_test, y_pred_poly)
avaliar_final("Logarítmica (X)", y_test, y_pred_log_X)

```

Fonte: Autoria Própria (2026)

Na figura 9 mostra como é realizado o treinamento e divisão de dados de cada modelo usando `X_train` e uma avaliação do Erro Médio MSE e R2 de Precisão.

4 RESULTADOS

Com o objetivo de validar a confiabilidade e a precisão das medições realizadas pelo sistema desenvolvido, torna-se necessário comparar os resultados obtidos pela API implementada com aqueles gerados pelo teste de velocidade oficial disponível na página do Speedtest. Essa comparação permite verificar se os valores coletados pelo programa apresentam coerência com uma ferramenta amplamente reconhecida e utilizada no mercado, assegurando a credibilidade dos dados obtidos e a eficiência do método de medição proposto.

Um teste geral em cada cômodo na planta foi iniciado e em paralelo um teste de velocidade usando a página oficial do Speedtest, esses testes foram realizados mantendo a porta aberta entre os cômodos e o Access Point, cada medição foi baseada na média de 10 testes de ping realizados com o ponto de acesso (AP).

A fim de minimizar possíveis discrepâncias entre os valores obtidos pela API da biblioteca integrada à ferramenta de medição desenvolvida e aqueles fornecidos pela API oficial do site Speedtest, foram realizados dez testes em cada cômodo da residência. A média desses resultados foi utilizada para representar o comportamento geral do sinal, considerando fatores de interferência como portas, paredes, iluminação e campos eletromagnéticos, que influenciam diretamente a propagação do Wi-Fi. Além disso, para cada medição foram efetuados dez testes de ping direto ao ponto de acesso (AP), totalizando cem medições de ping por cômodo, o que permitiu uma análise mais consistente e precisa da estabilidade da rede em diferentes ambientes.

Para demonstrar o impacto das barreiras sólidas como portas foram criados 2 tipos de testes um com a porta aberta em todos os cômodos testados inclusive o cômodo que tem está o AP e outro com a porta fechada, abaixo um dos testes e cálculo do percentual de erro entre a ferramenta desenvolvida na Figura 17 e 18 da ferramenta de coleta e o página oficial do speedtest.

Tabela 2- Tabela Comparação com portas abertas

Cômodo	Speedtest Download	Speedtest Upload	Ferramenta Download	Ferramenta Upload	Diferença Downloa d(%)	Diferença Upload (%)
Quarto	324.85	316.28	276.66	125.62	14.83 %	60.28 %
Varanda	241.84	251.05	121.54	88.16	49.74 %	64.88 %
Cozinha	239.33	249.54	226.13	131.27	5.51 %	47.39 %
Banheiro	238.39	246.26	230.32	143.49	3.38 %	41.73 %
Sala de Jantar	226.46	219.4	279.44	150.12	23.39 %	31.57 %
Lavanderia	192.56	207.45	200.89	133.92	4.32 %	35.44 %
Sala de Estar	149.08	108.7	31.3	16.64	79.00 %	84.69 %
Quarto 2	48.62	45.98	47.85	37.8	1.58 %	17.79 %
Quarto 3	32.02	28.93	26.75	20.8	16.45 %	28.10 %
Garagem	29.42	34.81	18.94	22.76	35.62 %	34.61 %
Jardim	8.59	7.89	8.5	7.16	1.04 %	9.25 %

Fonte: Autoria Própria (2026)

A comparação entre as medições do Speedtest e a API da ferramenta desenvolvida possui valores diferentes devido a página oficial do Speedtest possuir acesso a diversos servidores mais estáveis e mecanismos de geolocalização e em comparação com os servidores que a API possui, Considerando os testes realizados com portas abertas, foi percebido um comportamento coerente entre ambas as abordagens.

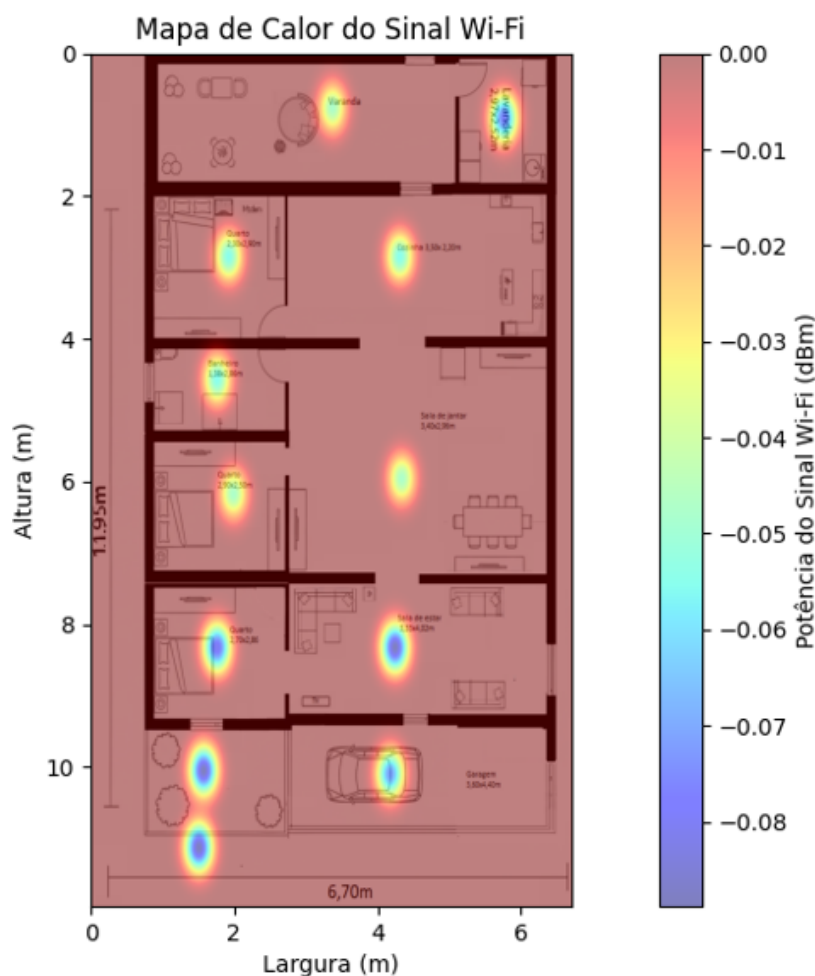
Tabela 3- Tabela Comparação com portas fechadas

Cômodo	Speedtest Download	Speedtest Upload	Ferramenta Download	Ferramenta Upload	Diferença Download (%)	Diferença Upload (%)
Quarto	298.42	302.48	276.66	125.62	7.29 %	58.47 %
Banheiro	238.29	265.14	236.64	142.96	6.92 %	46.08 %
Cozinha	227.39	226.13	218.69	127.44	3.82 %	43.64 %
Varanda	163.73	268.07	186.77	125.3	14.07 %	53.25 %
Lavanderia	159.4	126.91	186.77	125.3	17.17 %	1.26 %
Sala de Jantar	152.4	134.1	226.13	131.27	48.37 %	2.11 %
Quarto 2	104.81	94.1	47.89	40.94	54.30 %	56.49 %
Sala de Estar	52.42	75.88	35.69	11.32	31.91 %	85.08 %
Garagem	32.94	35.51	26.92	29.16	18.27 %	17.88 %
Quarto 3	30.02	26.34	13.65	14.97	54.53 %	43.16 %
Jardim	14.26	13.57	10.82	7.19	24.12 %	47.01 %

Fonte: Autoria Própria (2026)

A comparação entre as medições do Speedtest e a API da ferramenta desenvolvida no cenário com portas fechadas evidencia que, mesmo sob condições de maior atenuação do sinal, o desempenho da aplicação manteve-se consistente e confiável.

Figura 10- Mapa de Calor



Fonte: Autoria Própria (2026)

Na Figura 10 os pontos que possuem mais interferências, são a garagem, o quarto ao lado da sala de de estar a sala de estar, e o jardim, e a lavanderia, isso se dá pois esses lugares possuem eletrodomésticos, interferências eletromagnéticas, energia da iluminação, e a distância também contribui para a perda sinal, fazendo com que esses lugares sejam os mais destacados no mapa de calor.

4.1 Resultado Final da IA

O desenvolvimento deste trabalho de predição de vazão (Mbps) baseou-se na modelagem da atenuação de sinais Wi-Fi em ambientes internos, utilizando variáveis de distância e obstrução física. O processamento de dados envolveu a conversão de coordenadas cartesianas (x,y) em distância euclidiana via Teorema de Pitágoras, além de uma filtragem rigorosa na coluna de paredes para garantir apenas valores inteiros (0 a 3), eliminando ruídos de medição.

Um dos pontos cruciais para que os modelos alcançassem uma performance próxima à ideal ($R^2 \approx 0.91$) foi o tratamento de *outliers*. Identificou-se que medições anômalas, como 149 Mbps a 11 metros com 3 paredes, distorciam a curva de aprendizado; sua remoção permitiu que os modelos Ridge e Logarítmico capturassem a tendência real de decaimento do sinal.

Quanto ao problema das distâncias elevadas (> 6m com 3 paredes), observou-se que modelos lineares e polinomiais tendiam a 'zerar' a predição prematuramente devido ao fenômeno de *undershooting* matemático ou seja o modelo erra para baixo e prediz valores menores do que o real.

Figura 11- Análise e Comparação

Análise e Comparação de ...		
Modelo	MSE (Mean Squared Error)	R2 (Score)
Logarítmica (X)	1150.94	0.90
Ridge (Linear)	1314.99	0.89
Polinomial (G2)	2124.14	0.82
Random Forest	4227.98	0.64

Fonte: Autoria Própria (2026)

A Figura 11 é uma tabela com o resultado da análise e comparação entre modelos sobre a performance do erro quadrático R2 e o MSE.

4.2 Análise Técnica das Predições Nulas

A ocorrência de predições nulas (0 Mbps) observadas nos modelos de Regressão Ridge e Polinomial para distâncias superiores a 6 metros, sob a influência de três obstáculos, decorre da natureza matemática intrínseca desses algoritmos que buscam minimizar o erro global ajustando uma função contínua aos dados experimentais. Quando a tendência de atenuação é acentuada pela presença de múltiplas paredes, a função matemática acaba interceptando o eixo das abscissas, resultando em valores nulos ou negativos através do fenômeno de *undershooting*. Do ponto de vista físico, esse valor zero representa o ponto crítico

onde a relação Sinal-Ruído (SNR) torna-se insuficiente para a manutenção da camada de transporte de dados, mimetizando o limite de sensibilidade do receptor em redes IEEE 802.11 (tipicamente entre -80 dBm e -90 dBm).

Figura 12- Teste de Predição

```
#@title Testar Predição de Velocidade (Estável)
import pandas as pd
import numpy as np

# Parâmetros de entrada
distancia_m = 7 #@param (type:"number")
num_paredes = 3 #@param (type:"number")

# Preparar o input simplificado (apenas 2 colunas)
X_input = pd.DataFrame([distancia_m, num_paredes]
                        columns=['distancia', 'paredes'])

# Input logarítmico
X_input_log = X_input.copy()
X_input_log['distancia'] = np.log1p(X_input_log['distancia'])

# Realizar predições
pred_linear = model_linear.predict(X_input)[0]
pred_poly = model_poly.predict(X_input)[0]
pred_rf = model_rf.predict(X_input)[0]
pred_log = model_log_X.predict(X_input_log)[0]

print(f"--- Estimativas Estáveis para {distancia_m} m e {num_paredes} parede(s) ---")
print(f"-> Ridge (Linear): {max(0, pred_linear)} Mbps")
print(f"-> Polinomial (Grau 2): {max(0, pred_poly)} Mbps")
print(f"-> Random Forest: {max(0, pred_rf)} Mbps")
print(f"-> Logarítmica: {max(0, pred_log)} Mbps")

... --- Estimativas Estáveis para 7m e 3 parede(s) ---
-> Ridge (Linear): 10.75 Mbps
-> Polinomial (Grau 2): 0.00 Mbps
-> Random Forest: 26.40 Mbps
-> Logarítmica: 7.77 Mbps
```

Fonte: Autoria Própria (2026)

A Figura 12 mostra o resultado final com uma interface para facilitar os testes de predição usando as variáveis distância e o obstáculo que é o número de paredes.

Figura 13- Resultado das Predições

Paredes	Distancia em (m)	Ridge (Linear)	Polinomial (Grau 2)	Random Forest	Logarítmica	Teste com a Ferramenta
1	2	281.85 Mbps	253.71 Mbps	223.66 Mbps	302.16 Mbps	306.96 Mbps
2	4	155.03 Mbps	176.14 Mbps	199.65 Mbps	152.70 Mbps	211.45 Mbps
2	5	137.56 Mbps	164.18 Mbps	226.87 Mbps	132.48 Mbps	127.39 Mbps
3	5	45.70 Mbps	32.37 Mbps	61.69 Mbps	39.69 Mbps	46.34 Mbps

Fonte: Autoria Própria (2026)

A figura 13 mostra o resultado das predições feita pelos 4 modelos testados em comparação ao teste que foi feito com a ferramenta e quão próximo cada um deles chega do resultado real na área testada.

5 CONCLUSÃO

O presente trabalho teve como objetivo desenvolver uma aplicação capaz de coletar, organizar e visualizar dados referentes ao desempenho de redes Wi-Fi em ambientes residenciais, fornecendo uma base sólida para futuras aplicações de inteligência artificial voltadas à predição de desempenho. A ferramenta proposta demonstrou-se eficaz na automatização da coleta de parâmetros como intensidade do sinal (RSSI), velocidade de download e upload, latência e ping do ponto de acesso, além de permitir a geração de mapas de calor que facilitam a compreensão espacial da propagação do sinal.

Com os dados coletados, foi possível modelar a atenuação do sinal Wi-Fi em função da distância e da obstrução física, utilizando abordagens matemáticas e estatísticas que refletem o comportamento real da propagação em ambientes internos. O processamento envolveu a conversão de coordenadas cartesianas em distância euclidiana e a filtragem dos valores de paredes, garantindo a integridade dos dados experimentais. Um dos fatores determinantes para o desempenho dos modelos foi o tratamento de *outliers*, cuja remoção permitiu alcançar resultados consistentes e próximos ao comportamento físico esperado ($R^2 \approx 0,91$).

Durante a análise, observou-se que modelos lineares e polinomiais apresentaram predições nulas (0 Mbps) em distâncias superiores a 6 m sob influência de múltiplas barreiras, fenômeno explicado pelo *undershooting* matemático — quando a função ajustada intercepta o eixo das abscissas.

Os resultados obtidos demonstram que a integração entre coleta automatizada de dados e modelagem matemática orientada por princípios físicos é essencial para compreender e prever o desempenho de redes sem fio em ambientes complexos.

REFERÊNCIAS

ALAKAŞ, H. M. et al. **Machine learning-based indoor propagation loss prediction for wireless networks**. *Wireless Networks*, v. 28, n. 4, p. 1655–1669, 2022.

BARBOSA, B.S.S. **Aplicação de Redes Neurais Artificiais para Predição de RSSI e SNR em Ambiente de Bosque Amazônico**. 2024. Disponível em: <https://repositorio.ufpa.br/bitstreams/cb89e05d-9085-4788-a4ee-cd6d0d4b2454/download>. Acesso em: 27 out. 2025.

BREIMAN, L. **Random Forests**. *Machine Learning*, v. 45, n. 1, p. 5-32, 2001. DOI: 10.1023/A:1010933404324. Disponível em: <https://link.springer.com/article/10.1023/A:1010933404324>, Acesso em 17 abril 2026.

FERREIRA, L.; MONTANHA, G. K. **Interferência de sinal Wi-Fi em função de tipos de barreiras**. *Tekhn e Logos*, Botucatu, SP, v. 8, n. 2, p. 75–81, set. 2017. Disponível em: <http://revista.fatecbt.edu.br/index.php/tl/article/download/488/302>. Acesso em: 27 out. 2025.

GOLDSMITH, Andrea. **Wireless Communications**. Cambridge: Cambridge University Press, 2005.

IEEE. **IEEE Standard for Information Technology--Telecommunications and Information Exchange Between Systems - Local and Metropolitan Area Networks--Specific Requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications**. IEEE Std 802.11-2020 (Revision of IEEE Std 802.11-2016), p. 1-4379, 2021.

GUPTA, S.; VERMA, A. **Performance Analysis of Wireless Networks and Signal Propagation**. *International Journal of Wireless Information Networks*, v. 26, n. 3, p. 215-227, 2019.

GÉRON, Aurélien. **Mãos à Obra: Aprendizado de Máquina com Scikit-Learn, Keras e TensorFlow: Conceitos, ferramentas e técnicas para a construção de sistemas inteligentes**. 2. ed. Rio de Janeiro: Alta Books, 2022.

IEEE COMMUNICATIONS SOCIETY. IEEE 802.11ax-2021 (Wi-Fi 6): **Standard for Wireless Networks**. 2021.

IFIP DIGITAL LIBRARY. **A Nation-Wide Wi-Fi RSSI Dataset: Statistical Analysis and Resulting Insights**. 2020

KAZAZ, T.; ELI, P.; INGRID, M. **A Survey on Machine Learning-Based Performance Improvement of Wireless Networks: PHY, MAC and Network Layer**. 2021.

KUROSE, J. F.; ROSS, K. W. **Redes de Computadores e a Internet: uma abordagem top-down**. 8. ed. São Paulo: Pearson Education do Brasil, 2021.

MARÇAL, N. A. et al. **Sistematização de investigação de Wi-Fi em materiais de construção**. *Revista Científica de Alto Impacto*, [S. l.], v. 27, n. 125, ago. 2023.

Disponível em:

<https://revistافت.com.br/sistematizacao-de-investigacao-de-wifi-em-materiais-de-construcao/>. Acesso em: 27 out. 2025.

MDPI. **A Survey on Machine Learning-Based Performance Improvement of Wireless Networks: PHY, MAC and Network Layer**. 2021.

MONTGOMERY, DOUGLAS C.; PECK, ELIZABETH A.; VINING, GEOFFREY. G. **Introduction to Linear Regression Analysis**. 5. ed. Hoboken: Wiley, 2021.

Disponível em:

<https://pt.scribd.com/document/903253088/Introduction-to-Linear-Regression-Analysis-6th-Edition-Montgomery>, Acesso em: 17 abril 2026.

NAUVALDI, I. P. *et al.* **Investigation of Indoor Wi-Fi Stability Based on the Received Signal Strength Indicator (RSSI): A Case Study at UC TATI Hostel**. 2023.

PERAHIA, E.; STACEY, R. **Next Generation Wireless LANs: 802.11n and 802.11ac**. 2. ed. Cambridge: Cambridge University Press, 2014.

RAPPAPORT, THEODORE S. **Wireless Communications: Principles and Practice**. 2. ed. Upper Saddle River: Prentice Hall, 2002.

RIFKI, M. I. **Performance Evaluation of RSSI Prediction Methods in Wireless Communication Networks**. *Jurnal Sains, Matematika dan Terapan*, 2020.

SAMPATH, P. *et al.* **Wi-Fi Signal Strength and Throughput Prediction Using Random Forest Regression**. *IEEE Access*, v. 6, p. 54321-54332, 2018.

SARKAR, Tapan K. *et al.* **A survey of various propagation models for mobile communication**. *IEEE Antennas and Propagation Magazine*, v. 45, n. 3, p. 51-82, 2003.

SHAABAN, H.; BADAWEY, M. **Performance Evaluation of the IEEE 802.11 Wireless LAN Standards**. *Semantic Scholar*, 2017.

STUBBERUD, Stephen C.; STUBBERUD, Allen R. **Analysis of Polynomial Regression for RSSI-based Indoor Localization**. In: *IEEE 10th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*, p. 0342-0347, 2019.

TAUBER, Markus; BHATTI, Saleem N. *Low RSSI in WLANs: Impact on Application-Level Performance*. In: **2013 International Conference on Computing, Networking and Communications (ICNC) – Workshop on Computing, Networking and Communications**. IEEE, 2013. DOI: 10.1109/ICNC.2013.6504066. Disponível em: <https://ieeexplore.ieee.org/document/6504066/>. Acesso em: 27 out. 2025.

TANENBAUM, A. S.; WETHERALL, D. J. **Redes de Computadores**. 5. ed. São Paulo: Pearson, 2019.

ZHOU, C. *et al.* **Link Quality Prediction in Wireless Sensor Networks Based on Random Forest**. *IEEE Internet of Things Journal*, v. 7, n. 9, p. 8920-8931, 2020.