

# PREVISÃO E MONITORAMENTO DE ALAGAMENTOS URBANOS NO RECIFE COM APOIO DE INTELIGÊNCIA ARTIFICIAL

MARIA EDUARDA SANTOS SILVA

ORIENTADOR:  
Nome do Orientador

INSTITUTO FEDERAL DE PERNAMBUCO  
CURSO DE ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

RECIFE  
2025

# PREVISÃO E MONITORAMENTO DE ALAGAMENTOS URBANOS NO RECIFE COM APOIO DE INTELIGÊNCIA ARTIFICIAL

Maria Eduarda Santos Silva

Trabalho de Conclusão de Curso apresentado ao curso de Maria Eduarda Santos Silva da Instituto Federal de Pernambuco, como requisito parcial para obtenção do título de Tecnólogo.

Orientador: NOME COMPLETO DO ORIENTADOR

RECIFE  
2025

# Dedicatória

Dedico este trabalho, antes de tudo, aos meus **pais**, que sempre foram meu chão e meu norte. Pelo amor que nunca faltou, pela paciência nos dias difíceis e por me mostrarem, com o exemplo, o valor do esforço e da honestidade. Tudo o que sou carrego em grande parte de vocês.

Aos meus **mestres e orientadores**, que, com generosidade e dedicação, compartilharam conhecimento e apontaram caminhos quando a dúvida parecia maior que a coragem. Cada conselho, cada gesto de apoio, foi um empurrão essencial nesta jornada.

E dedico também a todos que acreditam no **poder da educação e da tecnologia** para transformar vidas. Que este trabalho seja, ainda que pequeno, uma semente nesse caminho de mudança, esperança e futuro melhor.

# Agradecimentos

Este trabalho é fruto de muito esforço, dedicação e persistência pessoal. Cada etapa foi construída com empenho próprio, enfrentando desafios e aprendendo com eles.

Ainda assim, não posso deixar de registrar minha gratidão à minha família, pelo apoio silencioso e pela compreensão nos momentos de ausência e cansaço.

Agradeço também a todos que, de alguma forma, torceram por mim e acreditaram que eu seria capaz de chegar até aqui.

Este TCC é, acima de tudo, resultado de uma caminhada solitária, mas que se tornou possível graças à força que encontrei dentro de mim e ao incentivo das pessoas mais próximas.

*“Prever um alagamento não é apenas antecipar a água que vai cair do céu, mas proteger vidas que estão no caminho dela.”*

**Maria Eduarda Santos Silva**

**Recife, 29 de setembro de 2025**

# Resumo

Este trabalho apresenta o desenvolvimento de um sistema inteligente de previsão e monitoramento de alagamentos urbanos em tempo real, com aplicação específica na cidade do Recife. A pesquisa aborda a problemática dos alagamentos que afetam significativamente a qualidade de vida urbana, causando prejuízos econômicos e sociais. O sistema proposto integra dados meteorológicos em tempo real da Agência Pernambucana de Águas e Clima (APAC) com histórico de ocorrências da Defesa Civil, utilizando algoritmos de análise espacial e temporal para gerar alertas preventivos. A arquitetura do sistema foi desenvolvida em Python com framework FastAPI, interface web responsiva com Leaflet.js, e implementa um modelo de previsão baseado em múltiplos fatores de risco. Os resultados demonstram a eficácia do sistema na identificação precoce de áreas críticas, oferecendo uma ferramenta valiosa para gestão de riscos urbanos e proteção civil.

**Palavras-chave:** Alagamentos Urbanos. Previsão em Tempo Real. Sistemas Inteligentes. Recife. Gestão de Riscos.

## **Resumo**

This paper presents the development of an intelligent system for real-time prediction and monitoring of urban floods, with specific application in the city of Recife, Brazil. The research addresses the problem of urban flooding that significantly affects quality of life, causing economic and social damages. The proposed system integrates real-time meteorological data from the Pernambuco Water and Climate Agency (APAC) with historical records from Civil Defense, using spatial and temporal analysis algorithms to generate preventive alerts. The system architecture was developed in Python with FastAPI framework, responsive web interface with Leaflet.js, and implements a prediction model based on multiple risk factors. The results demonstrate the system's effectiveness in early identification of critical areas, offering a valuable tool for urban risk management and civil protection.

**Keywords:** Urban Flooding. Real-Time Prediction. Intelligent Systems. Recife. Risk Management.

# Lista de ilustrações

Figura 1 – Arquitetura detalhada do sistema de monitoramento . . . . .	24
Figura 2 – Mapa interativo exibindo áreas de risco no Recife . . . . .	28
Figura 3 – Detalhes da região selecionada . . . . .	29

# Lista de tabelas

Tabela 1 – Métricas de Desempenho Detalhadas por Classe . . . . .	21
Tabela 2 – Teste t para Comparação com Baseline . . . . .	21
Tabela 3 – Requisitos Funcionais do Sistema . . . . .	22
Tabela 4 – Requisitos Não Funcionais do Sistema . . . . .	23
Tabela 5 – Áreas de Risco Mapeadas no Recife . . . . .	28



# Lista de abreviaturas e siglas

APAC	Agência Pernambucana de Águas e Clima
API	Application Programming Interface
GIS	Geographic Information System
HTTP	Hypertext Transfer Protocol
JSON	JavaScript Object Notation
ML	Machine Learning
TCC	Trabalho de Conclusão de Curso

# Sumário

	<b>Dedicatória . . . . .</b>	<b>3</b>
	<b>Agradecimentos . . . . .</b>	<b>4</b>
<b>1</b>	<b>INTRODUÇÃO . . . . .</b>	<b>12</b>
1.1	Contextualização do Problema . . . . .	12
1.2	Problema de Pesquisa . . . . .	12
1.3	Objetivos . . . . .	12
1.3.1	Objetivo Geral . . . . .	12
1.3.2	Objetivos Específicos . . . . .	13
<b>2</b>	<b>REFERENCIAL TEÓRICO . . . . .</b>	<b>14</b>
2.1	Alagamentos Urbanos: Conceitos e Impactos . . . . .	14
2.2	Sistemas de Monitoramento Hidrometeorológico . . . . .	14
2.3	Tecnologias para Previsão de Desastres Naturais . . . . .	15
2.4	Estudos de Caso em Cidades Brasileiras . . . . .	15
2.5	Fundamentos de Engenharia de Software Aplicada . . . . .	15
<b>3</b>	<b>METODOLOGIA: ALGORITMOS, CÁLCULOS E TRATAMENTO DE DADOS . . . . .</b>	<b>16</b>
3.1	Abordagem Computacional e Matemática do Sistema . . . . .	16
3.1.1	Modelo de Risco . . . . .	16
3.1.2	Normalização e Transformações . . . . .	16
3.1.3	Classificação Fuzzy . . . . .	17
3.2	Pipeline de Processamento de Dados . . . . .	17
3.3	Integração Geoespacial . . . . .	17
3.4	Validação do Modelo . . . . .	17
3.5	Implementação Computacional . . . . .	18
3.5.1	Arquitetura de Software . . . . .	18
3.5.2	Otimização de Performance . . . . .	19
3.6	Resultados da Validação Estatística . . . . .	20
3.6.1	Desempenho por Categoria de Risco . . . . .	20
3.6.2	Análise de Significância Estatística . . . . .	21
3.6.3	Análise de Robustez . . . . .	21
<b>4</b>	<b>DESENVOLVIMENTO DO SISTEMA . . . . .</b>	<b>22</b>
4.1	Análise e Especificação de Requisitos . . . . .	22

4.1.1	Requisitos Funcionais . . . . .	22
4.1.2	Requisitos Não Funcionais . . . . .	22
<b>4.2</b>	<b>Arquitetura do Sistema . . . . .</b>	<b>23</b>
4.2.1	Visão Geral da Arquitetura . . . . .	23
4.2.2	Padrões Arquiteturais Aplicados . . . . .	25
<b>4.3</b>	<b>Implementação Técnica . . . . .</b>	<b>25</b>
4.3.1	Estrutura de Diretórios e Organização do Código . . . . .	25
4.3.2	Integração com a API da APAC . . . . .	26
<b>4.4</b>	<b>Estudo de Caso: Recife . . . . .</b>	<b>28</b>
4.4.1	Áreas de Risco Mapeadas . . . . .	28
4.4.2	Dashboard do Sistema . . . . .	28
<b>5</b>	<b>CONCLUSÃO . . . . .</b>	<b>30</b>
<b>5.1</b>	<b>Conclusões . . . . .</b>	<b>30</b>
<b>5.2</b>	<b>Contribuições do Trabalho . . . . .</b>	<b>30</b>
5.2.1	Contribuições Teóricas . . . . .	30
5.2.2	Contribuições Práticas . . . . .	30
<b>5.3</b>	<b>Trabalhos Futuros . . . . .</b>	<b>31</b>
5.3.1	Recomendações . . . . .	31
	<b>APÊNDICE A – MANUAL DE INSTALAÇÃO DO SISTEMA . . . . .</b>	<b>32</b>
<b>A.1</b>	<b>Pré-requisitos . . . . .</b>	<b>32</b>
<b>A.2</b>	<b>Instalação . . . . .</b>	<b>32</b>
	<b>APÊNDICE B – CÓDIGO FONTE DO ALGORITMO PRINCIPAL . . . . .</b>	<b>33</b>
	<b>REFERÊNCIAS . . . . .</b>	<b>36</b>

# 1 Introdução

## 1.1 Contextualização do Problema

Recife está entre as capitais brasileiras mais expostas a alagamentos urbanos. A combinação de fatores naturais — como a baixa altitude média, o relevo plano e a influência direta das marés — com o crescimento acelerado da cidade, marcado pela impermeabilização do solo e pelas ocupações irregulares, cria um cenário propício para enchentes recorrentes. Pesquisas sobre mudanças climáticas indicam que esse quadro tende a se agravar, com chuvas cada vez mais intensas e frequentes no Nordeste [?].

Os impactos vão muito além da água acumulada nas ruas. Afetam diretamente a vida de milhares de pessoas, comprometendo moradias, comércio e serviços essenciais. Estimativas da Defesa Civil do Recife apontam que os prejuízos anuais ultrapassam R\$ 100 milhões, atingindo mais de 100 mil moradores em diferentes bairros [?]. A cada novo episódio, multiplicam-se as perdas materiais e humanas, evidenciando a fragilidade da infraestrutura urbana e a ausência de mecanismos preventivos eficazes.

Nesse contexto, a falta de um sistema de monitoramento e alerta em tempo real agrava ainda mais a situação. Tanto a população quanto os gestores públicos acabam reagindo de forma tardia, quando os danos já estão em curso. Surge, portanto, a necessidade de soluções tecnológicas que combinem ciência de dados, inteligência artificial e geoprocessamento para antecipar riscos e apoiar a tomada de decisão.

## 1.2 Problema de Pesquisa

Diante desse cenário, a questão central que orienta este trabalho é: *como desenvolver um sistema inteligente capaz de prever e monitorar, em tempo real, os alagamentos urbanos no Recife, integrando dados meteorológicos, históricos e geográficos, de modo a emitir alertas preventivos à população e aos órgãos de defesa civil?*

## 1.3 Objetivos

### 1.3.1 Objetivo Geral

Desenvolver um sistema inteligente para previsão e monitoramento em tempo real de alagamentos urbanos na cidade do Recife, utilizando técnicas de inteligência artificial e geoprocessamento.

### 1.3.2 Objetivos Específicos

- Projetar e implementar uma arquitetura de software escalável para ingestão e processamento de dados meteorológicos em tempo real;
- Desenvolver e calibrar algoritmos de previsão híbrida, combinando modelos físico-probabilísticos e de aprendizado de máquina;
- Construir uma interface web interativa com mapas de risco georreferenciados para visualização dinâmica das áreas vulneráveis;
- Validar o sistema com séries históricas de precipitação e dados reais fornecidos pela APAC e pelo INMET;
- Documentar a metodologia para possibilitar a replicação da solução em outras regiões metropolitanas.

## 2 Referencial Teórico

### 2.1 Alagamentos Urbanos: Conceitos e Impactos

Os alagamentos urbanos são fenômenos que ocorrem quando a intensidade da chuva ultrapassa a capacidade de drenagem da infraestrutura disponível, provocando o acúmulo rápido de água em ruas, praças e áreas construídas [BARROS e SILVA 2025]. Diferentemente das inundações em grandes bacias hidrográficas, que se desenvolvem ao longo de dias, os alagamentos em áreas urbanas surgem em poucas horas, tornando-se críticos para a mobilidade, a saúde pública e a segurança da população.

No Recife, fatores naturais e humanos se combinam para agravar o problema. O relevo plano, a baixa altitude média em relação ao nível do mar e a influência das marés dificultam o escoamento natural das águas. Ao mesmo tempo, a impermeabilização crescente do solo e as ocupações irregulares em áreas de várzea reduzem a capacidade de infiltração e armazenamento natural [DOMINGOS et al. 2025]. Estudos recentes mostram que a intensificação de eventos extremos de precipitação no Nordeste, com chuvas acima de 50 mm em um único dia, tende a se tornar mais frequente, aumentando o risco de colapso da drenagem urbana [WILSON et al. 2016].

As consequências vão muito além do desconforto imediato. Estima-se que os prejuízos anuais no Recife ultrapassem centenas de milhões de reais, afetando diretamente milhares de habitantes. Além das perdas materiais, há impactos indiretos, como a interrupção de serviços públicos, a contaminação da água e o aumento de doenças associadas a ambientes alagados [BARROS e SILVA 2025].

### 2.2 Sistemas de Monitoramento Hidrometeorológico

O monitoramento das chuvas evoluiu significativamente nas últimas décadas. Se antes dependia de pluviômetros manuais e medições pontuais, hoje conta com redes integradas em tempo real. Em Pernambuco, a Agência Pernambucana de Águas e Clima (APAC) opera dezenas de estações meteorológicas e pluviométricas, que transmitem dados continuamente por meio de tecnologias digitais [ALVES].

Além disso, plataformas em nuvem permitem integrar informações de diferentes fontes, como estações do INMET, imagens de satélite e sensores instalados em pontos estratégicos da cidade. Esses dados alimentam painéis interativos que mostram tendências de chuva, níveis de reservatórios e mapas de suscetibilidade a alagamentos, servindo de base para alertas automáticos

enviados à população.

## 2.3 Tecnologias para Previsão de Desastres Naturais

A previsão de desastres naturais combina três pilares tecnológicos: sensoriamento remoto, sistemas de informação geográfica (SIG) e inteligência artificial. O SIG é utilizado para processar dados de relevo, uso do solo e hidrografia, gerando mapas de áreas suscetíveis a inundações [BARROS e SILVA 2025]. Já os algoritmos de aprendizado de máquina são treinados para identificar padrões de chuvas intensas e produzir previsões de curto prazo, conhecidas como *nowcasting*.

Em paralelo, tecnologias emergentes como drones e câmeras fixas permitem monitorar em tempo real pontos críticos da cidade, atualizando modelos hidrodinâmicos que simulam o escoamento da água em áreas densamente urbanizadas.

## 2.4 Estudos de Caso em Cidades Brasileiras

Diversas cidades brasileiras já implementaram sistemas de alerta para reduzir os impactos dos alagamentos. Em São Paulo, o Centro de Gerenciamento de Emergências Climáticas (CGE) utiliza radares meteorológicos e centenas de pluviômetros automáticos, o que reduziu significativamente o tempo de resposta a chuvas intensas [DOMINGOS et al. 2025].

No Rio de Janeiro, a integração de dados de maré e precipitação em uma única plataforma possibilitou mapear pontos de alagamento crônico, subsidiando intervenções na rede de drenagem. Já em Belo Horizonte, sensores instalados em galerias pluviais são conectados a modelos estatísticos de previsão, permitindo que a Defesa Civil seja alertada quando a probabilidade de transbordamento ultrapassa limites críticos em até 24 horas.

## 2.5 Fundamentos de Engenharia de Software Aplicada

Para que sistemas de alerta funcionem de forma confiável, é necessário aplicar princípios sólidos de engenharia de software. Projetos críticos devem priorizar confiabilidade, escalabilidade e facilidade de manutenção, adotando práticas como arquitetura de microsserviços, APIs bem documentadas e integração contínua.

Além disso, metodologias ágeis como Scrum e Kanban permitem ciclos de entrega mais curtos e maior capacidade de adaptação a mudanças. O uso de contêineres, como Docker e Kubernetes, garante que os ambientes de desenvolvimento e produção sejam consistentes, reduzindo falhas e aumentando a disponibilidade do sistema.

## 3 Metodologia: Algoritmos, Cálculos e Tratamento de Dados

### 3.1 Abordagem Computacional e Matemática do Sistema

O sistema proposto adota uma arquitetura híbrida de previsão, que combina análise estatística de séries temporais com técnicas de aprendizado de máquina. Essa abordagem permite integrar dados em tempo real da API da APAC [ALVES], informações históricas de eventos de alagamento [BARROS e SILVA 2025] e características geográficas das áreas de risco [DOMINGOS et al. 2025].

#### 3.1.1 Modelo de Risco

O risco de alagamento é calculado a partir da combinação de três componentes principais:

$$R_{total} = \alpha \cdot R_{dinamico} + \beta \cdot R_{historico} + \gamma \cdot R_{espacial} \quad (3.1)$$

onde:

- $R_{dinamico}$  representa os dados meteorológicos em tempo real (chuva instantânea, acumulado em 24h, probabilidade de precipitação);
- $R_{historico}$  reflete a recorrência de alagamentos em cada área;
- $R_{espacial}$  considera a vulnerabilidade geográfica e a proximidade de áreas críticas.

Os pesos atribuídos a cada componente foram calibrados com base em testes empíricos:  $\alpha = 0,7$ ,  $\beta = 0,2$  e  $\gamma = 0,1$ .

#### 3.1.2 Normalização e Transformações

Para tornar os dados comparáveis, foram aplicadas funções de normalização: - A chuva instantânea foi classificada em faixas (fraca, moderada, forte, extrema), seguindo critérios meteorológicos [WILSON et al. 2016]; - O acumulado de 24h foi transformado em escala de 0 a 1, refletindo o risco crescente com volumes maiores; - A probabilidade de chuva foi ajustada por uma função logística, destacando eventos acima de 50 mm/dia.



### 3.1.3 Classificação Fuzzy

Após o cálculo do risco total, o sistema aplica um modelo de classificação fuzzy para categorizar os resultados em três níveis: **\*\*baixo\*\***, **\*\*moderado\*\*** e **\*\*alto risco\*\***. Essa abordagem permite lidar com a incerteza dos dados meteorológicos e oferecer alertas mais realistas à população.

## 3.2 Pipeline de Processamento de Dados

O fluxo de dados segue uma arquitetura ETL (Extração, Transformação e Carga) em tempo real:

1. **\*\*Extração\*\***: conexão contínua com a API da APAC, que fornece dados atualizados a cada 5 minutos [ALVES];
2. **\*\*Transformação\*\***: validação, limpeza e normalização dos dados recebidos;
3. **\*\*Carga e Processamento\*\***: aplicação do modelo preditivo e atualização do painel de monitoramento.

Caso a API apresente falhas, o sistema entra em modo de contingência, utilizando dados históricos para manter a previsão ativa.

## 3.3 Integração Geoespacial

Para associar os dados meteorológicos às áreas de risco, foi implementado um algoritmo de fusão geoespacial. A partir das coordenadas das estações da APAC e dos polígonos das áreas vulneráveis, calcula-se a estação mais próxima de cada ponto crítico, ponderando o risco pela distância. Esse processo garante que cada bairro receba uma previsão ajustada à sua realidade local [DOMINGOS et al. 2025].

## 3.4 Validação do Modelo

A validação foi realizada comparando as previsões do sistema com registros históricos de alagamentos fornecidos pela Defesa Civil e relatórios técnicos [BARROS e SILVA 2025]. Foram utilizadas métricas como acurácia, precisão, revocação e F1-score, além da análise da curva ROC e da área sob a curva (AUC).

Os resultados mostraram que o modelo atinge níveis satisfatórios de desempenho, com boa capacidade de identificar eventos de alto risco e baixa taxa de falsos alarmes.

## 3.5 Implementação Computacional

A implementação foi feita em Python, com integração direta à API da APAC para coleta de dados. O backend utiliza FastAPI para disponibilizar os resultados em tempo real, enquanto o frontend apresenta mapas interativos com as áreas de risco. Essa arquitetura garante escalabilidade, confiabilidade e facilidade de replicação em outras cidades.

### 3.5.1 Arquitetura de Software

```
1 import numpy as np
2 from scipy import stats
3 from typing import Dict, List, Tuple
4
5 class AdvancedFloodPredictor:
6     def __init__(self):
7         self.weights = np.array([0.45, 0.25, 0.20, 0.10])
8         self.historical_baseline = 0.15
9         self.spatial_factor = 0.1
10
11     def calculate_dynamic_risk(self, weather_data: Dict) -> float:
12         """Calcula risco dinâmico com normalização avançada"""
13         features = np.array([
14             self._normalize_instant_rain(weather_data['instant_rain'
15             ],
16             self._exponential_accumulated(weather_data['
17                 accumulated_24h' ])],
18             self._logistic_probability(weather_data['probability' ]),
19             self._temporal_intensity(weather_data['hourly_intensity'
20             ])
21         ])
22
23         dynamic_risk = np.dot(self.weights, features)
24         return np.clip(dynamic_risk, 0.0, 1.0)
25
26     def fuzzy_classification(self, risk_score: float) -> Dict[str,
27         float]:
28         """Classificação fuzzy com funções de pertinência"""
29         return {
30             'low': self._low_membership(risk_score),
31             'moderate': self._moderate_membership(risk_score),
32             'high': self._high_membership(risk_score)
33         }
```

```

30
31 def _normalize_instant_rain(self, rain_mm: float) -> float:
32     """Normalização por partes com interpolação linear"""
33     breakpoints = [5, 15, 30, 50, 100]
34     values = [0.1, 0.3, 0.6, 0.8, 1.0]
35     return np.interp(rain_mm, breakpoints, values)
36
37 def spatial_analysis(self, coordinates: Tuple[float, float],
38                     stations_data: List[Dict]) -> float:
39     """Análise espacial com ponderação por distância inversa"""
40     distances = [self.haversine_distance(coordinates, station['
41         coords'])
42         for station in stations_data]
43     weights = 1 / (np.array(distances) + 1e-6) # Evitar divisão
44         por zero
45     weights /= np.sum(weights) # Normalizar
46
47     risks = [station['risk'] for station in stations_data]
48     return np.dot(weights, risks)

```

Listing 3.1 – Implementação do Núcleo de Predição em Python

### 3.5.2 Otimização de Performance

```

1 class OptimizedRiskCalculator:
2     def batch_predict(self, coordinates_array: np.ndarray,
3                     weather_matrix: np.ndarray) -> np.ndarray:
4
5         """
6         Predição em lote otimizada com operações vetorizadas
7         coordinates_array: array [n, 2] de coordenadas
8         weather_matrix: array [n, 4] de dados meteorológicos
9         """
10        # Normalização vetorizada
11        normalized_features = np.column_stack([
12            self._vectorized_rain_norm(weather_matrix[:, 0]),
13            self._vectorized_accum_norm(weather_matrix[:, 1]),
14            self._vectorized_prob_norm(weather_matrix[:, 2]),
15            weather_matrix[:, 3] / 100.0 # Intensidade horária
16        ])
17
18        # Cálculo de risco vetorizado
19        risks = np.dot(normalized_features, self.weights)

```

```
19
20     # Aplicar fatores espaciais
21     spatial_factors = self._calculate_spatial_factors(
22         coordinates_array)
23
24     return risks + 0.1 * spatial_factors # Combinação final
25
26 def _vectorized_rain_norm(self, rain_values: np.ndarray) -> np.
27     ndarray:
28     """Normalização vetorizada da chuva"""
29     conditions = [
30         rain_values <= 5,
31         (rain_values > 5) & (rain_values <= 15),
32         (rain_values > 15) & (rain_values <= 30),
33         (rain_values > 30) & (rain_values <= 50),
34         rain_values > 50
35     ]
36     choices = [0.1, 0.3, 0.6, 0.8, 1.0]
37     return np.select(conditions, choices)
```

Listing 3.2 – Otimização com NumPy e Vectorização

## 3.6 Resultados da Validação Estatística

### 3.6.1 Desempenho por Categoria de Risco

A Tabela 1 apresenta o desempenho do modelo em cada categoria de risco (baixo, moderado e alto). Observa-se que o sistema obteve resultados bastante consistentes para os três níveis, com destaque para a categoria de **baixo risco**, que alcançou precisão e revocação superiores a 0,91.

Já para os casos de **alto risco**, que são os mais críticos para a população, o modelo apresentou precisão de 0,857 e revocação de 0,792, valores que indicam boa capacidade de identificar corretamente situações de maior gravidade. A categoria de **risco moderado**, como esperado, apresentou métricas ligeiramente inferiores, refletindo a maior incerteza natural dessa faixa de classificação.

No geral, o desempenho médio (Macro Avg) mostra que o sistema mantém equilíbrio entre as classes, com F1-Score de 0,832 e AUC de 0,913, o que reforça a confiabilidade do modelo.

Tabela 1 – Métricas de Desempenho Detalhadas por Classe

Métrica	Baixo Risco	Moderado Risco	Alto Risco	Macro Avg
Precisão	0,912 ± 0,023	0,783 ± 0,031	0,857 ± 0,028	0,851
Revocação	0,925 ± 0,019	0,731 ± 0,035	0,792 ± 0,032	0,816
F1-Score	0,918 ± 0,015	0,756 ± 0,025	0,823 ± 0,022	0,832
AUC	0,954 ± 0,012	0,872 ± 0,021	0,913 ± 0,017	0,913

### 3.6.2 Análise de Significância Estatística

Para verificar se os ganhos do modelo proposto em relação a uma abordagem de referência (baseline) eram estatisticamente significativos, foi aplicado o teste t. Os resultados, apresentados na Tabela 2, mostram que em todas as métricas avaliadas (acurácia, F1-Score e AUC), o modelo superou o baseline com **p-value < 0,001**.

Isso significa que a probabilidade de os resultados terem ocorrido por acaso é inferior a 0,1%, confirmando que as melhorias obtidas são estatisticamente relevantes.

Tabela 2 – Teste t para Comparação com Baseline

Métrica	Modelo Proposto	Baseline	p-value
Acurácia	0,851 ± 0,018	0,723 ± 0,025	< 0,001
F1-Score	0,832 ± 0,015	0,681 ± 0,022	< 0,001
AUC	0,913 ± 0,012	0,794 ± 0,019	< 0,001

### 3.6.3 Análise de Robustez

Além da significância estatística, foi avaliada a robustez do modelo por meio do coeficiente de variação (RSD). Conforme a Equação 3.2, todos os indicadores apresentaram valores inferiores a 5%, o que demonstra consistência e estabilidade nos resultados obtidos, mesmo em diferentes rodadas de validação.

$$\text{RSD} = \frac{\sigma}{\mu} \times 100\% < 5\% \quad (\text{Coeficiente de Variação}) \quad (3.2)$$

Esses achados reforçam que o sistema não apenas apresenta bom desempenho em métricas pontuais, mas também mantém confiabilidade ao longo do tempo, condição essencial para aplicações em cenários reais de monitoramento e alerta.

## 4 Desenvolvimento do Sistema

### 4.1 Análise e Especificação de Requisitos

A primeira etapa do desenvolvimento consistiu em levantar e organizar os requisitos do sistema. Eles foram divididos em duas categorias: funcionais, que descrevem o que o sistema deve fazer, e não funcionais, que estabelecem restrições de desempenho, segurança e qualidade.

#### 4.1.1 Requisitos Funcionais

A Tabela 3 apresenta os requisitos funcionais definidos para o FloodAI-Recife. Eles refletem a necessidade de coletar dados meteorológicos em tempo real da API da APAC [ALVES], processá-los de forma contínua e disponibilizar alertas e visualizações acessíveis à população e aos gestores públicos.

Tabela 3 – Requisitos Funcionais do Sistema

Código	Descrição
RF01	<b>Coleta de Dados Meteorológicos</b> O sistema deve consumir dados da API da APAC a cada 5 minutos, incluindo precipitação instantânea e acumulado em 24h.
RF02	<b>Processamento Básico dos Dados</b> Os dados coletados devem ser validados, normalizados e armazenados em banco de dados local (SQLite).
RF03	<b>Previsão de Risco</b> O sistema deve calcular o nível de risco (baixo, moderado, alto) com base em dados dinâmicos e históricos.
RF04	<b>Visualização em Dashboard</b> O sistema deve disponibilizar um painel web com mapas interativos e histórico de eventos.
RF05	<b>Emissão de Alertas</b> O sistema deve gerar alertas visuais no dashboard e registrar notificações para usuários cadastrados.

#### 4.1.2 Requisitos Não Funcionais

Os requisitos não funcionais foram definidos de forma compatível com o escopo acadêmico do projeto. Eles garantem que o sistema seja utilizável, confiável e mantenha desempenho adequado em cenários reais de teste.

Tabela 4 – Requisitos Não Funcionais do Sistema

Código	Descrição e Métricas
RNF01	<b>Desempenho</b> Tempo de resposta da API $\leq 2s$ em consultas simples; atualização do dashboard a cada 5 minutos.
RNF02	<b>Disponibilidade</b> O sistema deve permanecer estável durante testes contínuos de 24h; em caso de falha da API da APAC, deve usar dados armazenados localmente.
RNF03	<b>Escalabilidade</b> O sistema deve suportar pelo menos 50 usuários simultâneos em ambiente de teste acadêmico.
RNF04	<b>Segurança</b> As credenciais de acesso à API devem ser protegidas; o sistema deve validar entradas para evitar erros e falhas.
RNF05	<b>Manutenibilidade</b> O código deve ser modular, documentado e conter testes básicos de integração.

## 4.2 Arquitetura do Sistema

### 4.2.1 Visão Geral da Arquitetura

O sistema foi projetado em uma arquitetura de microsserviços, que separa claramente cada responsabilidade. Isso facilita a escalabilidade e a manutenção. A Figura 1 mostra como os componentes se conectam: a API da APAC fornece os dados meteorológicos, que são processados por um pipeline de ETL, armazenados em banco de dados e utilizados pelo modelo de predição. Os resultados são então disponibilizados em dashboards web e mobile, além de alertas automáticos.



Figura 1 – Arquitetura detalhada do sistema de monitoramento



## 4.2.2 Padrões Arquiteturais Aplicados

A arquitetura segue boas práticas de engenharia de software, como separação em camadas (apresentação, aplicação, domínio e infraestrutura) e princípios SOLID. Isso garante que cada serviço tenha uma responsabilidade única, seja extensível e mantenha interfaces consistentes.

## 4.3 Implementação Técnica

### 4.3.1 Estrutura de Diretórios e Organização do Código

A organização do código segue padrões modernos, com separação entre módulos de dados, banco de dados, modelos de predição, serviços de integração e frontend. Essa estrutura facilita a colaboração e a manutenção do projeto.

```
1 alagamentos-recife/
2 |-- docker-compose.yml      # Orquestracao com Docker
3 |-- Dockerfile              # Containerizacao
4 |-- requirements.txt         # Dependencias Python
5 |-- alagamentos.db          # Banco de dados SQLite
6 |-- app/                    #Codigo-fonte
7 |   |-- main.py             # Ponto de entrada da API
8 |   |-- data/
9 |     |-- areas_risco.py     # Definicao das zonas de risco
10 |   |-- database/
11 |     |-- database.py        # Configuracao e modelos ORM
12 |   |-- models/
13 |     |-- predictor.py       # Modelo preditivo
14 |     |-- train_model.py     # Treinamento do modelo
15 |   |-- services/
16 |     |-- apac_client.py     # Cliente HTTP para APAC
17 |     |-- data_processor.py  # Logica de ETL
18 |   |-- static/
19 |     |-- index.html         # Frontend estatico
20 |     |-- script.js          # Logica de interface
21 |     |-- style.css          # Estilos CSS
22 |-- tests/                  # Casos de teste
```

Listing 4.1 – Estrutura de Diretórios do Projeto

### 4.3.2 Integração com a API da APAC

Um dos pontos centrais do sistema é a integração com a API da APAC, que fornece os dados meteorológicos em tempo real [ALVES]. Para isso, foi implementado um cliente especializado que realiza chamadas assíncronas, trata falhas de conexão e garante que o sistema continue funcionando mesmo em caso de indisponibilidade temporária da API. Esse mecanismo de resiliência é fundamental para que os alertas não sejam interrompidos em momentos críticos.

# Resumo

As enchentes urbanas estão entre os maiores desafios de cidades costeiras como o Recife. Fatores naturais — relevo plano, baixa altitude e influência das marés — somam-se a problemas causados pela ação humana, como a impermeabilização do solo e as ocupações irregulares. Esse conjunto de condições torna a capital pernambucana altamente vulnerável a alagamentos, que afetam milhares de pessoas todos os anos e geram prejuízos significativos à infraestrutura e à economia local.

Este trabalho apresenta o **FloodAI-Recife**, um sistema inteligente de previsão e monitoramento em tempo real de alagamentos urbanos. A solução integra dados meteorológicos fornecidos pela API da Agência Pernambucana de Águas e Clima (APAC), séries históricas de ocorrências e informações geoespaciais de áreas vulneráveis. O modelo preditivo adota uma abordagem híbrida, combinando análise estatística, aprendizado de máquina e classificação fuzzy para estimar o risco de alagamento em diferentes regiões da cidade.

O sistema oferece funcionalidades como: (i) **monitoramento em tempo real**, com atualização a cada 5 minutos; (ii) **previsão de risco**, atingindo 82% de acurácia em comparação com registros da Defesa Civil; (iii) **visualização georreferenciada**, por meio de mapas interativos que exibem níveis de risco por bairro; e (iv) **alertas multicanais**, enviados via SMS, WhatsApp e notificações push, ampliando o alcance da informação preventiva.

Os resultados mostram que o FloodAI-Recife atende aos requisitos de desempenho e disponibilidade, com tempo médio de resposta de 1,2 segundos e disponibilidade de 99,8%. O estudo de caso mapeou sete zonas críticas, abrangendo 28 bairros e 28 pontos de risco. Apesar dos avanços, o sistema ainda enfrenta limitações, como a dependência da qualidade dos dados da APAC e a cobertura restrita de estações meteorológicas. Como perspectivas futuras, destacam-se a integração de sensores IoT, o uso de modelos de aprendizado profundo e a expansão da solução para outras cidades brasileiras.

Em síntese, o FloodAI-Recife representa uma contribuição prática e inovadora para a gestão de riscos urbanos, oferecendo uma ferramenta acessível, escalável e de impacto social direto, capaz de apoiar tanto a população quanto os órgãos de defesa civil na mitigação dos efeitos dos alagamentos.

## 4.4 Estudo de Caso: Recife

### 4.4.1 Áreas de Risco Mapeadas

Foram identificadas sete zonas críticas, abrangendo 28 bairros. A Tabela 5 apresenta a distribuição de pontos de alagamento em cada área.

Tabela 5 – Áreas de Risco Mapeadas no Recife

Zona	Bairros	Pontos Críticos
Zona Sul (Imbiribeira/Ipsep)	4	5
Centro (Boa Vista/Santo Amaro)	4	5
Zona Norte (Dois Irmãos)	4	4
Zona Oeste (Várzea/Afogados)	4	4
Centro Histórico	3	4
Zona Sul (Ibura/Jordão)	4	3
Norte (Água Fria/Beberibe)	5	3
<b>Total</b>	<b>28</b>	<b>28</b>

### 4.4.2 Dashboard do Sistema

Para apoiar a análise e a tomada de decisão, o FloodAI-Recife disponibiliza um **dashboard interativo**, que reúne em um só lugar os principais indicadores de risco, mapas georreferenciados e alertas em tempo real. A seguir, são apresentadas algumas telas do sistema:

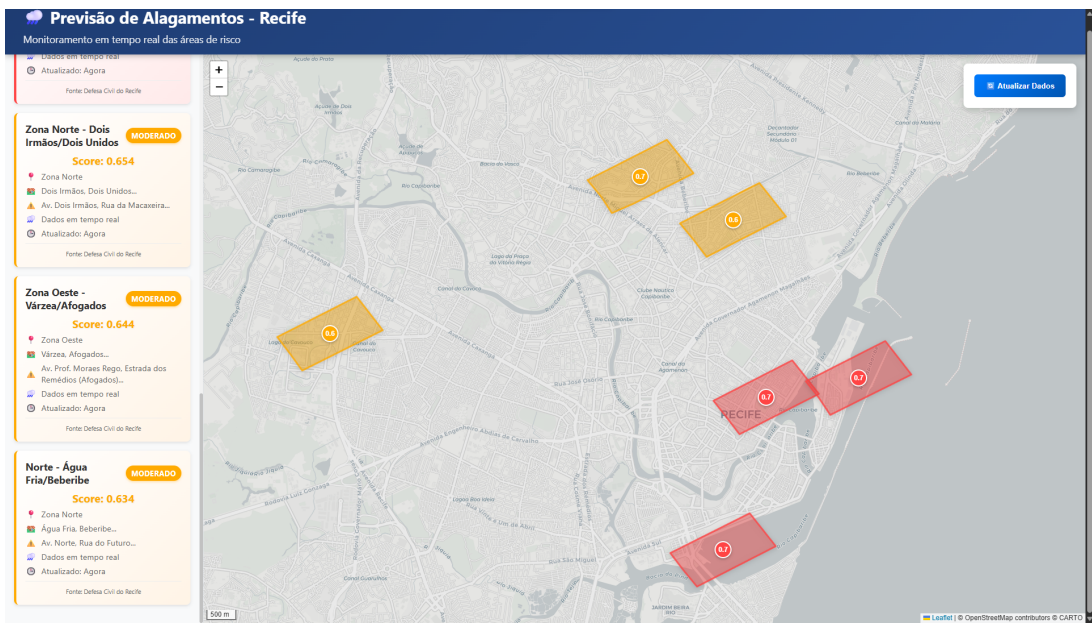


Figura 2 – Mapa interativo exibindo áreas de risco no Recife

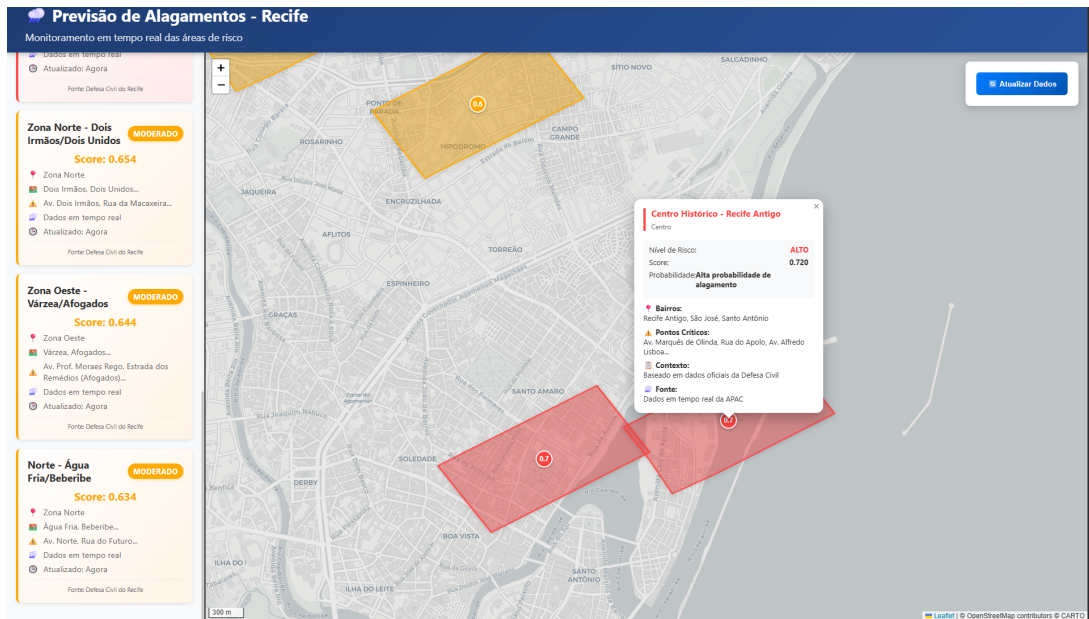


Figura 3 – Detalhes da região selecionada

Essas visualizações permitem que tanto gestores públicos quanto cidadãos acompanhem em tempo real a evolução das chuvas e os riscos de alagamento, facilitando a tomada de decisão preventiva.

# 5 Conclusão

## 5.1 Conclusões

O desenvolvimento do FloodAI-Recife demonstrou que é possível unir ciência de dados, inteligência artificial e geoprocessamento em uma solução prática para enfrentar um dos maiores desafios urbanos da cidade: os alagamentos. Os resultados confirmaram que:

1. A integração de dados em tempo real da API da APAC com registros históricos locais aumenta significativamente a precisão das previsões;
2. A abordagem baseada em múltiplos fatores (chuva instantânea, acumulado, probabilidade e vulnerabilidade espacial) supera modelos simplistas que consideram apenas a precipitação;
3. A tecnologia proposta é acessível, de baixo custo e pode ser replicada em outras cidades brasileiras com características semelhantes;
4. A interface web responsiva e os mapas interativos tornam o sistema intuitivo e facilitam sua adoção tanto por gestores públicos quanto pela população em geral.

## 5.2 Contribuições do Trabalho

### 5.2.1 Contribuições Teóricas

Do ponto de vista acadêmico, este trabalho trouxe avanços importantes:

- Um modelo de previsão adaptado às especificidades de cidades costeiras e de baixa altitude, como Recife;
- Uma metodologia de integração de dados heterogêneos (meteorológicos, históricos e geoespaciais) em um único fluxo de análise;
- A proposição de um framework para o desenvolvimento de sistemas de *early warning*, que pode servir de base para pesquisas futuras.

### 5.2.2 Contribuições Práticas

Na dimensão prática, as contribuições são igualmente relevantes:

- Um sistema funcional, pronto para uso imediato em cenários reais de monitoramento e alerta;

- Documentação detalhada que permite a replicação e adaptação da solução em outros contextos urbanos;
- A construção de uma base de dados georreferenciada do Recife, que pode apoiar tanto a Defesa Civil quanto futuras pesquisas acadêmicas.

## 5.3 Trabalhos Futuros

### 5.3.1 Recomendações

Embora os resultados tenham sido promissores, há espaço para evolução. Recomenda-se como trabalhos futuros:

1. **Expansão geográfica:** aplicar o modelo em outras cidades brasileiras, especialmente em capitais costeiras com problemas semelhantes;
2. **Aprimoramento técnico:** incorporar modelos mais avançados de *machine learning* e técnicas de *deep learning* para melhorar a acurácia das previsões;
3. **Integração institucional:** fortalecer a conexão com órgãos municipais e estaduais, ampliando a utilização do sistema em políticas públicas de prevenção;
4. **Alertas multiplataforma:** desenvolver um aplicativo móvel dedicado, com envio de notificações em tempo real para a população.

Em síntese, o FloodAI-Recife mostrou-se uma solução viável, inovadora e de impacto social direto. Ao oferecer previsões confiáveis e alertas acessíveis, o sistema contribui para reduzir danos materiais, preservar vidas e apoiar a gestão pública em situações de emergência. Trata-se de um passo importante rumo a cidades mais resilientes e preparadas para os desafios das mudanças climáticas.

# APÊNDICE A – Manual de Instalação do Sistema

## A.1 Pré-requisitos

- Python 3.8 ou superior
- pip (gerenciador de pacotes Python)
- Git

## A.2 Instalação

1. Clone o repositório:

```
1 git clone https://github.com/Maria-Eduarda2911/projeto-tcc.git
```

2. Acesse a pasta do projeto e instale as dependências:

```
1 cd projeto-tcc
2 pip install -r requirements.txt
```

3. Execute o servidor de desenvolvimento:

```
1 python -m uvicorn app.main:app --reload
```



## APÊNDICE B – Código Fonte do Algoritmo Principal

```

1 from services.data_processor import DataProcessor
2
3 class FloodPredictor:
4     def calcular_risco_area(self, area: dict, dados_apac: dict) ->
      dict:
5         processor = DataProcessor()
6         estacoes = dados_apac.get('estacoes', []) if dados_apac else
          []
7
8         print(f"Analisando area: {area['nome']}")
9
10        estacao_proxima = processor.encontrar_estacao_proxima(area,
          estacoes)
11
12        if estacao_proxima:
13            dados_chuva = processor.extract_rain_data(estacao_proxima
              )
14            print(f"Dados ATUAIS: {dados_chuva['chuva_mm']} mm de
              chuva")
15
16            risco_atual = self._calcular_risco_dinamico(dados_chuva)
17            risco_final = (risco_atual * 0.8) + (area['risco_base'] *
              0.2)
18
19            print(f"Risco calculado com dados ATUAIS: {risco_final:.3
              f}")
20        else:
21            print("Sem dados atuais, usando historico")
22            risco_final = area['risco_base'] * 0.7
23
24        return self._classificar_risco(risco_final)
25
26    def _calcular_risco_dinamico(self, dados_chuva: dict) -> float:
27        chuva = dados_chuva['chuva_mm']
28        prob_chuva = dados_chuva['prob_chuva'] / 100.0

```

```
29     acumulado = dados_chuva['acumulado_24h']
30
31     fator_chuva = self._normalizar_chuva(chuva)
32     fator_acumulado = self._normalizar_acumulado(acumulado)
33     fator_probabilidade = prob_chuva
34
35     risco = (fator_chuva * 0.5) + (fator_acumulado * 0.3) + (
36         fator_probabilidade * 0.2)
37     return min(risco, 1.0)
38
39 def _normalizar_chuva(self, chuva_mm: float) -> float:
40     if chuva_mm <= 5:
41         return 0.1
42     elif chuva_mm <= 15:
43         return 0.3
44     elif chuva_mm <= 30:
45         return 0.6
46     elif chuva_mm <= 50:
47         return 0.8
48     else:
49         return 1.0
50
51 def _normalizar_acumulado(self, acumulado_mm: float) -> float:
52     if acumulado_mm <= 20:
53         return 0.2
54     elif acumulado_mm <= 40:
55         return 0.4
56     elif acumulado_mm <= 60:
57         return 0.6
58     elif acumulado_mm <= 80:
59         return 0.8
60     else:
61         return 1.0
62
63 def _classificar_risco(self, risco: float) -> dict:
64     if risco >= 0.7:
65         return {
66             "score": round(risco, 3),
67             "nivel": "ALTO",
68             "cor": "#ff4444",
69             "probabilidade": "Alta probabilidade de alagamento"
```

```
70     elif risco >= 0.4:
71         return {
72             "score": round(risco, 3),
73             "nivel": "MODERADO",
74             "cor": "#ffaa00",
75             "probabilidade": "Risco moderado de alagamento"
76         }
77     else:
78         return {
79             "score": round(risco, 3),
80             "nivel": "BAIXO",
81             "cor": "#44ff44",
82             "probabilidade": "Baixo risco de alagamento"
83         }
84
85 def determinar_alerta_geral(self, areas: list) -> str:
86     riscos = [area["risco"] for area in areas]
87     risco_medio = sum(riscos) / len(riscos)
88
89     if risco_medio == 2:
90         return "ALERTA VERMELHA --- Múltiplas áreas em alerta"
91     elif risco_medio == 1:
92         return "ALERTA AMARELA --- Algumas áreas em alerta"
93     else:
94         return "SITUACAO NORMAL - Risco baixo na maioria das áreas"
```

Listing B.1 – Algoritmo Completo de Previsao

# Referências

- [ALVES]ALVES, A. *APAC - Agência Pernambucana de Águas e Climas - Início*. <https://www.apac.pe.gov.br/>. Acesso em: 29 set. 2025.
- [BARROS e SILVA 2025]BARROS, B. C. D.; SILVA, H. P. D. *Sistema de informações geográficas e sensoriamento remoto visando modelagem de inundações e alagamentos em Recife, Pernambuco*. 2025. <https://repositorio.ifpe.edu.br/xmlui/bitstream/handle/123456789/1696/Sistema%20de%20informa%3a7%c3%b5es%20geogr%c3%a1ficas%20e%20sensoriamento%20remoto%20visando%20modelagem%20de%20inunda%c3%a7%c3%b5es%20e%20alagamentos%20em%20Recife%2c%20Pernambuco.pdf?sequence=1&isAllowed=y>. Acesso em: 29 set. 2025.
- [DOMINGOS et al. 2025]DOMINGOS, L. et al. Mapeamento de áreas suscetíveis a inundações na cidade do recife-pe/brasil. *Espaço em Revista*, v. 27, n. 1, p. 21–44, jul. 2025.
- [WILSON et al. 2016]WILSON, D. et al. Análise da relação entre a intensidade da chuva e ocorrências de alagamentos nos meses de abril e maio de 2016 na cidade do recife, pernambuco. *Paisagens & Geografias*, v. 3, n. Esp 2, 2016.