UNIVERSIDADE FEDERAL DE PERNAMBUCO

CAMPUS RECIFE

JOÃO PEDRO VELOSO

LUCAS FLORENCIO DE SOUSA

LUCAS LEANDRO DA SILVA

GERAÇÃO DE TESTES POR LLM'S PARA API'S REST: UMA REVISÃO DOS FATORES DE IMPLEMENTAÇÃO

RECIFE
2025

1 Introdução

O avanço das tecnologias de Inteligência Artificial (IA) tem transformado profundamente diversos campos, incluindo o desenvolvimento e manutenção de APIs REST, elementos fundamentais na arquitetura de software moderna. Com a crescente complexidade e número de endpoints dessas APIs, surgem desafios significativos para garantir qualidade, confiabilidade e conformidade com especificações. Nesse contexto, os testes automatizados desempenham um papel crucial, mas as abordagens tradicionais apresentam limitações que dificultam sua aplicação em larga escala.

Este estudo tem como objetivo realizar uma revisão sistemática da literatura para analisar o estado da arte e as tendências de pesquisa relacionadas à geração automatizada de testes para APIs REST utilizando modelos de linguagem avançados (LLMs). A pesquisa busca identificar as principais práticas, desafios e inovações no uso dessas tecnologias emergentes, destacando como elas podem superar as limitações dos métodos tradicionais. Além disso, o trabalho investiga fatores contextuais que influenciam a eficácia dos testes gerados por IA, comparando-os com abordagens manuais e ferramentas específicas existentes.

2 Motivação e Trabalhos Relacionados

O desenvolvimento e manutenção de APIs REST constituem elementos fundamentais na arquitetura de software moderna, sendo essenciais para a integração de sistemas distribuídos e aplicações multiplataforma. À medida que a complexidade e o número de endpoints dessas APIs crescem, torna-se progressivamente mais desafiador garantir sua qualidade, confiabilidade e conformidade com as especificações. Testes automatizados representam uma estratégia crítica para enfrentar esse desafio, mas as abordagens tradicionais apresentam limitações significativas.

2.1 O Problema da Testabilidade de APIs REST

O teste adequado de APIs REST enfrenta diversos desafios técnicos e organizacionais. Um desenvolvedor responsável por uma API com dezenas ou centenas de endpoints precisa criar e manter casos de teste que verifiquem:

- A correta validação de diferentes tipos de entrada
- O gerenciamento apropriado de estados e dependências
- O tratamento adequado de erros e exceções
- A conformidade com contratos de API e documentação
- O comportamento correto em diferentes contextos e configurações

As abordagens tradicionais para testes de API REST – sejam manuais ou baseadas em ferramentas especializadas como Postman, REST Assured ou frameworks específicos de linguagem – exigem investimento considerável de tempo e expertise. Para ilustrar concretamente, considere o cenário de uma API de e-commerce com endpoints para gerenciamento de produtos, pedidos, usuários e pagamentos: cada endpoint pode requerer dezenas de casos de teste para verificar diferentes combinações de entradas, condições de negócio e cenários de erro. A escrita manual desses testes não apenas consome recursos significativos, mas também frequentemente resulta em cobertura inconsistente ou incompleta.

2.2 Limitações das Abordagens Atuais

Os métodos convencionais de teste de API apresentam várias limitações críticas:

- Custo de desenvolvimento elevado: A criação manual de testes abrangentes é intensiva em trabalho, exigindo conhecimento profundo tanto da API quanto das ferramentas de teste.
- Manutenção complexa: Mudanças na API frequentemente quebram testes existentes, criando um ciclo contínuo de manutenção que compete com o desenvolvimento de novos recursos.
- Cobertura inconsistente: Limitações de tempo e recursos muitas vezes levam a decisões pragmáticas sobre o que testar, resultando em áreas da API com cobertura insuficiente.
- Barreira de conhecimento: Ferramentas especializadas de teste de API geralmente possuem curvas de aprendizado acentuadas, limitando quem pode efetivamente contribuir para o processo de teste.
- Dificuldade em acompanhar a evolução: APIs modernas evoluem rapidamente, e os testes tradicionais frequentemente ficam defasados em relação às especificações atualizadas.

O surgimento dos Large Language Models (LLMs) como o GPT-4, Claude e LLaMA introduziu uma abordagem potencialmente transformadora: a geração automatizada de código de teste a partir de especificações de API, documentação ou mesmo análise de código existente. Esta capacidade promete reduzir drasticamente o esforço manual enquanto potencialmente aumenta a cobertura e consistência dos testes.

2.3 Consequências das Limitações Atuais

As limitações nas abordagens tradicionais de teste de API resultam em consequências significativas:

- Qualidade comprometida: Testes inadequados ou incompletos permitem que bugs e comportamentos inesperados cheguem aos ambientes de produção, afetando a experiência do usuário e potencialmente causando perdas financeiras.
- Ciclos de desenvolvimento mais lentos: O tempo elevado necessário para criar e manter testes manuais retarda a entrega de novos recursos e melhorias.

- Dívida técnica acumulada: À medida que as APIs evoluem, os testes desatualizados ou incompletos contribuem para uma crescente dívida técnica que eventualmente exige investimentos significativos para ser sanada.
- Resistência à refatoração: Sem uma suíte de testes confiável e abrangente, desenvolvedores hesitam em refatorar ou redesenhar APIs existentes, resultando em deterioração gradual da qualidade do código.
- Documentação e especificação desatualizadas: A falta de testes que verifiquem a conformidade com a documentação frequentemente leva a uma divergência crescente entre a API implementada e sua especificação publicada.

Embora a utilização de LLMs para geração de testes de API REST apresente um potencial significativo para superar essas limitações, ainda existe uma lacuna de conhecimento sistemático sobre sua eficácia em diferentes contextos, fatores que influenciam seu sucesso, melhores práticas emergentes e como eles se comparam a abordagens tradicionais. A ausência deste conhecimento estruturado impede que organizações e desenvolvedores tomem decisões informadas sobre quando e como adotar esta tecnologia emergente.

É precisamente esta lacuna que nosso estudo visa preencher, proporcionando uma análise sistemática e comparativa do uso de LLMs para geração de testes de API REST em diferentes contextos e escalas de aplicação, baseada em evidências documentadas na literatura recente.

3 Metodologia

Diante das limitações identificadas nas abordagens tradicionais de teste para APIs REST e do potencial transformador apresentado pelos modelos de linguagem avançados (LLMs), surge a necessidade de uma análise aprofundada sobre o tema. Essa lacuna no conhecimento motiva a realização deste estudo, que busca compreender como as soluções baseadas em IA podem superar os desafios existentes, como custos elevados, manutenção complexa e cobertura inconsistente.

Dessa forma, este estudo adota a Revisão Sistemática da Literatura (RSL) como metodologia principal. Essa escolha se justifica pela capacidade da RSL de organizar e sintetizar o conhecimento existente na área, permitindo uma análise abrangente das evidências científicas disponíveis. A seguir, detalham-se as etapas metodológicas empregadas para garantir rigor e consistência na condução da pesquisa.

A metodologia de Kitchenham é composta por sete etapas distintas:

1. Planejamento
2. Protocolo de pesquisa
3. Definição e aplicação de filtros
4. Qualificação dos estudos
5. Seleção de artigos
6. Coleta de evidências
7. Análise e síntese dos dados

As etapas foram utilizadas para consolidar os dados que fundamentam este estudo, conforme ilustrado na Figura 1.

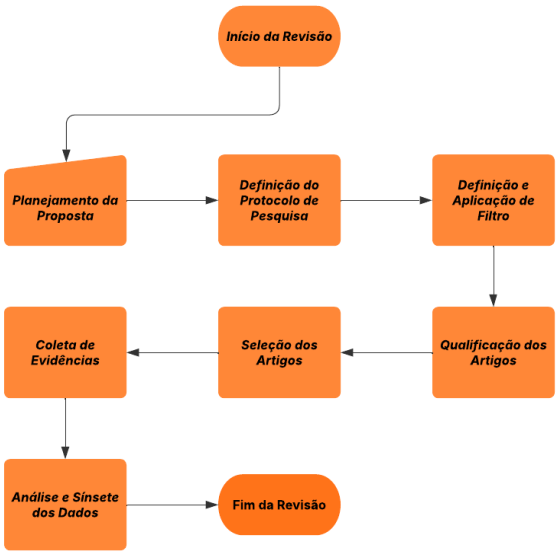

Figura 1: Etapas da Metodologia

Nas primeiras etapas, o problema central de pesquisa, assim como as perguntas de pesquisa são definidas. Seguidas das definições de quais bibliotecas digitais serão pesquisadas e da criação da string de busca, além dos critérios de inclusão e exclusão para validar o protocolo. Em seguida, são realizadas buscas sistemáticas em bases de dados científicas para identificar estudos relevantes, aplicando critérios de seleção e avaliando a qualidade dos estudos escolhidos. Em seguida, os dados são extraídos e sintetizados para responder às perguntas da pesquisa. A última fase é a documentação dos resultados encontrados.

A aplicação desta metodologia possibilitou a organização de conceitos e a compreensão das principais problemáticas relacionadas à testes gerados por IA para API 's REST.

3.1 Questões de Pesquisa

O objetivo geral desta pesquisa é investigar as principais práticas, empecilhos e os fatores que podem contribuir para a eficácia de testes em APIs gerados por IA. O estudo do estado da arte do tema é necessário para identificar as evidências científicas existentes. Visando a concordância com essas temáticas, a pesquisa tem como questão central: “Quais são as melhores práticas e lições aprendidas com a implementação de testes de API REST por IA?”. A partir da questão central, foram elaboradas quatro questões específicas para concretizar e aprofundar o tema, vistas na tabela 1 abaixo:

ID	Questão de Pesquisa
QP1	Quais fatores contextuais da API que influenciam o sucesso da implementação?
QP2	Como a eficácia dos testes gerados por IA para APIs REST varia em diferentes contextos e escalas de aplicação?

QP3	Como os testes gerados por IA se comparam aos testes manuais ou de outras ferramentas específicas para APIs REST?

Tabela 1: Questões de Pesquisa

3.2 String de Pesquisa

Para garantir uma cobertura abrangente e relevante da literatura científica, foi desenvolvida uma string de busca detalhada, adaptada especificamente para a RSL.

("software testing" OR "test automation" OR "automated testing" OR "test generation") AND ("REST API" OR "RESTful API" OR "web API") AND ("artificial intelligence" OR "AI" OR "machine learning" OR "LLM" OR "generative AI" OR "ChatGPT")

As palavras-chave foram escolhidas por sua abrangência, enriquecendo a seleção de estudos com abordagens de vários campos de estudo. As siglas relacionadas a diferentes tipos de API foram selecionadas para garantir a cobertura de todo o contexto de pesquisa da área de API, enriquecendo o conteúdo quando falamos do ambiente em que essas tecnologias são utilizadas.

3.3 Seleção dos Estudos

Com a finalização das questões de pesquisa e definição da string de busca, foi necessário selecionar as bibliotecas digitais que garantiriam a qualidade e relevância dos resultados obtidos. Foram selecionadas três bases científicas que contemplam as áreas tecnológicas relacionadas ao tema, sendo elas: IEEE Xplore Digital Library, ACM Digital Library e Elsevier Science Direct. Após a primeira consulta, foram encontrados 58 estudos, dos últimos 5 anos.

Biblioteca Digital	Quantidade de Estudos
IEEE	3
ACM	55
Science Direct	0

Tabela 2: Quantidade dos Estudos na primeira busca

3.4 Critérios de Seleção

A pesquisa focou na seleção de critérios que favorecessem a relação com o tema principal. Os estudos teriam que abordar API, IA ou testes. Para desfavorecer um dos pontos de ameaça à validade, requisitos como tempo de publicação e temática foram ajustados. Esses critérios estão apresentados na tabela abaixo.

ID	**Critério de Inclusão**
CI1	Artigos que mencionam API e IA ou Testes.
CI2	Artigos em Inglês ou português.
CI3	Artigos de revisão
ID	**Critério de Exclusão**
CE1	Artigos que não mencionem API, IA ou testes
CE2	Artigos publicados há mais de 6 anos.
CE3	Artigos com menos de 5 páginas
CE4	Artigos com mais de 30 páginas
CE5	Artigos que não estejam em ingles ou portugues
CE6	Artigos que não tenham informações bem estruturadas e documentadas

Tabela 3: Critérios de Inclusão e Exclusão

Primeiramente, foi realizada a seleção pela leitura dos títulos, seguida da leitura dos resumos dos estudos restantes. A conclusão de todas as etapas foi importante para remover os estudos que poderiam enviesar o resultado.

Biblioteca Digital	Quantidade de estudos	Quantidade de exclusões após filtro	Quantidade de artigos restantes nessa etapa
IEEE	3	0	3
ACM	55	30	25

Tabela 4: Quantidade de Estudos no Processo de Inclusão e Exclusão

3.5 Avaliação de Qualidade

Os estudos selecionados após os filtros iniciais passaram por uma avaliação qualitativa que avaliava a aderência com a temática central, a presença de informações sobre pontos positivos e de impacto para a validade, além do impacto da pesquisa. De acordo com Kitchenham e Charters (2007), essa etapa de avaliação é crucial para garantir a qualidade do estudo. Como medida de cálculo, foram definidas as pontuações e suas respectivas categorias. Os critérios foram pontuados seguindo os valores:

- Nota 0 caso o artigo não atenda ao critério
- Nota 0,5 caso o artigo atenda parcialmente ao critério
- Nota 1 caso o artigo atenda totalmente

ID	Questões de Avaliação de Qualidade
AQ1	Existe um contexto claro do tema a ser abordado?

AQ2	A metodologia de pesquisa é bem definida?
AQ3	Existe uma organização lógica e clara dos tópicos do estudo?
AQ4	As limitações e ameaças são comentadas?

Tabela 5: Questões de Avaliação da Qualidade dos Estudos

Sendo, portanto, 5, a nota máxima possível na pontuação total para a avaliação do artigo. Aqueles que obtiveram notas inferiores a 3 foram excluídos. No fim, a lista final tinha 12 artigos selecionados.

Base de Dados	Filtros de Seleção	Critérios de Seleção	Análise de Qualidade	Resultado final
IEEE	3	3	3	3
ACM	55	25	9	9
Total	58	29	12	12

Tabela 6: Quantidade de Estudos da Lista Final

3.7 Ameaças à Validade

Durante a construção das etapas da pesquisa, alguns pontos podem ser uma ameaça à validade da pesquisa.

Quanto à seleção de artigos, algumas bibliotecas digitais não aceitam strings de busca grandes, que conseguem abordar todo o campo de estudo requerido. Para contornar esse ponto, foi feita a síntese da string de busca com as palavras chaves mais utilizadas, retiradas das questões de pesquisa. Além disso, a delimitação de tempo para filtro dos estudos colabora para que a investigação seja relevante, unindo os estudos mais atuais. Outro ponto importante foi a dificuldade na coleta de artigos na biblioteca Science Direct, que mostrou-se engessada no processo, não tendo estudos retornados e selecionados.

4 Resultados

A seleção dos artigos resultou em uma lista final com 12 artigos. A aplicação dos critérios de exclusão pôde eliminar boa parte dos estudos que não tinham relação direta com a temática e proporcionou uma lista de estudos relevantes para a resposta das questões de pesquisa. A análise também evidenciou que a maioria dos artigos selecionados foram publicados no ano de 2024. Esses dados sugerem que a temática encontrou-se em expansão após 2023 com o lançamento do GPT, mas, até agora, pode estar sendo deixada de lado perante a quantidade de IAs sendo criadas.

A ACM começou os filtros dominando a quantidade de publicações. Mesmo após a aplicação dos filtros, essa biblioteca permanece com mais de 70% dos artigos presentes.

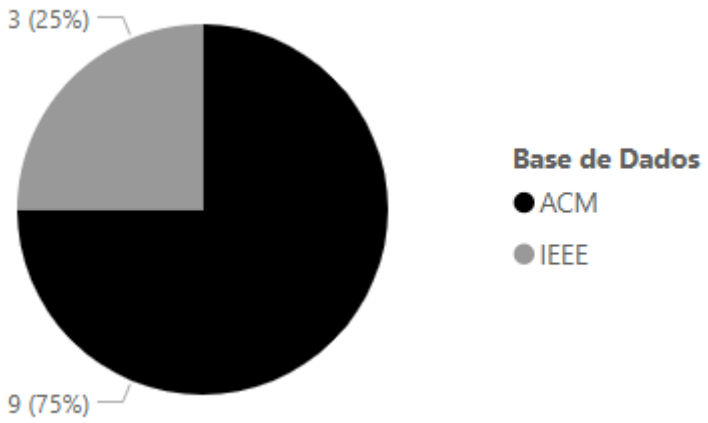

Figura 3: Percentual de artigos por base de dados

Em sua maioria, os estudos selecionados respondem três ou mais questões de pesquisa. No entanto, percebeu-se, após análise das perguntas de qualidade, que existe uma limitação de informações sobre as características das API estudadas.

5 Discussão

Q1- Quais fatores contextuais (domínio, tamanho da API, complexidade do contrato) influenciam o sucesso da abordagem?

A aplicação de LLMs na geração de testes para APIs REST surge como uma abordagem promissora para automatizar e aprimorar o processo de teste [Kim, 2024]. No entanto, o sucesso desta abordagem não é homogêneo, sendo significativamente influenciado por diversos fatores contextuais inerentes à API e ao seu ambiente de aplicação. Dentre os principais fatores, destacam-se o domínio da aplicação, o tamanho da API e a complexidade dos seus requerimentos.

No que se refere ao domínio da aplicação, a natureza específica das funcionalidades e dos dados transacionados pela API impõe diferentes desafios para a geração de testes eficazes por LLMs. Em domínios críticos, como o da saúde, a mera cobertura sintática provida pelas especificações da API (e.g., OpenAPI Specification) pode ser insuficiente. A capacidade do LLM de compreender e gerar testes que validem regras de negócio complexas e nuances semânticas específicas do domínio torna-se crucial [Laaber, 2023]. A geração de valores de parâmetros que respeitem as restrições contextuais e as relações implícitas nos dados de saúde, por exemplo, exige um nível de compreensão semântica que LLMs bem treinados podem potencialmente oferecer [Kim et al., 2024]. Contudo, a avaliação do sucesso dos testes neste contexto requer métricas que vão além da cobertura de código, como a capacidade de exercitar regras médicas específicas [Laaber, 2023].

O tamanho da API, medido pelo número de endpoints, parâmetros e a extensão das suas funcionalidades, representa outro fator contextual relevante. APIs maiores apresentam um espaço de busca de testes muito mais vasto, o que pode tornar abordagens de geração de testes puramente exploratórias ou baseadas em amostragem aleatória menos eficientes [Kim et al., 2022]. Nesse cenário, a capacidade dos LLMs de priorizar a geração de testes para áreas mais críticas da API, com base na análise da especificação e potencialmente em informações adicionais como a frequência de uso ou o risco associado a determinadas funcionalidades, pode ser um diferencial para o sucesso da abordagem. Técnicas que utilizam aprendizado por reforço, como o DeepREST

[Corradini et al., 2024], demonstram potencial para navegar em espaços de testes complexos, aprendendo a gerar sequências de chamadas de API mais eficazes.

A complexidade dos requerimentos por contrato da API, que abrange a presença de dependências inter-parâmetros, a lógica de negócio expressa em linguagem natural nas descrições da especificação e outros requisitos não formalizados, exerce uma influência significativa no sucesso da geração de testes por LLMs. Especificações formais, como a OpenAPI, podem não capturar todas as nuances e restrições semânticas de uma API [Martin-Lopez et al., 2019]. A capacidade dos LLMs de interpretar as descrições em linguagem natural presentes na especificação para inferir restrições implícitas e dependências entre parâmetros, como demonstrado pelo NLPtoREST [Kim et al., 2023] e pelo RESTGPT [Kim et al., 2024], é fundamental para gerar testes válidos e que explorem cenários relevantes. A ausência dessa capacidade pode levar à geração de um grande número de testes inválidos ou à falha em cobrir aspectos importantes da lógica de negócio da API.

Em suma, o sucesso da abordagem de testes gerados por LLMs para APIs REST é intrinsecamente ligado aos fatores contextuais da aplicação. Um domínio com requisitos semânticos complexos exige LLMs com forte capacidade de compreensão contextual e geração de dados válidos. APIs de grande porte demandam a habilidade dos LLMs de priorizar a exploração do espaço de testes de forma inteligente. Por fim, requerimentos de API com alta complexidade semântica e dependências não triviais requerem LLMs capazes de inferir essas informações a partir de fontes textuais. A capacidade de adaptar as técnicas de geração de testes baseadas em LLMs a esses diferentes fatores contextuais será determinante para o sucesso e a adoção desta promissora área de pesquisa.

Q2 - Como a eficácia dos testes gerados por IA para APIs REST varia em diferentes contextos e escalas de aplicação?

Em relação ao contexto de teste, a disponibilidade e a qualidade da especificação da API, como a OpenAPI Specification (OAS), desempenham um papel fundamental. Muitas ferramentas de teste baseadas em IA utilizam essas especificações para guiar a geração de testes. Contudo, as especificações podem ser incompletas ou carecer de

informações semânticas importantes, como dependências interparâmetros e restrições implícitas. Trabalhos como o de Martin-Lopez et al. identificaram a comum ocorrência de dependências interparâmetros não formalmente especificadas em OAS2. Para mitigar essas limitações, técnicas de Processamento de Linguagem Natural (PLN) têm sido exploradas para extrair informações relevantes de descrições em linguagem natural nas especificações, incluindo valores de exemplo e dependências. A ferramenta RESTGPT, por exemplo, utiliza Modelos de Linguagem Grandes (LLMs) para extrair regras e gerar valores de parâmetros a partir dessas descrições, demonstrando melhorias na extração de regras e geração de valores em comparação com abordagens existentes.

A natureza do teste (caixa-preta ou caixa-branca) também influencia a eficácia. Ferramentas como o EvoMaster podem operar em ambos os modos. No modo caixa-branca, com acesso ao código-fonte, algoritmos evolutivos são utilizados para maximizar a cobertura do código. Estudos têm demonstrado a capacidade do EvoMaster em atingir alta cobertura de código e detectar erros em APIs REST. Já no modo caixa-preta, a geração de testes se baseia principalmente na especificação da API e em técnicas como geração aleatória de entradas ou análise de dados de requisição e resposta para informar parâmetros dependentes, como na abordagem adaptativa com aprendizado por reforço ARAT-RL. ARAT-RL mostrou-se eficaz na geração de requisições válidas e indutoras de falhas, cobrindo mais operações e alcançando maior cobertura de código em comparação com outras ferramentas.

A complexidade da API é outro fator determinante. APIs com muitas operações, parâmetros e interações complexas representam um espaço de busca vasto para a geração de testes. Técnicas que consideram as relações produtor-consumidor entre operações e que são capazes de gerar testes com estado são cruciais para APIs complexas. Ferramentas como RESTler e RestTestGen buscam inferir essas dependências para produzir testes mais eficazes, embora com diferentes graus de precisão. DeepREST, utilizando aprendizado por reforço profundo, demonstrou ser eficaz na seleção de ordenamentos de operações e na geração de valores de entrada válidos, superando outras ferramentas em termos de cobertura e detecção de falhas em um conjunto de APIs de estudo de caso.

Em relação à escala de aplicação, a eficácia dos testes gerados por IA tem sido avaliada em diferentes cenários, desde APIs menores até APIs industriais reais. Um estudo de caso no Cancer Registry of Norway utilizou o EvoMaster para testar o motor de regras GURI, demonstrando a aplicabilidade de ferramentas de teste de API baseadas em IA em contextos do mundo real, embora com desafios na geração de dados médicos sintéticos válidos. A intenção de realizar testes contínuos em APIs industriais como YouTube e Spotify também indica um movimento para avaliar a escalabilidade e a eficácia dessas técnicas em aplicações de grande escala.

Apesar dos avanços, existem desafios como a geração de dados semanticamente válidos, especialmente em domínios específicos, e o tratamento eficaz de dependências complexas e comportamentos com estado. Além disso, a incapacidade dos estudantes em formular perguntas suficientemente detalhadas para assistentes inteligentes baseados em GPT pode levar a exemplos de código incorretos, levantando dúvidas sobre sua utilidade para o aprendizado de testes de software.

Q3- Como os testes gerados por IA se comparam aos testes manuais ou gerados por outras ferramentas específicas para APIs REST?

Em relação aos testes manuais, um estudo de caso no Cancer Registry of Norway (CRN) [Laaber et al., 2023] observou que o teste manual é uma prática comum na indústria, incluindo o teste do motor de regras médicas GURI. No entanto, este estudo aplicou uma ferramenta de teste de API REST baseada em IA de código aberto, o EvoMaster, para avaliar sua eficácia em comparação com a prática manual existente. Embora o estudo não compare diretamente a qualidade dos testes manuais existentes com os gerados pelo EvoMaster em profundidade, a motivação para o uso de uma ferramenta automatizada baseada em IA como o EvoMaster, demonstrou superioridade a outras ferramentas REST, sugere uma busca por maior eficiência e potencial cobertura que pode ser difícil de alcançar com testes puramente manuais. A automatização através de IA pode reduzir o esforço manual e, potencialmente, descobrir cenários de teste mais diversos e complexos [Martin-Lopez et al., 2021].

Ao comparar testes gerados por IA com testes produzidos por outras ferramentas específicas para APIs REST (que podem não utilizar técnicas avançadas de IA como

PLN ou aprendizado por reforço), diversos estudos empíricos fornecem informações valiosas. Um estudo comparativo de dez ferramentas de teste de API REST de última geração [Kim et al., 2022] incluiu tanto ferramentas que poderíamos considerar "específicas para APIs" (como Dredd, RESTest, Schemathesis) quanto uma ferramenta com capacidades de IA (EvoMaster). Os resultados demonstraram variações significativas no desempenho entre as ferramentas em termos de cobertura de código e falhas únicas detectadas, indicando que a presença de técnicas de IA (como em EvoMaster) pode levar a resultados superiores em certos aspectos [Kim et al., 2022].

Outro estudo [Corradini et al., 2024] avaliou o DeepREST, uma ferramenta de geração de testes baseada em aprendizado por reforço profundo, comparando-o com uma "linha de base de última geração" de outras ferramentas de teste de API REST. Os resultados indicaram que o DeepREST demonstrou maior eficiência em termos de cobertura de branch, linha e método, além de ser superior à maioria das ferramentas de última geração na revelação de falhas únicas [Corradini et al., 2024]. Isso sugere que abordagens de IA mais avançadas, como o aprendizado por reforço, podem oferecer vantagens significativas na geração de testes mais eficazes em comparação com ferramentas que utilizam outras estratégias.

Além disso, a emergente utilização de Processamento de Linguagem Natural (PLN) para aprimorar a geração de testes, como nas ferramentas NLPtoREST [Kim et al., 2023] e RESTGPT [Kim et al., 2024], busca extrair informações semânticas de especificações de API em linguagem natural para gerar testes mais contextuais e significativos. Embora esses estudos se concentram principalmente na comparação com outras técnicas de IA ou abordagens baseadas em conhecimento, a motivação subjacente é superar as limitações das ferramentas tradicionais que se baseiam apenas na estrutura formal das especificações, sugerindo um potencial para gerar testes que vão além do que outras ferramentas específicas para APIs REST podem produzir sem essa compreensão semântica [Kim, 2023,2024]

5 Conclusão e Trabalhos Futuros

O presente estudo enfatizou que as fontes analisadas oferecem uma visão abrangente do campo da testagem de APIs REST impulsionada por IA, destacando a sua crescente relevância e os múltiplos desafios inerentes. A pesquisa tem explorado diversas abordagens para automatizar a geração de casos de teste, incluindo o uso de algoritmos evolutivos, técnicas de caixa-preta e caixa-branca, e mais recentemente, o aprendizado por reforço e grandes modelos de linguagem (LLMs). Estudos empíricos comparam a eficácia de várias ferramentas, revelando a necessidade de melhorias na cobertura de código, detecção de falhas e tratamento de dependências entre operações, além da importância de métricas de avaliação específicas do domínio.

Os resultados evidenciaram que os LLMs podem melhorar a cobertura e a consistência dos testes, além de reduzir o esforço manual. No entanto, sua eficácia varia conforme o contexto e a escala de aplicação. Além disso, comparações com abordagens manuais ou ferramentas específicas demonstram que os LLMs têm vantagens significativas em termos de eficiência e capacidade de adaptação a cenários complexos, mas ainda enfrentam limitações em domínios altamente especializados.

6 Referências

[1] Corradini, D., Montolli, Z., Pasqua, M. and Ceccato, M. (2024) "DeepREST: Automated Test Case Generation for REST APIs Exploiting Deep Reinforcement Learning", In: Proceedings of the ACM Conference, ACM, https://doi.org/10.1145/3691620.3695511.

[2] Gao, C., Hu, X., Gao, S., Xia, X. and Jin, Z. (2025) "The Current Challenges of Software Engineering in the Era of Large Language Models", In: Proceedings of the ACM Conference, ACM, https://doi.org/10.1145/3712005.

[3] Kim, M., Corradini, D., Sinha, S., Orso, A., Pasqua, M., Tzoref-Brill, R. and Ceccato, M. (2023) "Enhancing REST API Testing with NLP Techniques", In: Proceedings of the ACM Conference, ACM, https://doi.org/10.1145/3597926.3598131.

[4] Kim, M., Sinha, S. and Orso, A. (2024) "Adaptive REST API Testing with Reinforcement Learning", In: Proceedings of the ACM Conference, ACM, https://doi.org/10.1109/ASE56229.2023.00218.

[5] Kim, M., Stennett, T., Shah, D., Sinha, S. and Orso, A. (2024) "Leveraging Large Language Models to Improve REST API Testing", In: Proceedings of the ACM Conference, ACM, https://doi.org/10.1145/3639476.3639769.

[6] Kim, M., Xin, Q., Sinha, S. and Orso, A. (2022) "Automated Test Generation for REST APIs: No Time to Rest Yet", In: Proceedings of the ACM Conference, ACM, https://doi.org/10.1145/3533767.3534401.

[7] Konuk, M., Baglum, C. and Yayan, U. (2024) "Evaluation of Large Language Models for Unit Test Generation", In: Proceedings of the IEEE Conference, IEEE, https://ieeexplore.ieee.org/document/10756954/.

[8] Laaber, C., Yue, T., Ali, S., Schwitalla, T. and Nygård, J. F. (2023) "Automated Test Generation for Medical Rules Web Services: A Case Study at the Cancer Registry of Norway", In: Proceedings of the ACM Conference, ACM, https://doi.org/10.1145/3611643.3613882.

[9] Leu, B., Volken, J., Kropp, M., Dogru, N., Anslow, C. and Biddle, R. (2024) "Reducing Workload in Using AI-based API REST Test Generation", In: Proceedings of the IEEE Conference, IEEE, https://ieeexplore.ieee.org/document/10556480/.

[10] Martin-Lopez, A. (2020) "AI-Driven Web API Testing", In: Proceedings of the IEEE Conference, IEEE, https://ieeexplore.ieee.org/document/9270322/.

[11] Mezzaro, S., Gambi, A. and Fraser, G. (2024) "An Empirical Study on How Large Language Models Impact Software Testing Learning", In: Proceedings of the ACM Conference, ACM, https://doi.org/10.1145/3661167.3661273.

[12] Olsthoorn, M. (2022) "More Effective Test Case Generation with Multiple Tribes of AI", In: Proceedings of the ACM Conference, ACM, https://doi.org/10.1145/3510454.3517066.