

Teste de aceitação gerado por Inteligência Artificial: um estudo exploratório sobre seu potencial de uso por analistas de teste

Caio Jordan de Lima Maia



CENTRO DE INFORMÁTICA
UNIVERSIDADE FEDERAL DA PARAÍBA

João Pessoa, 2023

Caio Jordan de Lima Maia

Teste de aceitação gerado por Inteligência Artificial: um estudo exploratório sobre seu potencial de uso por analistas de teste

Artigo apresentado ao curso Ciência da
Computação do Centro de Informática, da Universidade Federal
da Paraíba, como requisito para a obtenção do grau de Bacharel
em Ciência da Computação

Orientadora: Yuska Paola Costa Aguiar

Novembro de 2023

Catálogo na publicação
Seção de Catalogação e Classificação

M217t Maia, Caio Jordan de Lima.

Teste de aceitação gerado por inteligência artificial: um estudo exploratório sobre seu potencial de uso por analistas de teste / Caio Jordan de Lima Maia. - João Pessoa, 2023.
20 f. : il.

Orientação: Yuska Aguiar.
TCC (Graduação) - UFPB/CI.

1. Teste de aceitação. 2. Plano de teste. 3. Garantia de qualidade. 4. Inteligência artificial. 5. ChatGPT. 6. Bard. I. Aguiar, Yuska. II. Título.

UFPB/CI

CDU 004.8



CENTRO DE INFORMÁTICA
UNIVERSIDADE FEDERAL DA PARAÍBA

Trabalho de Conclusão de Curso de Ciência da Computação intitulado **Teste de aceitação gerado por Inteligência Artificial: um estudo exploratório sobre seu potencial de uso por analistas de teste** de autoria de Caio Jordan de Lima Maia, aprovada pela banca examinadora constituída pelos seguintes professores:

Prof. Dr. Yuska Paola Costa Aguiar
Universidade Federal da Paraíba

Prof. Dr. Danielle Rousy Dias Ricarte
Universidade Federal da Paraíba

Prof. Dr. Yuri de Almeida Malheiros Barbosa
Universidade Federal da Paraíba

João Pessoa, 7 de novembro de 2023

Teste de aceitação gerado por Inteligência Artificial: um estudo exploratório sobre seu potencial de uso por analistas de teste

Caio J. Lima Maia¹, Yuska P. Costa Aguiar¹

¹Centro de Informática – Universidade Federal da Paraíba (UFPB)

caio.maia@cc.ci.ufpb.br, yuska@ci.ufpb.br

Abstract. *This article explores the use of AI-based natural language processing models, such as ChatGPT and Bard, in the generation of software acceptance test cases to optimize this process. It compares the coverage of tests created manually by professionals with different levels of experience in software quality and the automatic generation of test cases by AI tools. The results show that collaboration between AI and senior QA professionals can optimize the process, offering efficiency and speed in the creation of test cases.*

Key words: *Acceptance Test. Test Plan. Quality Assurance. Artificial Intelligence. ChatGPT. Bard.*

Resumo. *Este artigo explora o uso de modelos de processamento de linguagem natural baseados em IA, como ChatGPT e Bard, na geração de casos de testes de aceitação de software para a otimização desse processo. É comparado a cobertura de testes criados manualmente por profissionais com diferentes níveis de experiência em qualidade de software e a geração automática de casos de teste pelas ferramentas de IA. Resultados demonstram que a colaboração entre IA e profissionais de QA sênior pode otimizar o processo, oferecendo eficácia e rapidez na criação de casos de teste.*

Palavras-chave: *Teste de Aceitação. Plano de Teste. Garantia de Qualidade. Inteligência Artificial. ChatGPT. Bard.*

1 Introdução

A engenharia de software engloba uma ampla gama de atividades que abrangem todo o ciclo de produção de software, incluindo a especificação, desenvolvimento, validação e evolução do software (Sommerville, 2011). A validação do software é uma etapa de extrema importância que perpassa todo o processo de desenvolvimento, focalizando a garantia da qualidade do produto, dos envolvidos e do próprio processo em si. Um processo de desenvolvimento de alta qualidade desempenha um papel crucial na redução de defeitos, no encurtamento de prazos e na entrega eficiente do software (Pressman, 2016). A realização de testes é uma prática vital ao longo de todo esse processo, e a incorporação de uma abordagem de gestão de qualidade desde as fases iniciais, como o planejamento, possibilita a identificação de problemas antes mesmo de sua implementação.

Na fase inicial do ciclo de desenvolvimento de software, a de levantamento de requisitos, também são delineados os testes de aceitação. Segundo a definição de Sommerville (2011), o teste de aceitação é "um processo de teste de usuário no qual o objetivo é decidir se o software é bom o suficiente para ser implantado e usado". Por

meio dele, é possível validar se o software se comporta conforme o esperado pelos usuários, de acordo com as entradas fornecidas.

A criação de casos de teste de aceitação assume um papel vital nesse processo de testes de software. Esses casos oferecem uma estrutura organizada para o que precisa ser testado, delineando os fluxos de execução dos testes e estabelecendo os resultados esperados. Além disso, contribuem para uma compreensão mais clara do sistema por parte da equipe de desenvolvimento.

A adoção de metodologias ágeis trouxe mudanças significativas na forma como os casos de teste são elaborados. Atualmente, muitos projetos optam pelo Desenvolvimento Guiado por Comportamento (BDD), uma abordagem que se concentra na compreensão e especificação do comportamento desejado de um sistema antes mesmo da escrita de código [3]. Essa abordagem envolve a colaboração entre desenvolvedores, testadores e partes interessadas, como os próprios usuários ou clientes, para criar cenários de teste descritos em linguagem natural, conhecidos como "especificações". Essas especificações descrevem o comportamento esperado do sistema em termos de entradas, saídas e interações.

No contexto atual, os Testes de Aceitação desempenham um papel fundamental na garantia de que o software atenda aos requisitos dos usuários e do sistema. No entanto, a criação e manutenção desses testes frequentemente demandam um investimento considerável de tempo, esforço e expertise por parte dos profissionais de Qualidade (QA).

Não é surpresa que, com o surgimento de ferramentas que fazem uso de Inteligência Artificial (IA), tenham surgido métodos para simplificar e acelerar o processo de desenvolvimento de software. Ferramentas como o ChatGPT (OpenAI, 2022) [1] e o Bard (Google, 2023) [2] despertam a curiosidade sobre como podem ter um impacto positivo na área de qualidade de software.

Neste artigo, exploraremos a possibilidade de utilizar essas ferramentas para a criação de casos de teste em Testes de Aceitação. O objetivo é avaliar se essas ferramentas podem ser úteis na criação dos casos de teste de forma que mantenham a qualidade e a cobertura dos testes, ao mesmo tempo em que reduzem o tempo e o esforço necessários para a criação desses testes. Com isso, podendo auxiliar o QA em seu trabalho, e dando uma possibilidade de otimizar o seu trabalho.

2 Fundamentação Teórica

2.1. Testes de Aceitação

Conforme definido pelo ISTQB no Syllabus(2023), nos testes de aceitação, os casos de teste são criados com base nos critérios de aceitação, servindo como exemplificações do funcionamento do software. Isso desempenha um papel fundamental na garantia de que a equipe implemente com precisão a história do usuário. Os casos de teste devem abranger todas as funcionalidades descritas na história do usuário, sem ultrapassá-las. No entanto, é importante observar que os critérios de aceitação podem detalhar certos aspectos específicos da história do usuário. Adicionalmente, é

recomendável evitar que dois casos de teste descrevam as mesmas funcionalidades da história do usuário, garantindo assim uma cobertura abrangente[Syllabus 2023, p.49].

2.2 Criação de Casos de Teste e sua Importância

A criação de casos de teste é um processo no qual definimos cenários que serão executados para verificar se um requisito do sistema funciona conforme esperado. Eles podem ter pré-condições, um passo-a-passo de como realizar o teste, e até pós condições para finalizar o caso, além disso ele também fornece entradas a serem utilizadas e saídas esperadas.

Com uma criação adequada destes casos podemos garantir a cobertura dos requisitos, uma documentação para as partes interessadas pois facilita a compreensão dos requisitos e dos comportamentos esperados, além de garantir um rastreio no caso de surgir algum “bug”, podendo facilitar a reprodução do teste. Com isso também é garantido um aumento de confiança nos testes pelas partes interessadas, já que demonstra que o sistema funciona conforme o esperado.

2.3 Padrão internacional ISO 29119-3

A ISO 29119-3 fornece diretrizes gerais para planejar, projetar e executar casos de teste funcionais. Ela tem como benefício a consistência já que fornece um padrão para a documentação, ele também é mais reduzido e claro de entender, além de ser um padrão reconhecido internacionalmente. O template contempla os seguintes elementos: identificador do caso de teste, descrição, pré-condição, passos do teste, resultados esperados. Apesar disso, o template também pode variar de acordo com o projeto ou organização, para se adaptar às necessidades específicas, tornando mais flexível. Um exemplo de um caso de teste utilizando o padrão pode ser visto na Tabela 1. [12]

Caso de Teste	Pré-Condição	Passos	Objetivos	Resultado Esperado
1.Listar Tarefas	Ter tarefas	1. Ir para tela de tarefas 2. Clicar no botão de Listar Tarefas	Validar a funcionalidade de listar tarefas	Aparecerá uma lista de tarefas existentes

2.4 ChatGPT e Bard

Tanto o ChatGPT da OpenAI quanto o Bard da Google são modelos de processamento de linguagem natural que viabilizam a criação de diálogos mediante o uso de prompts, gerando respostas de forma artificial. A partir deles, é possível atender a demandas por textos e informações. [5, 6]

A principal distinção entre esses sistemas reside na fonte de dados que os alimenta. Enquanto o ChatGPT utiliza um banco de dados gerido pela própria OpenAI, o Bard se vale do Google como fonte para buscar informações [6]. É importante salientar que uma diferença crucial entre os dois é a atualização de seus dados. Enquanto a versão gratuita do ChatGPT dispõe de informações até janeiro de 2022, o Bard gratuito é regularmente atualizado e é capaz de acessar e processar informações em tempo real por meio do Google, o que o torna mais atualizado em comparação com o ChatGPT [1, 2].

2.5 Aplicação da IA no Contexto de Teste Software

A aplicação da Inteligência Artificial (IA) no contexto de teste de software representa uma área de pesquisa e desenvolvimento em crescimento dentro da engenharia de software [7]. A IA é uma disciplina que visa criar sistemas capazes de realizar tarefas que normalmente exigiriam inteligência humana, e seu uso no teste de software oferece oportunidades significativas para melhorar a eficiência e a eficácia dos processos de teste.

A inteligência artificial (IA) tem desempenhado um papel cada vez mais significativo no campo de testes, com diversas aplicações emergentes, incluindo a automação da geração de testes, detecção de defeitos e criação de casos de teste [8, 9]. Essa área tem experimentado um desenvolvimento impulsionado por novas pesquisas e aplicações inovadoras. Por exemplo, no artigo de Xie et al. (2023) é demonstrado como o ChatGPT pode ser empregado de forma eficaz na geração automatizada de testes unitários. Porém não foi possível encontrar pesquisas mostrando se essas ferramentas seriam úteis na geração de casos de teste para testes de aceitação, de forma que mantivesse a qualidade e a cobertura dos testes.

3 Trabalhos Relacionados

Com o objetivo de compreender a viabilidade deste estudo, foram realizadas buscas por pesquisas relacionadas que possuísem objetivos semelhantes. Utilizamos palavras-chave como "ChatGPT," "Bard", "acceptance testing", "Test-cases," e "Generation," priorizando resultados mais recentes (até 2023), considerando que tanto o ChatGPT (2022) [1] quanto o Bard (2023) [2] são ferramentas relativamente novas. Essa abordagem resultou em uma seleção limitada de apenas quatro estudos correlatos. Não foi possível identificar uma pesquisa com objetivos idênticos a este, no entanto, encontramos investigações que exploravam o uso dessas ferramentas no contexto de teste de software, por exemplo, a pesquisa de Talasbek (2023) demonstrou a utilização do ChatGPT tanto na criação de casos de teste de aceitação quanto na geração de código para automatização desses testes, porém não houve uma comparação do trabalho feito pela IA com testers humanos.

Outros estudos identificados estavam concentrados na aplicação do ChatGPT na geração de código de testes unitários. No entanto, o foco deste trabalho é a comparação entre a criação de casos de teste realizada por indivíduos com diferentes níveis de experiência na área e a geração de casos feita pelo ChatGPT e pelo Bard. Nosso objetivo é avaliar a viabilidade e eficácia dessas abordagens, visando reduzir o tempo gasto nesta fase do ciclo de testes e minimizar a carga de trabalho dos testadores.

Buscando demonstrar a diferença dos objetivos dos artigos relacionados e a proposta deste artigo, a Tabela 2 apresenta informações e resultados de cada trabalho.

Tabela 2. Trabalhos Relacionados.

Ano	Título	Tipo de Teste	Metodologia
2023	ARTIFICIAL AI IN TEST AUTOMATION: SOFTWARE TESTING OPPORTUNITIES WITH OPENAI TECHNOLOGY-CHAT GPT [8]	Teste de Aceitação	Geração de casos de teste e de código para automação dos testes
2023	ChatUniTest: a ChatGPT-based automated unit test generation tool [9]	Teste Unitário	Uso do ChatUniTest como complemento ao ChatGPT para geração dos testes
2023	Nuances are the Key: Unlocking ChatGPT to Find Failure-Inducing Tests with Differential Prompting. In: Proceedings of the 45th ACM/IEEE International Conference on Software Engineering [10]	Teste Unitário	Uso do ChatGPT para geração de entradas para casos de teste que induzam falhas no código
2023	ChatGPT vs SBST: A Comparative Assessment of Unit Test Suite Generation [11]	Teste Unitário	Uma comparação da geração de casos de teste unitários entre ChatGPT e SBST
2023	Teste de aceitação gerado por Inteligência Artificial: um estudo exploratório sobre seu potencial de uso por analistas de teste	Teste de Aceitação	Uma comparação entre a geração de casos de teste de aceitação criados por modelos de processamento de linguagem natural (Bard e ChatGPT) com humanos da área de qualidade software (testadores)

Como é possível ver logo acima, este artigo se diferencia dos outros pelo fato de realizar uma comparação entre a geração dos casos de teste de aceitação criados por humanos e gerados por IAs, tendo como principal objetivo verificar se será útil como ferramenta para o QA, de forma que o mesmo diminua o tempo e o esforço gasto com a criação desses testes.

4. Procedimentos Metodológicos

A pesquisa descrita neste artigo teve como objetivo de fazer um estudo de viabilidade da utilização de modelos de processamento de linguagem natural baseados em inteligência artificial como ferramentas para criação de casos de teste de forma que verifique se o processo de especificação será mais rápido e se pode cobrir mais casos que talvez o QA não tenha visto. Para sua execução, a metodologia aplicada consistiu no conjunto das seguintes etapas:

(i) solicitar a duas pessoas, uma delas iniciante na área de qualidade de software (com menos de 1 ano trabalhando na área) e a outra com maior senioridade (tendo cargo de senior e trabalhando na área a cerca de 4 anos), a criação de casos de teste para uma funcionalidade que será descrita para eles como requisito funcional e também será pedido para utilizarem o padrão estabelecido pela ISO 29119-3 [12] na qual foi-lhes mostrado um template onde só é preciso colocar o título do teste, os passos, o objetivo e o resultado esperado, a mesma descrição do requisito será utilizada: "Como uma loja virtual, eu devo ter a funcionalidade de colocar, retirar ou finalizar compras no meu carrinho, lembrando que não posso finalizar um carrinho vazio e ao adicionar um mesmo produto ao carrinho não deve haver duplicações de produtos, será adicionado à sua quantidade no carrinho";

(ii) realizar um segundo ciclo de levantamento de casos de teste, mas desta vez utilizando dois modelos de linguagem natural baseados em inteligência artificial: o ChatGPT e o Bard (usando as versões gratuitas mais atuais disponíveis até a data presente), utilizando como entrada a mesma descrição da funcionalidade anterior, com o adicional da solicitação antes da descrição: "Crie casos de teste funcionais de acordo com o ISO-29119-3 para a seguinte funcionalidade:", e realizar 3 gerações (em cada modelo), selecionando aquele que melhor atender ao padrão pré definido dos casos de teste;

(iii) realizar análise da estruturação dos casos, verificando se estão organizados de maneira lógica e clara, seguindo o padrão estabelecido; (iv) realizar comparação entre o gabarito dos casos de teste com os testes criados por humanos e os gerados pelo ChatGPT e Bard, verificando a quantidade de casos, a cobertura.

Durante a experiência, todos os participantes (incluindo o Bard e o ChatGPT), também terão o padrão estabelecido pelo ISO 29119-3, no qual é dado um formato específico para a geração dos casos de teste, para que assim seja possível avaliar melhor os casos gerados. Com isso, esse processo de comparação entre casos de teste elaborados por pessoas com diferentes níveis de experiência, bem como por modelos de IA, permitirá uma análise abrangente da qualidade dos casos gerados.

5. Análise dos Resultados

A pesquisa descrita neste artigo teve como objetivo realizar um estudo exploratório da utilização de modelos de processamento de linguagem natural baseados em inteligência artificial como ferramentas para a criação de casos de teste, com o intuito de agilizar o processo de planejamento e aumentar a cobertura de testes, abrangendo casos que poderiam ter passado despercebidos pelos testadores de qualidade (QA). Importante ressaltar que não foi passado nenhum limite ou quantidade mínima de casos a serem criados, sendo os mesmos responsáveis por decidir se os que foram criados já serviram para cobrir todos os cenários. Seguindo a metodologia descrita anteriormente temos os seguintes resultados:

Resultado do Levantamento Inicial de Casos de Teste (Etapa i):

Ambos os testadores levantaram os casos de testes com as mesmas informações passadas, e demandaram um tempo considerável, cerca de 1 hora para levantar 6 casos no caso de um iniciante na área como pode ser visto na Figura 2 e 30 minutos no caso de um QA Senior para levantar a mesma quantidade de testes [Figura 3].

Resultados da Utilização de Modelos de Linguagem Natural (Etapa ii):

Na segunda etapa, empregamos os modelos de linguagem natural baseados em inteligência artificial, o ChatGPT e o Bard, para gerar casos de teste. Foi utilizado em ambos a seguinte entrada: Crie casos de teste funcionais de acordo com o ISO-29119-3 para a seguinte funcionalidade: "Como uma loja virtual, eu devo ter a funcionalidade de colocar, retirar ou finalizar compras no meu carrinho, lembrando que não posso finalizar um carrinho vazio e ao adicionar um mesmo produto ao carrinho não deve haver duplicações de produtos, será adicionado à sua quantidade no carrinho". As saídas desta etapa incluíram os casos de teste criados pelos modelos de IA, ChatGPT [Figura 4] e Bard [Figuras 5, 6 e 7]. É igualmente crucial recordar que, em ambos os casos, realizou-se a geração de resultados em três repetições, selecionando aquele que melhor atendesse ao padrão predefinido dos casos de teste.

Análise da Estruturação dos Casos de Teste (Etapa iii):

Tanto o Bard quanto o ChatGPT conseguiram desenvolver os casos de teste de forma que fosse possível compreender o funcionamento de cada teste, seguindo precisamente o padrão estabelecido pela ISO 29119-3. No entanto, é importante ressaltar que foi necessário realizar três iterações para que o ChatGPT conseguisse aderir ao padrão de geração de casos de teste, o qual abrange pré-condições, passos, objetivos e resultados esperados. Em contrapartida, o Bard demonstrou a capacidade de gerar os casos de teste de maneira correta logo na primeira iteração. Não obstante, ambos os modelos conseguiram estabelecer a estrutura adequada para a criação de casos de teste, seguindo o padrão e garantindo que os profissionais de garantia de qualidade possam compreendê-la com clareza.

Comparação dos Casos de Teste (Etapa iv):

Na tabela a seguir [Tabela 3], apresentamos uma comparação dos casos de teste gerados com base na quantidade, avaliando se abrangem todos os cenários, incluindo a

comparação com os cenários do gabarito [Figura 1]. Além disso, verificamos se os casos criados aderem às regras do requisito, tais como "finalizar carrinho com produto", "finalizar carrinho sem produto", "adicionar produto no carrinho", "retirar produto do carrinho" e "acrescentar quantidade de produto no carrinho". Também registramos o tempo gasto na criação desses casos de teste.

Tabela 3. Comparação da geração dos casos de teste.

	Quantidade Casos de teste	Cobre todos os cenários?	Valida todas as regras do requisito?	Tempo gasto na criação
Gabarito	9	Sim	Sim	-
QA Junior	6	Não	Não	1 hora
QA Senior	6	Não	Sim	30 minutos
Bard	5	Não	Não	< 1 min
ChatGPT	5	Não	Sim	< 1 min

A partir da tabela, fica evidente que o ChatGPT supera um profissional júnior de garantia de qualidade (QA Junior) na criação de casos de teste. Mesmo gerando um caso a menos, o ChatGPT conseguiu validar todas as regras do requisito proposto em menos de 1 minuto, em contraste com a uma hora gasta pelo QA Junior. Tanto o Bard quanto o QA Junior não conseguiram criar casos de teste que abarcassem todas as regras do requisito. No caso do Bard, não foi gerado um caso de teste que abrangesse a finalização do carrinho com um produto, enquanto o profissional júnior não criou um caso de teste que validasse a finalização do carrinho vazio.

Além disso, é relevante fazer uma comparação da redação dos casos de teste [Figuras 2, 3, 4, 5, 6 e 7]. Embora os objetivos e resultados esperados tanto dos seres humanos quanto dos modelos de processamento de linguagem natural sejam semelhantes, é perceptível que os seres humanos forneceram mais detalhes nos "passos" a serem seguidos, como "realizar uma pesquisa pelo produto". Isso pode ocorrer devido ao fato de que, no processo de teste (TDD), os seres humanos têm maior liberdade para imaginar como o teste será executado, uma vez que o teste é criado antes do desenvolvimento. Além disso, também é notável que o ChatGPT e o Bard criaram pré-condições que não foram diretamente fornecidas a eles, tais como "o usuário estar logado" ou "o produto estar disponível". Isso se deve ao fato de que eles se baseiam em requisitos semelhantes aos já existentes para a criação dos casos de teste, demonstrando a capacidade desses modelos de aprender e aplicar conhecimento prévio.

Portanto, podemos concluir que o ChatGPT demonstrou eficácia na criação de casos de teste, tornando seu uso interessante para a geração dos mesmos, desde que se disponham de requisitos bem definidos. Além disso, é vantajoso empregar o ChatGPT em conjunto com um profissional de garantia de qualidade (QA), permitindo ao QA

sênior avaliar os casos gerados e utilizar o tempo economizado na criação desses casos para explorar mais cenários, assim ampliando a cobertura dos testes. Enquanto para o QA júnior, essa abordagem pode servir como uma oportunidade de aprendizado para criar casos de teste abrangentes que cobrem todas as regras do requisito, contribuindo para seu desenvolvimento profissional na área de qualidade.

6. Considerações Finais

Neste artigo, apresentamos casos de teste desenvolvidos por dois profissionais de QA com experiências distintas e comparamos esses casos aos gerados por duas ferramentas: ChatGPT e Bard. Nosso objetivo foi avaliar a eficácia dessas ferramentas em gerar casos de teste e determinar sua viabilidade. Com base em nossa experiência, concluímos que é, de fato, possível gerar casos de teste eficazes com o ChatGPT. No entanto, é importante ressaltar que com a constante evolução desses modelos, é provável que tanto o ChatGPT quanto o Bard se tornem ainda mais proficientes na cobertura de casos de teste no futuro.

Além disso, é interessante explorar o uso dessas ferramentas no contexto do aprendizado, uma vez que elas podem ser empregadas como ferramentas auxiliares para o QA Junior na compreensão dos requisitos e na criação dos casos de teste.

Essa abordagem nos permite concluir que essas ferramentas podem ser utilizadas com êxito como auxílio na criação de casos de teste. Além de sua praticidade e rapidez, é notável que enquanto os profissionais de QA levaram de 30 a 60 minutos para criar um número limitado de cenários, as IAs foram capazes de gerá-los em questão de segundos. Portanto, podemos afirmar que essas ferramentas representam uma adição valiosa ao arsenal de recursos disponíveis para o trabalho de QA, oferecendo uma eficiência significativa no processo de criação de casos de teste.

Referências

- [1] ChatGPT. (2022). Home. <https://openai.com/blog/chatgpt/>. Acesso em 25 de setembro de 2023
- [2] Bard. (2023). Home. <https://bard.google.com>. Acesso em 25 de setembro de 2023
- [3] SMART, John Ferguson; MOLAK, Jan. BDD in Action, Second Edition: Behavior-Driven Development for the Whole Software Lifecycle. Manning Publications, 2022.
- [4] SYLLABUS. (2023). Brazilian Software Testing Qualifications Board - Tradução realizada pelo GT Traduções do BSTQB do syllabus do ISTQB, https://bcr.bstqb.org.br/docs/syllabus_ctfl_4.0br.pdf. Acesso em: 29 out. 2023.
- [5] FIA BUSINESS SCHOOL. O que é Chat GPT, impactos e como usar esta inteligência artificial?. Disponível em: <https://fia.com.br/blog/chat-gpt/>. Acesso em: 30 mar. 2023.
- [6] FIA BUSINESS SCHOOL. O que é Bard e como usar a inteligência artificial do Google?. Disponível em: <https://fia.com.br/blog/bard/>. Acesso em: 27 mar. 2023.

- [7] Hourani, H., Hammad, A., & Lafi, M. (2019). The Impact of Artificial Intelligence on Software Testing. In 2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT) (pp. 565-570). IEEE.
- [8] Talasbek, A. ARTIFICIAL AI IN TEST AUTOMATION: SOFTWARE TESTING OPPORTUNITIES WITH OPENAI TECHNOLOGY-CHATGPT. Suleyman Demirel University, 2023
- [9] XIE, Z. et al. ChatUniTest: a ChatGPT-based automated unit test generation tool. arXiv (Cornell University), 8 maio 2023.
- [10] Li, T. et al. Nuances are the Key: Unlocking ChatGPT to Find Failure-Inducing Tests with Differential Prompting. In: Proceedings of the 45th ACM/IEEE International Conference on Software Engineering (ASE 2023). New York, NY, USA: Association for Computing Machinery, 2023. p. 60-71.
- [11] TANG, Y. et al. ChatGPT vs SBST: A Comparative Assessment of Unit Test Suite Generation. arXiv preprint arXiv:2307.00588, 2023.
- [12] ISO/IEC/IEEE 29119-3:2021. (2021). 13-14 de 17. Disponível em: <<https://www.iso.org/standard/79429.html>>. Acesso em: 2 out. 2023.

Anexo

Casos de teste	Pré-Condição	Passos	Objetivos	Resultado esperado
1. Colocar um produto em um carrinho	Não tenha produtos no carrinho	1. Procuro por um produto 2. Clico em "Adicionar ao carrinho" 3. Vou para o carrinho 4. O produto é exibido no carrinho	Validar que é possível adicionar um produto no carrinho	Que o produto é adicionado ao carrinho com sucesso com quantidade 1
2. Retirar um produto do carrinho	Possua um produto no carrinho	1. Vou para o carrinho 2. Selecciono um produto do carrinho 3. Clico em remover produto do carrinho 4. O carrinho fica vazio	Validar que é possível remover um produto do carrinho	O carrinho fica vazio
3. Finalizar carrinho com um produto	Possua um produto no carrinho	1. Vou para o carrinho 2. Selecciono um produto do carrinho 3. Clico em finalizar o carrinho 4. O carrinho fica vazio	Validar que é possível finalizar um carrinho com um produto com sucesso	Que o usuário finalize com sucesso o carrinho constando um produto e o carrinho fique vazio
4. Tentar Finalizar carrinho vazio	Não tenha produtos no carrinho	1. Abro a página do carrinho 2. Clico em finalizar o carrinho	Validar que não é possível finalizar um carrinho vazio	Deve informar que não é possível finalizar um carrinho vazio
5. Adicionar um produto diferente no carrinho	Já possui um produto no carrinho	1. Procuro por um produto diferente do que já está no carrinho 2. Clico em "Adicionar ao carrinho" 3. Vou para o carrinho 4. Ambos os produtos são exibidos no carrinho	Validar que é possível adicionar produtos diferentes no carrinho	Que o usuário adicione com sucesso um outro produto no carrinho e que ambos sejam exibidos
6. Colocar dois produtos iguais no carrinho	Já possui um produto no carrinho	1. Procuro pelo mesmo produto que já está no carrinho 2. Clico em "Adicionar ao carrinho" 3. Vou para o carrinho 4. O produto está presente no carrinho com quantidade 2	Validar que é possível adicionar um mesmo produto ao carrinho sem que ele fique duplicado, acrescentando somente a sua quantidade	Que o usuário adicione com sucesso dois produtos iguais, e que a quantidade dele aumente no carrinho, sem haver duplicação
7. Diminuir a quantidade de um produto do carrinho	Já possui um produto com quantidade 2 no carrinho	1. Abro a página do carrinho 2. Selecciono o produto 3. Diminuo sua quantidade em 1 4. Atualizo a página do carrinho	Validar que a contagem dos produtos é atualizada com sucesso no carrinho	Que o usuário consiga diminuir a quantidade de um produto do carrinho
8. Finalizar carrinho com produtos diferentes	Possua dois produtos diferentes no carrinho	1. Vou para o carrinho 3. Clico em finalizar o carrinho 4. O carrinho fica vazio	Validar que é possível finalizar um carrinho com produtos diferentes com sucesso	Que o usuário finalize com sucesso o carrinho constando os dois produtos diferentes e o carrinho fique vazio
9. Finalizar carrinho com um mesmo produto com quantidade maior que 1	Possua um produto no carrinho com quantidade 2	1. Vou para o carrinho 3. Clico em finalizar o carrinho 4. O carrinho fica vazio	Validar que é possível finalizar um carrinho com um produto com quantidade maior que 1 com sucesso	Que o usuário finalize com sucesso o carrinho constando a mesma quantidade do produto

Figura 1: Gabarito dos casos de teste

Casos de teste	Pré-condições	Passos	Objetivos	Resultado esperado
1. Validar funcionalidade de adicionar produtos diferentes ao carrinho	Estar com carrinho vazio	1. Busco um produto desejado na barra de pesquisa 2. Clique no produto desejado 3. Preencho o campo "quantidade" e demais campos personalizáveis 4. Clique em "Adicionar ao carrinho" 5. Busco por outro produto desejado na barra de pesquisa 6. Clique no produto desejado 7. Preencho o campo "quantidade" e demais campos personalizáveis 8. Clique em "Adicionar ao carrinho"	Validar que a funcionalidade adicionará ao carrinho produtos diferentes	Que o usuário possa adicionar produtos diferentes ao carrinho, separando-os na página do carrinho por produto e mostrando os preços individuais, além do valor total da soma dos preços de todos os itens adicionados
2. Validar funcionalidade de adicionar o mesmo produto ao carrinho	Estar com carrinho vazio	1. Busco um produto desejado na barra de pesquisa 2. Clique no produto desejado 3. Preencho o campo "quantidade" e demais campos personalizáveis 4. Clique em "Adicionar ao carrinho" 5. Na mesma tela do produto, altero os campos personalizáveis 6. Clique em "Adicionar ao carrinho"	Validar que a funcionalidade adicionará ao carrinho o produto desejado, alterando apenas a quantidade do mesmo	Que o usuário possa adicionar um mesmo produto ao carrinho, alterando na página do carrinho apenas a sua quantidade desejado e indicando nos detalhes do produto as especificações escolhidas pelo cliente
3. Validar funcionalidade de remover produto do carrinho - produtos diferentes e em quantidade única	Carrinho com produtos diferentes	1. Abro a página do carrinho 2. Seleciono a(s) checkbox(s) indicando quais produtos desejo selecionar 3. Clique em "remover produto(s) do carrinho" 4. Clique para confirmar remoção do(s) produto(s) do carrinho	Validar que a funcionalidade removerá do carrinho o(s) produto(s) desejado(s), mantendo os outros no carrinho	Que o usuário possa remover um ou mais produtos diferentes do carrinho de uma vez só
4. Validar funcionalidade de remover parcialmente um produto do carrinho - um mesmo produto com quantidade acima de 1	Carrinho com um mesmo produto quantidade acima de 1	1. Abro a página do carrinho 2. Reduzo a quantidade do produto que desejo alterar (0 para excluir produto) 3. Clique para confirmar alteração da quantidade do produto	Validar que a funcionalidade irá remover o produto quando a quantidade for reduzida para 0 e que também poderá apenas reduzir as quantidades previamente selecionadas pelo usuário	Que o usuário possa reduzir a quantidade ou remover o produto do carrinho
5. Validar funcionalidade de finalizar compra - diretamente da página do produto	Estar com carrinho vazio	1. Busco um produto desejado na barra de pesquisa 2. Clique no produto desejado 3. Preencho o campo "quantidade" e demais campos personalizáveis 4. Clique em "Finalizar compra" 5. Seleciono a forma de pagamento 6. Preencho os campos necessários para efetuar pagamento 7. Confirmando a compra	Validar que a funcionalidade dará a possibilidade de finalizar a compra diretamente da página do produto, sem passar pelos itens que constam do carrinho	Que o usuário possa finalizar compra de um único produto através da página do produto, sem contabilizar os itens que constam no carrinho
6. Validar funcionalidade de finalizar compra - na página do carrinho	Carrinho com produtos	1. Abro a página do carrinho 2. Clique em finalizar compra 3. Seleciono quais produtos desejo finalizar compra e confirmo 4. Seleciono a forma de pagamento 5. Preencho os campos necessários para efetuar pagamento 6. Confirmando a compra	Validar que a funcionalidade utilizada na página do carrinho irá somar os preços dos produtos do carrinho que o usuário deseja finalizar a compra	Que o usuário possa escolher os produtos de seu carrinho para efetuar a finalização da compra

Figura 2: Casos de teste levantados pelo QA iniciante na área

Casos de Teste	Pré-condições	Passos	Objetivos	Resultado Esperado
1. Validar adição de produto com sucesso ao carrinho	Carrinho vazio	1. Pesquiso um produto na barra de pesquisa 2. Clico no produto escolhido 3. Seleciono a quantidade desejada 4. Clico em "Adicionar ao Carrinho"	Validar que é possível adicionar um produto ao carrinho	O produto selecionado deve ser adicionado ao carrinho com quantidade igual a quantidade informada pelo usuário na seleção do produto
2. Validar adição de mais de um produto ao carrinho	Possuir um produto no carrinho	1. Pesquiso um produto diferente do carrinho na barra de pesquisa 2. Clico no produto escolhido 3. Seleciono a quantidade desejada 4. Clico em "Adicionar ao Carrinho"	Validar que é possível adicionar mais de um produto ao carrinho	Os produtos selecionados devem ser adicionados ao carrinho com quantidade igual a quantidade informada durante a seleção dos produtos
3. Adicionar produto já presente no carrinho	Possuir um produto no carrinho	1. Pesquiso um produto que já está presente no carrinho na barra de pesquisa 2. Clico no produto escolhido 3. Seleciono a quantidade desejada 4. Clico em "Adicionar ao Carrinho"	Validar que a quantidade do produto selecionado é incrementada	O produto selecionado deve ter sua quantidade incrementada no carrinho e não deve aparecer em duplicidade
4. Finalizar carrinho vazio	Carrinho vazio	1. Clico no carrinho 2. Caso tenha produtos, devo retirá-los 3. Clico em "Finalizar carrinho"	Validar que não é possível finalizar um carrinho vazio	Deve ser exibido uma mensagem amigável informando que não é possível finalizar um carrinho de compras vazio
5. Remover um produto do carrinho	Carrinho vazio	1. Pesquiso um produto na barra de pesquisa 2. Clico no produto escolhido 3. Seleciono a quantidade desejada 4. Clico em "Adicionar ao Carrinho" 5. Clico em "Carrinho" 6. Removo o item adicionado nos passos anteriores do carrinho	Validar que é possível remover um item do carrinho	1. Deve ser exibido uma mensagem amigável perguntando se o usuário deseja realmente remover o item do carrinho e, após a remoção, deve ser exibido outra mensagem confirmando a remoção do produto. 2. Após os passos informados, o carrinho deve ser exibido sem nenhum produto
6. Finalizar um carrinho com produtos	Carrinho vazio	1. Pesquiso um produto na barra de pesquisa 2. Clico no produto escolhido 3. Seleciono a quantidade desejada 4. Clico em "Adicionar ao Carrinho" 5. Clico em "Carrinho" 6. Clico em "Finalizar carrinho" 7. Informo todos os dados necessários na tela de checkout e finalizo a compra	Validar que é possível finalizar um carrinho válido	A compra deve ser concluída com sucesso e devo receber um e-mail de confirmação da compra

Figura 3: Casos de teste levantados pelo QA Senior

1. **Caso de Teste:** Adicionar um produto ao carrinho
 - **Descrição:** Verificar se é possível adicionar um produto ao carrinho.
 - **Pré-condições:** Usuário está logado na loja virtual e há pelo menos um produto disponível.
 - **Passos:**
 - 1.1. Navegar até a página do produto.
 - 1.2. Clicar no botão "Adicionar ao Carrinho".
 - **Resultados Esperados:**
 - O produto é adicionado ao carrinho.
 - O carrinho não está vazio.
 - Não há duplicações do mesmo produto no carrinho.
2. **Caso de Teste:** Remover um produto do carrinho
 - **Descrição:** Verificar se é possível remover um produto do carrinho.
 - **Pré-condições:** Usuário está logado na loja virtual e há pelo menos um produto no carrinho.
 - **Passos:**
 - 2.1. Acessar o carrinho de compras.
 - 2.2. Localizar o produto no carrinho.
 - 2.3. Clicar no botão "Remover".
 - **Resultados Esperados:**
 - O produto é removido com sucesso do carrinho.
 - O carrinho não está vazio após a remoção.
3. **Caso de Teste:** Finalizar uma compra com sucesso
 - **Descrição:** Verificar se é possível finalizar uma compra com sucesso.
 - **Pré-condições:** Usuário está logado na loja virtual, há pelo menos um produto no carrinho, e o carrinho não está vazio.
 - **Passos:**
 - 3.1. Acessar o carrinho de compras.
 - 3.2. Clicar no botão "Finalizar Compra".
 - **Resultados Esperados:**
 - A compra é finalizada com sucesso.
 - O carrinho fica vazio após a compra.
4. **Caso de Teste:** Tentar finalizar um carrinho vazio
 - **Descrição:** Verificar se não é possível finalizar um carrinho vazio.
 - **Pré-condições:** Usuário está logado na loja virtual e o carrinho está vazio.
 - **Passos:**
 - 4.1. Acessar o carrinho de compras.
 - 4.2. Tentar finalizar a compra.
 - **Resultados Esperados:**
 - O sistema exibe uma mensagem de erro informando que não é possível finalizar um carrinho vazio.
5. **Caso de Teste:** Adicionar o mesmo produto ao carrinho sem duplicação
 - **Descrição:** Verificar se não há duplicações de produtos ao adicionar o mesmo produto várias vezes ao carrinho.
 - **Pré-condições:** Usuário está logado na loja virtual e há pelo menos um produto disponível.
 - **Passos:**
 - 5.1. Navegar até a página do produto.
 - 5.2. Clicar no botão "Adicionar ao Carrinho" duas vezes seguidas.
 - **Resultados Esperados:**
 - O produto é adicionado ao carrinho apenas uma vez.
 - A quantidade do produto no carrinho é incrementada corretamente.

Figura 4: Casos de teste gerados pelo ChatGPT

✦ Casos de teste funcionais para a funcionalidade de carrinho de compras

Requisitos:

- O usuário deve poder adicionar produtos ao carrinho de compras.
- O usuário deve poder remover produtos do carrinho de compras.
- O usuário não deve poder finalizar um carrinho vazio.
- Ao adicionar um mesmo produto ao carrinho, não deve haver duplicações de produtos, será adicionado à sua quantidade

Casos de teste:

CT-01 - Adicionar produto ao carrinho

Objetivo: Testar a funcionalidade de adicionar um produto ao carrinho.

Pré-condições:

- O usuário deve estar logado no sistema.
- O produto deve estar disponível para compra.

Passos:

1. O usuário seleciona o produto que deseja adicionar ao carrinho.
2. O usuário clica no botão "Adicionar ao carrinho".
3. O produto deve ser adicionado ao carrinho.

Resultado esperado:

- O produto deve ser adicionado ao carrinho.

CT-02 - Remover produto do carrinho

Objetivo: Testar a funcionalidade de remover um produto do carrinho.

Pré-condições:

- O produto deve estar no carrinho.

Passos:

1. O usuário seleciona o produto que deseja remover do carrinho.
2. O usuário clica no botão "Remover do carrinho".
3. O produto deve ser removido do carrinho.

Resultado esperado:

- O produto deve ser removido do carrinho.

Figura 5: Casos de teste gerados pelo Bard

CT-03 - Finalizar compra com carrinho vazio

Objetivo: Testar a funcionalidade de finalizar uma compra com carrinho vazio.

Pré-condições:

- O carrinho deve estar vazio.

Passos:

1. O usuário clica no botão "Finalizar compra".

Resultado esperado:

- O sistema deve exibir uma mensagem de erro informando que o carrinho está vazio.

CT-04 - Adicionar mesmo produto duas vezes ao carrinho

Objetivo: Testar a funcionalidade de adicionar um mesmo produto duas vezes ao carrinho.

Pré-condições:

- O produto deve estar disponível para compra.

Passos:

1. O usuário adiciona o produto ao carrinho.
2. O usuário adiciona o produto novamente ao carrinho.

Resultado esperado:

- A quantidade do produto no carrinho deve ser atualizada para 2.

CT-05 - Adicionar produto indisponível ao carrinho

Objetivo: Testar a funcionalidade de adicionar um produto indisponível ao carrinho.

Pré-condições:

- O produto deve estar indisponível para compra.

Passos:

1. O usuário seleciona o produto que deseja adicionar ao carrinho.
2. O usuário clica no botão "Adicionar ao carrinho".

Resultado esperado:

- O sistema deve exibir uma mensagem de erro informando que o produto está indisponível.

Figura 6: Casos de teste gerados pelo Bard

Outros casos de teste:

- Testar a funcionalidade de adicionar produtos ao carrinho com diferentes quantidades.
- Testar a funcionalidade de remover produtos do carrinho com diferentes quantidades.
- Testar a funcionalidade de finalizar uma compra com carrinho com diferentes produtos e quantidades.

Estes são apenas alguns exemplos de casos de teste funcionais para a funcionalidade de carrinho de compras. Outros casos de teste podem ser necessários, dependendo dos requisitos específicos da loja virtual.

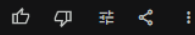


Figura 7: Casos de teste gerados pelo Bard