



Universidade Federal do Ceará
Centro de Ciências
Departamento de Computação
Verificação, Validação e Teste de Software (CK0241)

TB06 - Relatório de Falhas

Food Delivery Website - Tomato

Gabriel Moreira de Andrade - 536708

Álvaro Siqueira Galvão - 537649

Fábio Agostinho Filho - 538521

Histórico de versões

Versão	Data	Autor	Descrição
1.0	24/02/2025	Gabriel	Criação da versão inicial do relatório de falhas
1.1	24/02/2025	Gabriel	Criação da Introdução
1.2	24/02/2025	Gabriel	Adição de do casos de testes CT001 ao CT009 e CT001E
1.3	25/02/2025	Álvaro	Adição de do casos de testes CT010 e CT011 e CT012 ao CT014
1.4	28/02/2025	Fábio	Conclusão e glossário
1.5	28/02/2025	Gabriel, Fábio e Álvaro	Discussão
1.6	28/02/2025	Fábio	Adição dos casos de testes CT015 ao CT020

Sumário

1. Introdução	4
2. Resultados dos Testes	4
2.1 Escopo do teste	4
2.2 Resultados da execução dos testes por cenário	5
3. Discussão	6
4. Conclusão	6
5. Referências	7
6. Glossário	7
7. Apêndice	8

1. Introdução

Para garantir o cumprimento de todos os requisitos de uma aplicação, é essencial testar todos os casos de testes para cada cenário. Segundo Sommerville (2015), a verificação e validação são processos fundamentais para garantir a qualidade do software e reduzir falhas na produção. Também, de acordo com o IEEE 829-2008, a documentação detalhada dos casos de teste é essencial para garantir a rastreabilidade e reprodutibilidade dos defeitos identificados. Assim, esse documento tem o objetivo de apresentar o resultado e o relatório de falhas dos casos desenvolvidos nos TB4 [1].

A aplicação utilizada é a Tomato, uma aplicação web FullStack projetada para gerenciar pedidos de alimentos de forma simples e intuitiva, com foco em aprendizado e acessibilidade para desenvolvedores.[\[2\]](#)

2. Resultados dos Testes

2.1 Escopo do teste

As principais considerações prévias para um bom acompanhamento do relatório se dá pela mudança e correções de alguns casos e cenários do TB4.[1]

Inicialmente, foi trocado o ID de alguns testes de cenários. O RF005 possui o TS002, que foi trocado para TS004. Além disso, todos os ID's referentes aos requisitos posteriores partiram dessa contagem. Em relação aos testes modificados, foram adicionados e melhor explicados no apêndice.

Além disso, foram testados os CT's incluindo os dos RF's 001 ao 012, sendo constatados erros nos CT001E (caso de teste adicionado após o último documento), CT007, CT011, CT013 e no CT018, sendo todos esses erros detalhados e destacados na próxima tabela do tópico 2.2.

2.2 Resultados da execução dos testes por cenário

Requisito	ID do Cenário	Nome do cenário	Qtde de CTs executados	Qtde de CTs que falharam	IDs dos CTs que falharam
RF001	TS001	Validar login	3	0	N/a
RF002	TS002	Validar signup	4	1	CT001E
RF003	TS003	Adição de produtos ao carrinho	2	1	CT007
RF005	TS004	Redução de produtos do carrinho	1	0	N/a
RF007	TS005	Acessar o carrinho de compras como cliente	2	0	N/a
RF008	TS006	Acessar status do pedido como cliente	3	2	CT011; CT013
RF009	TS007	Remover produtos como admin	1	0	N/a
RF010	TS008	Acessar os pedidos como admin	2	0	N/a
RF011	TS009	Adicionar produtos como admin	2	1	CT018
RF012	TS010	Editar status de pedido como admin	2	0	N/a
Total			22	5	

3. Discussão

A análise dos resultados dos testes evidencia falhas funcionais relevantes no sistema Tomato, principalmente relacionadas à usabilidade e integridade dos processos de

compra e gerenciamento de pedidos. Dos 22 casos de testes executados, 5 apresentaram falhas, indicando uma taxa de erro próxima dos 25%.

Os principais problemas encontrados foram:

- **Erros na validação de credenciais e cadastro:** O sistema permitiu o cadastro de usuários já existentes (CT001E) e falhou em bloquear acessos indevidos.
- **Deficiências no fluxo de compra:** Falhas ao acessar status do pedido (CT011, CT013) e no processo de pagamento, que não fornecia feedback adequado (CT011).
- **Problemas na administração dos produtos:** A remoção de produtos teve um tempo de resposta elevado (CT014), levando a múltiplas notificações de erro.
- **Deficiências na interface e validação de arquivos:** A funcionalidade de upload de imagens permitiu a inclusão de arquivos inválidos, como PDFs (CT018).

Beizer (1990) destaca que falhas relacionadas à validação de entradas são uma das principais causas de vulnerabilidades em sistemas de software. No caso dos testes do Tomato, identificou-se que o sistema permitia o cadastro de usuários já existentes sem apresentar um alerta adequado (CT001E), o que pode comprometer a integridade do banco de dados.

Além disso, observou-se que alguns cenários não previstos no planejamento inicial deveriam ter sido considerados, como a validação da aplicação de códigos promocionais. Esses problemas destacam a importância da revisão contínua dos critérios de teste para garantir cobertura completa do sistema.

A correção dessas falhas deve ser acompanhada de novos testes para validar as melhorias implementadas. Como destaca Pressman (2016), um ciclo contínuo de verificação e validação contribui para um software mais robusto e confiável.

4. Conclusão

Este relatório apresentou os resultados dos testes funcionais realizados no sistema Tomato, destacando as falhas encontradas e suas possíveis implicações. A análise demonstrou que, apesar de a maioria das funcionalidades estarem operacionais, há pontos críticos que necessitam de correção para garantir a experiência do usuário e a integridade do sistema.

Os principais problemas identificados estão relacionados à validação de entradas, feedback do sistema e tempo de resposta, indicando a necessidade de aprimoramentos tanto na lógica de backend quanto na interface do usuário. Com base nessas informações, recomenda-se a correção das falhas identificadas e a execução de novos testes para validar as melhorias implementadas.

5. Referências

1. PRESSMAN, Roger S. **Engenharia de Software: uma abordagem profissional**. 8ª ed. McGraw Hill Brasil, 2016.
2. SOMMERVILLE, Ian. **Software Engineering**. 10ª ed. Pearson, 2015.
3. IEEE Standard for Software and System Test Documentation (IEEE 829-2008). Institute of Electrical and Electronics Engineers, 2008.
4. BEIZER, Boris. **Software Testing Techniques**. 2ª ed. Van Nostrand Reinhold, 1990.

6. Glossário




Termo	Definição
Caso de Teste(CT)	Conjunto de condições, entradas e ações definidas para verificar se um sistema se comporta conforme o esperado
Cenário de Teste(TS)	Conjunto de casos de teste que validam um requisito específico do sistema
Requisito Funcional(RF)	Requisito que descreve uma funcionalidade específica do sistema, ou seja, o que o software deve fazer
Requisito Não Funcional(RNF)	Requisito que define restrições e características do sistema, como desempenho, usabilidade e segurança
Partição de Equivalência	Técnica de teste que divide os dados de entrada em grupos (partições) para reduzir o número de testes, garantindo cobertura eficiente
Open Source	Modelo de desenvolvimento de software cujo código-fonte é disponibilizado publicamente, permitindo modificação e distribuição livre pelos usuários



7. Apêndice

Ao realizar-se o teste do CT010, referente ao TS005 e RF007, notou-se uma funcionalidade que poderia ter sido elencada para teste no TB4 mas acabou por ficar ausente. Trata-se do adição de código promocional ao pedido. No caso, o teste seria classificado como falho, já que o botão da funcionalidade sequer responde ao ser apertado, quando poderia emitir uma notificação indicando código inexistente, por exemplo. Posteriormente, analisando o código referente a esta página, percebeu-se que a funcionalidade não estava implementada.

Tomato.

[Home](#) [Menu](#) [Mobile-App](#) [Contact Us](#)



Items	Title	Price	Quantity	Total	Remove
	Pizza	\$20	11	\$220	x
	Calzone	\$13	1	\$13	x

Cart Total

Subtotal

\$233

Delivery Fee

\$2

Total

\$235

Proceed To Checkout

If you have a promo code, Enter it here

código aleatório




Apply

Figura 1: caso de teste não rastreado no requisito RF007: tela com código promocional fictício adicionado. Ao acionamento do botão “Apply” não se recebe resposta alguma da página.

Ao realizar-se o teste do CT011, referente ao TS006 e RF008, clicando no botão de realizar pagamento, a página apresentava um erro genérico no topo da página. De fato era sabido não haver nenhuma API de pagamento integrada ao sistema, por se tratar de uma aplicação *open source* com fins de estudo e de base para ser estendida em aplicações reais, mas optou-se por apresentar este caso como uma falha de teste pois seria algo necessário em uma aplicação real.

Tomato.

localhost:5173 diz
Error



Delivery Information

Álvaro

Siqueira

email@ficticio.com

endereço fantasia

endereço fantasia

terra do endereço fantasia

endereço fantasia

endereço fantasia

+55999999999

Cart Total

Subtotal

\$233

Delivery Fee

\$2

Total

\$2




Proceed To Payment

Figura 2: caso de teste CT011, TS006, RF008: tela com erro ao tentar fazer pagamento.


Ao realizar-se o teste do CT013, referente ao TS006 e RF008, na tela de pedidos cadastrados, quando de aciona o botão “Track Order”, em algum pedido pendente, para procurar por opção de cancelamento, a página não emite nenhuma resposta, tornando impossível que se cancele um pedido ainda não pago.

Tomato.

[Home](#) [Menu](#) [Mobile-App](#) [Contact Us](#)



My Orders




Pizza x 1l, Calzone x 1

\$235.00

Items: 2

Food Processing

Track Order




Pizza x 1l, Calzone x 1

\$235.00

Items: 2

Food Processing

Track Order




Pizza x 1l, Calzone x 1

\$235.00

Items: 2

Food Processing

Track Order




Pizza x 1l, Calzone x 1

\$235.00

Items: 2

Food Processing

Track Order




Pizza x 1l, Calzone x 1

\$235.00

Items: 2

Food Processing

Track Order




Pizza x 1l, Calzone x 1

\$235.00

Items: 2

Food Processing

Track Order




Pizza x 1l, Calzone x 1

\$235.00

Items: 2

Food Processing

Track Order



Pizza x 1l, Calzone x 1

\$235.00

Items: 2

Food Processing

Track Order

Figura 3: caso de teste CT013, TS006, RF008: tela com pedidos pendentes, não emitindo resposta ao se acionar “Track Order” à direita dos itens.

Ao realizar-se o teste do CT014, referente ao TS007 e RF009, o teste e requisito testados em si foram realizados de maneira bem sucedida, porém, surgiu um problema de natureza não funcional não rastreado sequer no TB2, quando elencamos requisitos não funcionais. Trata-se do tempo de resposta ao acionamento de botão, que seria perfeitamente plausível de ser realizado no presente trabalho. A situação em questão se deu quando, ao acionar-se o ícone “x” para remover um item da lista de produtos, a remoção demorou cerca de 6 a 7 segundos para ser efetivada, gerando uma percepção inicial de falha, que na ocasião levou a que o testador pressionasse seguidas vezes o tal botão, à espera de resposta. Porém, em seguida de fato o produto é removido e é gerada uma notificação de sucesso, como era esperado, mas essa notificação é seguida de diversas notificações de erro, uma para cada pressionamento após o comando efetivo de remoção, enquanto o produto ainda estava na tela.

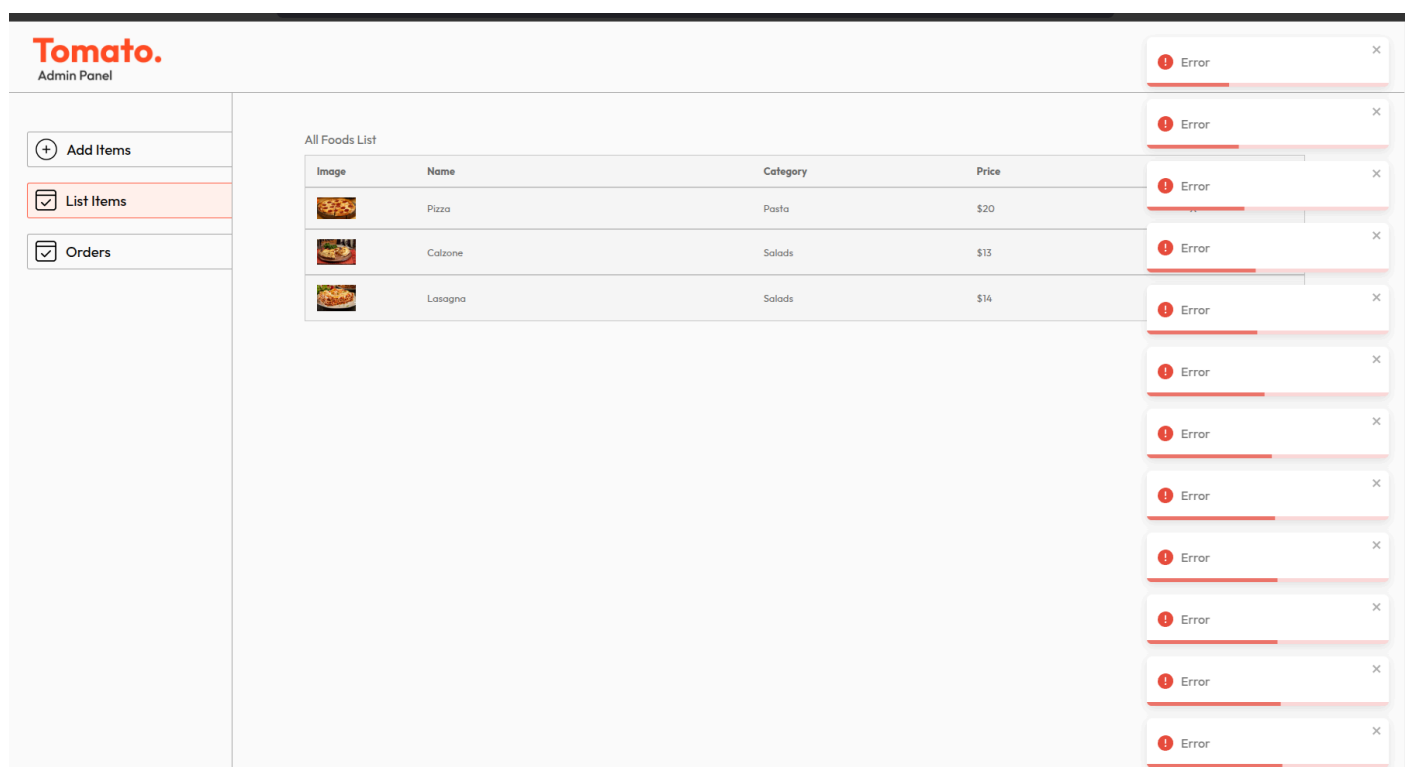


Figura 4: caso de teste não rastreado durante teste do CT014, TS007, RF009: tela com diversas notificações de erro após muitos acionamentos de botão devido a demora excessiva na remoção.

Como os CT001, CT002 e CT003 não foram bem descritos, foi considerado como “credenciais inválidas” quaisquer credenciais não presentes no banco de dados acessado pelo servidor. Além disso, com exceção do CT002, a caixa de termos foi sempre assinada.

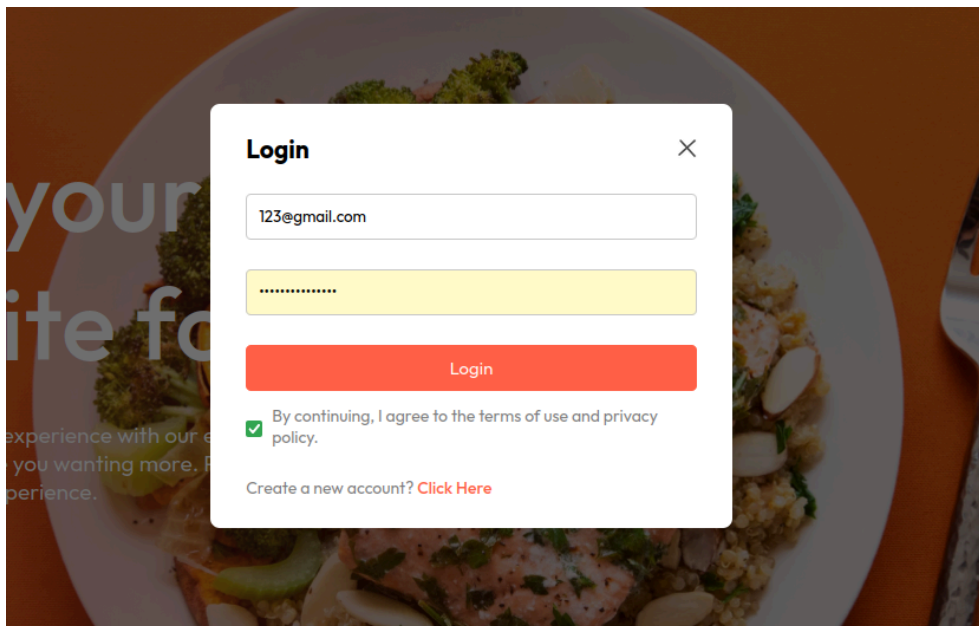


Figura 5: CT001, TS001, RF001: Login preenchido email inválido e com a caixa de termos assinada.

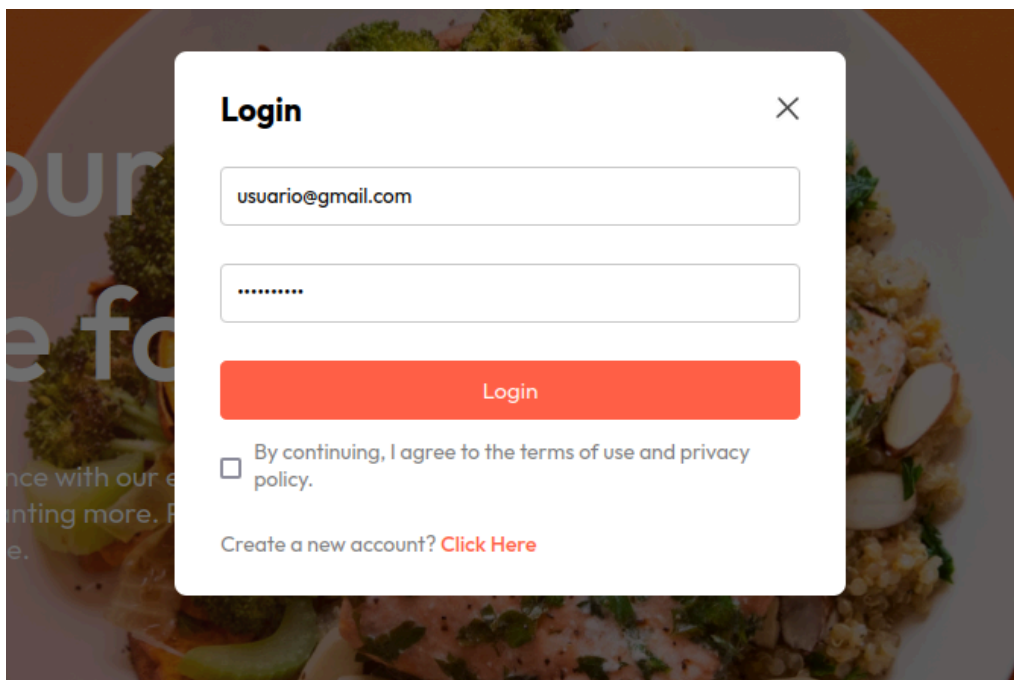


Figura 6: CT002, TS001, RF001: Login preenchido email inválido e com a caixa termos não assinada.

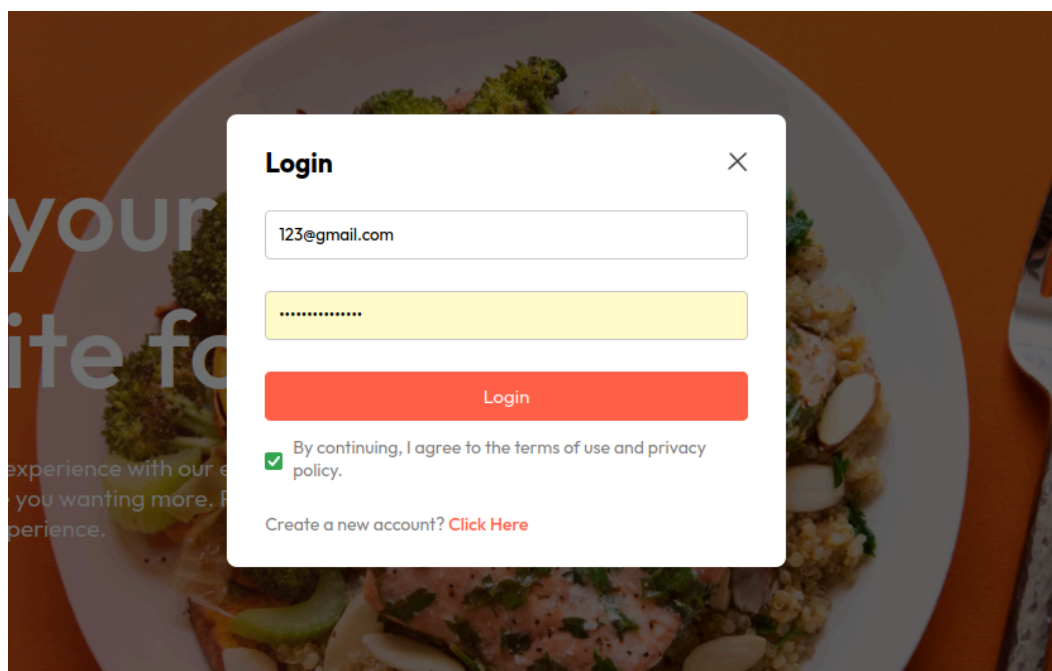


Figura 7: CT002, TS001, RF001: Login preenchido email válido e com a caixa termos assinada.

Em relação aos casos de testes CT005, CT006, CT007 e CT001E (referentes ao cenário de validação de signup). Foi adicionado o CT0012 que verifica se é possível adicionar um usuário que já existe (mesmo email). Além disso, para os outros CT's desse cenário, foi considerado "email inválido" emails sem o caractere @ (complementando as informações deixadas de lado). Segue o CT001E adicionado:

ID do RF/RNF	RF002
ID do Cenário de Teste	TS002
Nome do Cenário	Validar Signup
ID do Caso de Teste	CT001E
Nome do Caso de Teste	Cadastro com usuário já existente na base de dados
Precondição	O usuário não está autenticado
Dados de Entrada	Email, senha e usuário iguais ao já cadastrado
Passo a Passo	<ol style="list-style-type: none">1. Preencher no campo usuário: "usuario2. Preencher no campo email: "usuario@gmail.com"3. Preencher no campo senha: "usuario123"

Resultado Esperado	PopUp com “Usuário já cadastrado” e não permitir completar a ação
Técnica de Teste Funcional	Partição de Equivalência
Observações	

Ao realizar esse caso de teste, o usuário pode adentrar a área para os autenticados, ou seja, nenhuma mensagem de erro ou impedimento de acessar o ambiente. Segue a imagem referente à falha encontrada pelo CT001E:

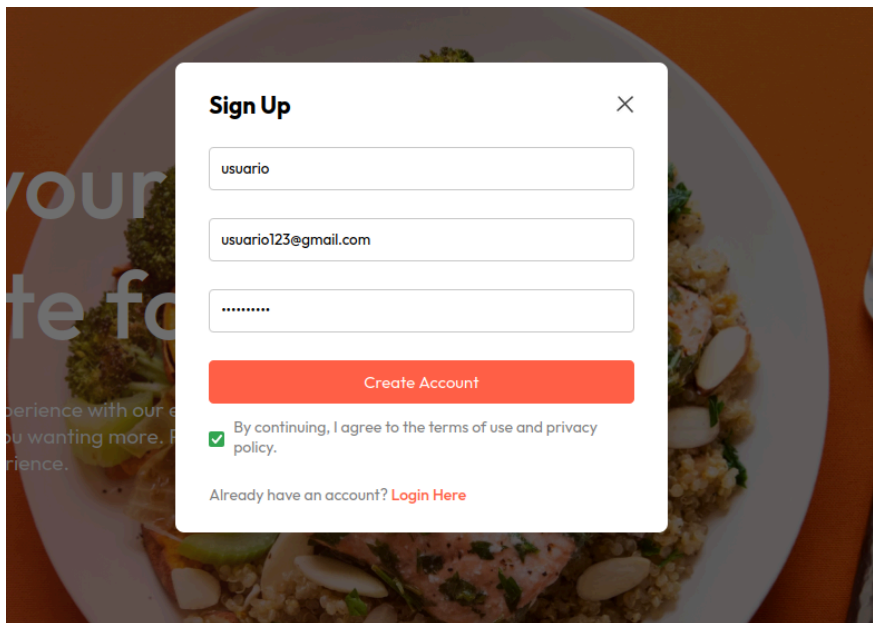
A screenshot of a 'Sign Up' modal form. The form has a title 'Sign Up' and a close button (X). It contains three input fields: 'usuario' (filled with 'usuario'), 'email' (filled with 'usuario123@gmail.com'), and 'password' (filled with '*****'). Below the fields is an orange 'Create Account' button. Under the button is a checkbox with a green checkmark and the text 'By continuing, I agree to the terms of use and privacy policy.' At the bottom, there is a link: 'Already have an account? [Login Here](#)'. The background of the page is a blurred image of a dish.

Figura 8: CT004, CT005, CT006, CT001E, TS001, RF002: Signup com todos os campos preenchidos

Em relação aos CT007 e CT008, seguem essas imagens para melhor entendimento e do erro gerado por ultrapassar 99 produtos a serem adicionados:

Top dishes near you



Figura 9: CT007, CT008, TS004, RF003: Imagem sem ter adicionado o produto ao carrinho.

Top dishes near you

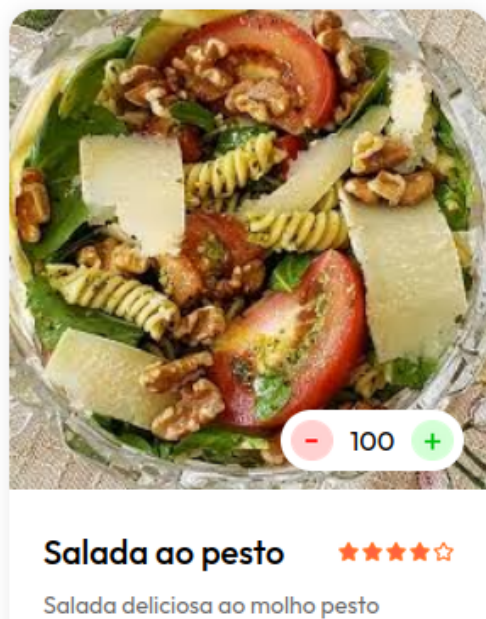


Figura 10: CT007, CT008, TS004, RF003 e RF005: Imagem já tendo adicionado e tendo ultrapassado 99.

Em relação ao CT017, segue em anexo as imagens do preenchimento do formulário e do erro. No caso, ao clicar no botão adicionar, nada ocorre, pois o botão não produz ação nenhuma:

The screenshot displays the 'Tomato. Admin Panel' interface. On the left, a sidebar contains three menu items: 'Add Items' (highlighted with a red border and a plus icon), 'List Items' (with a checkmark icon), and 'Orders' (with a checkmark icon). The main content area is titled 'Add Items' and includes the following fields:

- Upload Image:** A gray square button with a play icon.
- Product Name:** A text input field containing 'Salada César'.
- Product Description:** A large text area containing 'Salada César'.
- Product Category:** A dropdown menu with 'Salads' selected.
- Product Price:** A text input field containing '30'.
- Add:** A black button with white text.

Figura 11: CT017 e RF011: Adição de item preenchendo todos os campos.

Tomato.

Admin Panel

+ Add Items

✓ List Items

✓ Orders

Upload Image

Product Name

Salada César

Product Description

Write the description here

Product Category

Salads

30

Add

!

Preencha este campo.

Figura 12: CT018 e RF011: Adição de item não preenchendo todos os campos.

Ao realizar-se o teste do CT017 e do CT018, notou-se uma funcionalidade que poderia ter sido elencada para teste no TB4 mas acabou por ficar ausente. Trata-se do caso de o arquivo escolhido para a imagem não ser no formato de imagem. Ao tentar fazer o upload de um arquivo em pdf, foi possível perceber que o site não mandou mensagem de erro e deixou o arquivo como se fosse a imagem do produto:

Tomato.

Admin Panel

Add Items

List Items

Orders

Upload Image

Product Name

Type Here

Product Description

Write the description here

Product Category

Salads

Product Price

\$20

Add

Figura 13: Novo caso de teste, onde um arquivo pdf foi adicionado no campo de imagem.