



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

TB02 - Plano de V&V

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	08/12/2024	Gabriel	Criação do documento do Plano de V&V
1.1	08/12/2024	Gabriel, Fábio e Álvaro	Divisão de tarefas
2.0	09/12/2024	Álvaro	Inclusão das seções 1 e 2, bem como as respectivas adições de referências e glossário.
2.1	09/12/2024	Gabriel	Inclusão das seções 3, 4 e 5, bem como as respectivas adições de referências e glossário.
2.2	09/12/2024	Fábio	Inclusão das seções 7, 8 e 9 bem como as respectivas adições de referências e glossário.

Sumário

1. Identificador do Plano de Teste	4
2. Introdução	4
2.1. A aplicação	4
3. Escopo e Estratégia de Teste	4
3.1. Requisitos Funcionais	4
3.2. Requisitos Não Funcionais	6
4. Ambiente	6
4.1. Ferramentas	6
5. Recursos Humanos	6
5.1. Papéis e Responsabilidades	6
5.2. Treinamento	7
6. Produtos de V&V	7
7. Cronograma	7
8. Riscos	7
9. Conclusão	8
10. Referências	8
Glossário	8

1. Identificador do Plano de Teste

Título: Plano para o processo de verificação e validação do software Tomato.

Código de identificação: UFC-VV-001

Autores: Gabriel Moreira de Andrade, Álvaro Siqueira Galvão, Fábio Agostinho Filho

Local: Fortaleza-CE

Data: 09/12/2024

2. Introdução

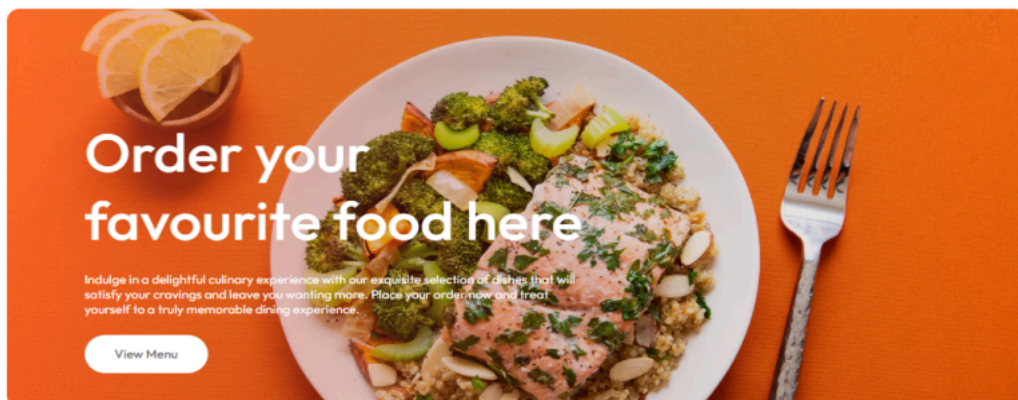
Para garantir o bom desenvolvimento do processo de verificação e validação (V&V) de um software, é de suma importância o planejamento das atividades a serem realizadas, de modo a definir dentro de um escopo o que é relevante a ser testado, quanto as atividades: os métodos, técnicas e ferramentas a serem empregados e os artefatos a serem produzidos, além dos critérios de aceitação (MYERS; SANDLER; BADGETT, 2012). Ademais, é boa ocasião para se prevenir em relação aos riscos a que o projeto está sujeito, podendo assim tomar de maneira mais eficiente os paliativos necessários, e delinear responsabilidades e prazos para a consecução de tais atividades entre a equipe. O presente documento se propõe a expor todas essas etapas e desse modo servir de guia para execução do projeto da disciplina de V&V.

2.1. A aplicação

O Tomato é uma aplicação web FullStack projetada para gerenciar pedidos de alimentos de forma simples e intuitiva, com foco em aprendizado e acessibilidade para desenvolvedores. A aplicação é de código aberto (open source) e foi construída com a stack MERN (MongoDB, Express.js, React.js e Node.js). Seu código fonte pode ser encontrado no github cujo link se encontra nas referências [\[1\]](#). O principal objetivo do Tomato é oferecer um ambiente onde clientes e administradores possam interagir com o sistema de pedidos de maneira simples: Lado do Cliente: Os usuários podem navegar por uma lista de pratos, visualizar descrições e detalhes, e adicionar itens ao carrinho para fazer pedidos.

2.1.1 - Funcionamento

Lado do Cliente: Os usuários podem navegar por uma lista de pratos e visualizar descrições e detalhes, como mostra a figura 2, e adicionar itens ao carrinho para fazer pedidos (figura 3). A figura 1 mostra a tela principal no lado cliente da aplicação.



Explore Our Menu

Choose from a diverse menu featuring a tantalizing array of flavors and dishes crafted to perfection, promising a culinary journey like no other. Dive in and discover your new favorites today!

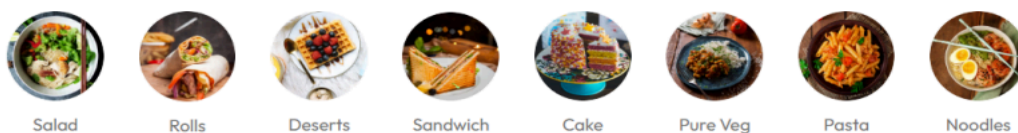


Figura 1. Tela principal da aplicação - lado cliente

Top dishes near you

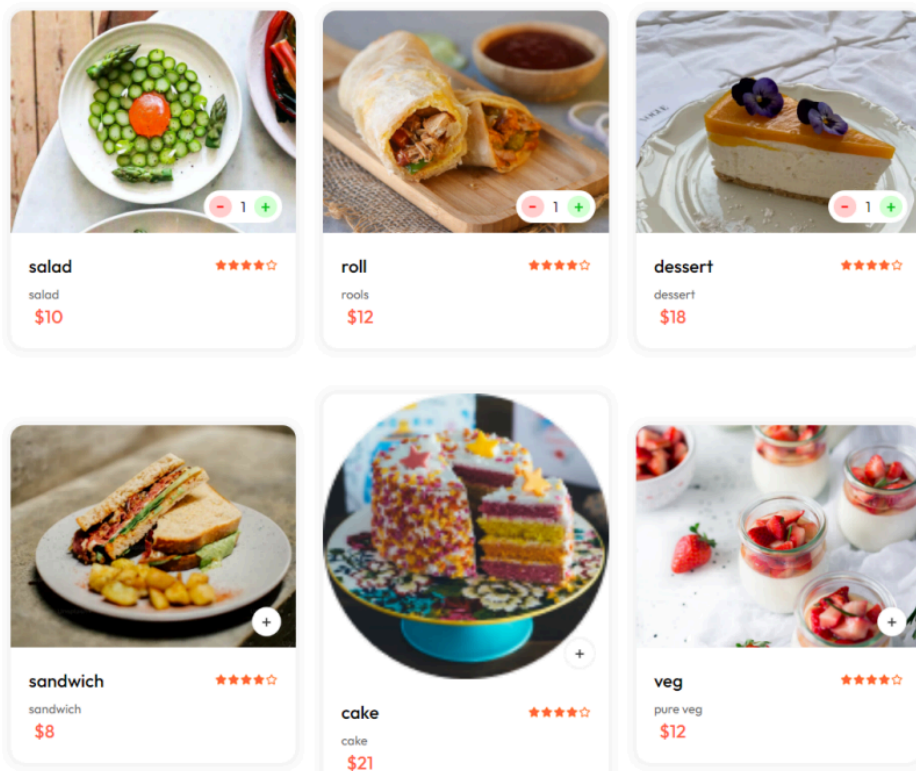





Figura 2. Tela para visualização dos pratos, com opção de adicioná-los ao pedido

Items	Title	Price	Quantity	Total	Remove
	salad	\$10	1	\$10	x
	roll	\$12	1	\$12	x
	dessert	\$18	1	\$18	x

Cart Total

Subtotal	\$40
Delivery Fee	\$2
Total	\$42

Proceed To Checkout

If you have a promo code, Enter it here


promo code

Apply

Figura 3. Tela para finalizar o pedido

Lado do Administrador: Os administradores têm acesso a uma interface que permite o controle sobre os pedidos recebidos e os pratos disponíveis. É possível adicionar pratos, como consta na figura 4, ou ainda editar ou remover (figura 5) e atualizar o status dos pedidos (figura 6).

Tomato.
Admin Panel




+ Add Items

☒ List Items

☒ Orders

Upload Image


Upload

Product Name

Type Here

Product Description

Write the description here

Product Category

Salads

Product Price

\$20

Add

Figura 4. Tela para criar novos pratos.

All Foods List









Image	Name	Category	Price	Action
	salad	Salad	\$10	X
	roll	Rolls	\$12	X
	dessert	Deserts	\$18	X
	sandwich	Sandwich	\$8	X
	cake	Cakes	\$21	X
	veg	Pure Veg	\$12	X
	pasta	Pasta	\$13	X
	noodles	Noodles	\$9	X

Figura 5. Tela para ver pratos disponíveis, com a opção de remoção à direita





<div> <div>+</div> <div>Add Items</div> </div> <div> <div>✓</div> <div>List Items</div> </div> <div> <div>✓</div> <div>Orders</div> </div>	<div>Order Page</div> <div> <div> <div></div> <div> <div>food1 x 1</div> <div>Items: 1</div> <div>\$23</div> <div>Delivered ▼</div> </div> </div> <div> <div>Athrva NAIK</div> <div>Navale,Nirmal,Nallasopara West,Palghar</div> <div>401304,</div> <div>mumbai, Maharashtra, 401304, India</div> <div>07666967661</div> </div> </div> <div> <div> <div></div> <div> <div>food1 x 1, food2 x 1</div> <div>Items: 2</div> <div>\$36</div> <div>Delivered ▼</div> </div> </div> <div> <div>Athrva NAIK</div> <div>Navale,Nirmal,Nallasopara West,Palghar</div> <div>401304,</div> <div>mumbai, Maharashtra, 401304, India</div> <div>07666967661</div> </div> </div> <div> <div> <div></div> <div> <div>roll x 1, veg x 1, pasta x 1</div> <div>Items: 3</div> <div>\$39</div> <div>Food Processi ▼</div> </div> </div> <div> <div>Athrva Naik</div> <div>Navale,Nirmal,Nallasopara West,Palghar</div> <div>401304,</div> <div>mumbai, Maharashtra, 401304, India</div> <div>07666967661</div> </div> </div> <div> <div> <div></div> <div> <div>salad x 1, dessert x 1</div> <div>Items: 2</div> <div>\$30</div> <div>Delivered ▼</div> </div> </div> <div> <div></div> <div></div> <div></div> <div></div> </div> </div>
--	---

Figura 6. Tela para controlar o status dos pedidos

3. Escopo e Estratégia de Teste

Como se trata de uma aplicação de gerenciamento de pedidos, com perfis de usuário e administrador, o foco será validar o login dos usuários comuns, já que o usuário administrador não exige autenticação por funcionar na máquina do administrador. Além disso, para os usuários comuns, o foco será em conseguir adicionar corretamente os pedidos no carrinho e conseguir passar as informações do pedido para o administrador que será testado se consegue adicionar corretamente produtos para os outros usuários e se consegue visualizar as informações dos pedidos.

O sistema será submetido a **testes unitários** para validar cada interação dos usuários com o *backend*, garantindo o correto funcionamento entre cliente e servidor. Além disso, será realizada uma **análise estática de código** para assegurar a adoção de boas práticas de programação. **Testes de interface** serão aplicados para verificar a navegabilidade da aplicação, complementados por **testes de carga** que ajudarão a identificar os limites de desempenho do sistema.

3.1. Requisitos Funcionais

Código	Descrição do requisito	Tipo de estratégia de teste
RF001	Autenticação como cliente	Teste de interface e unitário
RF002	Sign-up como cliente	Teste de interface e unitário
RF003	Aumentar os produtos do carrinho como cliente	Teste de interface
RF005	Diminuir os produtos do carrinho como cliente	Teste de interface
RF007	Acessar ao carrinho de compras como cliente	Teste de interface
RF008	Acessar ao status do pedido como cliente	Teste de interface
RF09	Remover produtos como admin	Teste de interface e unitário
RF010	Acessar aos pedidos feitos como admin	Teste de interface
RF011	Adicionar produtos como admin	Teste de interface e unitário
RF012	Editar o status de cada pedido como admin	Teste de interface e unitário

Dos requisitos mencionados no TB anterior, não serão testados os seguintes:

Requisito	Descrição	Motivo
RF004	Ver histórico de compras como cliente	Pouco relevante e está incluído no teste de interface de outros requisitos
RF006	Filtrar os produtos por categoria como cliente	Pouco relevante e está incluído no teste de interface de outros requisitos

3.2. Requisitos Não Funcionais

Requisito não funcional	Descrição	Meta	Tipo de teste
RNF001	Suportar até 100 usuários simultâneos	Suportar 80 usuários simultâneos de maneira funcional	Teste de carga

4. Ambiente

Para o hardware, será necessário um processador Intel Core i3 ou equivalente, além de 128GB de armazenamento e 8GB de RAM. Para o *software* será necessário a instalação do Node.js, que permite executar o JavaScript fora do navegador, no terminal, NPM (Node Package Manager) para instalar as dependências e do MongoDB para o banco de dados.

4.1. Ferramentas

Como ferramentas principais, será necessário um editor de código para que alguns scripts sejam feitos. **ESLint** para fazer a análise estática de código, **JEST** para testes unitários, **Cypress** para testes de interface e **Postman/Artillery** para testes de carga por meio de requisições.

5. Recursos Humanos

5.1. Papéis e Responsabilidades

Papel	Responsável	Responsabilidade	Horas/semana
Elaborar testes, testar e documentar	Gabriel, Fábio e Álvaro	Realizar a análise estática de código.	2
Elaborar testes, testar e documentar	Gabriel	Projetar os testes, executar e documentar os requisitos RF001 , RF002	3
Elaborar testes, testar e documentar	Fábio	Projetar os testes, executar e documentar os requisitos RF003 , RF005 , RF007 , RF008	3
Elaborar testes de Interface, testar e documentar	Álvaro	Projetar os testes de interface, executar e documentar os requisitos RF009 , RF010 , RF011 , RF012	2
Elaborar testes unitários, testar e documentar	Fábio	Projetar os testes unitários, executar e documentar os requisitos RF009 , RF011 , RF012	2
Elaborar testes de carga, testar e documentar	Gabriel	Projetar os testes de carga, executar e documentar os requisitos RNF001	1

5.2. Treinamento

Tipo de Treinamento/Conhecimento	Metas	Integrantes
Node.js	Compreender o funcionamento de servidores com Node.js	Fábio e Álvaro
ESLint	Compreender o uso do ESLint para análise estática de código para JavaScript	Gabriel, Fábio e Álvaro
JEST	Compreender o uso do JEST para realizar testes unitários em JavaScript	Gabriel, Fábio e Álvaro
Cypress	Compreender o uso do Cypress para realizar testes de interface em JavaScript	Gabriel, Fábio e Álvaro
Postman	Compreender o uso do Postman para fazer testes de carga por meio de requisições ao servidor	Fábio e Álvaro
Artillery	Compreender o uso do Artillery para realizar testes de carga em JavaScript	Gabriel

6. Produtos de V&V

A tabela abaixo indica quais artefatos deverão ser entregues durante o projeto de V&V.

Código	Artefato
A001	Plano de Testes
A002	Relatório de Análise Estática
A003	Relatório de Falhas
A004	Testes Unitários
A005	Relatório de Testes de Carga
A006	Relatório de Testes de Interface
A007	Relatório de Inspeção

7. Cronograma

Atividades	Artefato/ Milestone	Data de início	Data de fim	Responsável
Realizar o planejamento dos testes a serem realizados	A001	16/12/2024	20/12/2024	Álvaro Gabriel Fábio
Realizar a inspeção do software para especificação dos requisitos e da arquitetura	A007	06/01/2025	07/01/2025	Álvaro Fábio Gabriel
Realizar análise estática	A002	08/01/2025	17/01/2025	Álvaro Fábio Gabriel
Realizar os testes unitários	A004	20/01/2025	31/01/2025	Álvaro Fábio Gabriel
Elaborar os relatórios dos testes de interface	A006	03/02/2025	07/02/2025	Álvaro Fábio Gabriel
Elaborar o relatório de falhas	A003	10/02/2025	21/02/2025	Álvaro Fábio Gabriel

Elaborar o relatório dos testes de carga	A005	10/02/2025	21/02/2025	Gabriel
--	------	------------	------------	---------

8. Riscos

Categoria	Risco	Impacto*	Probabilidade**	Estratégia de Mitigação
Recursos Humanos	Redução de Equipe	Alto	Baixa	Reformular as atribuições de cada atividade, remanejando de acordo com o tempo disponível dos membros remanescentes
Ambiente	Falta de um ambiente de testes adequado	Alto	Baixa	Utilizar o ambiente que ajude mais nos testes a serem feitos
Teste	Geração de Novo Bug durante os testes	Alto	Médio	Realizar retestes ao final do processo e desfazer o que foi adicionado, deixando o código igual ao original
Recursos Humanos	Desgaste físico e mental dos membros em decorrência de sobrecarga de atividades	Médio	Média	Remanejar as atividades de forma que os membros desgastados possam ter uma carga mais leve e não sobrecarregue os outros membros
Teste	Não possibilidade de execução de alguns testes devido ao fato do software ser muito antigo	Médio	Baixa	Utilizar ferramentas também mais antigas de teste, de forma a realizar os testes conforme a edição e atualização do software

9. Conclusão

Tendo em vista os requisitos funcionais e não-funcionais listados do protótipo de E-Commerce, têm-se um foco em verificar e validar as funcionalidades de cadastros, tanto como administrador e cliente, e as características de interface e de carga, podendo chegar aos testes de interface, de carga e os unitários, além da análise estática.

A partir dessas escolhas, foi possível traçar os produtos de verificação e validação e o cronograma das atividades desse projeto, além de dividir as atividades entre os membros e encontrar alguns possíveis riscos relacionados aos membros, ao ambiente que será realizado os testes por cada membro e aos testes que serão feitos, resultando nesse plano de V&V para o projeto em questão.

10. Referências

1. Athrva Naik. Github, 2024. Tomato-FoodOrdering. Disponível em:
<<https://github.com/AthrvaNaik/Tomato-FoodOrdering?tab=readme-ov-file>>.
Acesso em: 10, Dezembro de 2024.
2. MYERS, G. J.; SANDLER, C.; BADGETT, T. The art of software testing. Hoboken, N.J.: John Wiley & Sons, 2012.

Glossário

Termo	Definição
Cliente e Servidor	Arquitetura que divide os dispositivos entre quem faz requisições (cliente) e quem fornece as respostas (servidor)
E-Commerce	Aplicação Web voltada para compra e venda de produtos ou serviços pela internet
Full Stack	Tipo de aplicação que utiliza Front-End e Back-End integrados