



API SERVEREST PLANO DE TESTE

Setembro, 2022

Feito por:

Karoline Nascimento D. da Silva
karoline.silva.pb@compasso.com.br



ÍNDICE

• Introdução	-----	2
• Equipe de testes	-----	2
• Objetivos	-----	3
• Cronograma	-----	3
• Escopo	-----	4
• Mapa mental	-----	5
• Suíte de casos de teste	-----	6
• Estratégias de teste	-----	9
• Ferramentas	-----	10
• Testes candidatos à automação	-----	11
• Relatório de bugs	-----	13
• Observações complementares	-----	19
• Conclusão	-----	21



INTRODUÇÃO

SOBRE O PROJETO

Este documento possui como objetivo descrever o planejamento para a criação e implementação das validações de cobertura back-end (testes funcionais e automação) para a API ServeRest.

Todos as partes interessadas devem ter acesso à este material, como os clientes, equipe de desenvolvimento, líderes, entre outros.

EQUIPE DE TESTES

Para a realização deste projeto foi necessária a colaboração dos seguintes funcionários:

- Líder de QA: Carolina de Jesus;
- Analista de testes: Joaquim Machado;
- Testador 1: Abdias Nascimento;
- Testador 2: Dandara Palmares;
- Testador 3: Tereza de Benguela;
- Dev. Automação back-end: Antonieta de Barros;
- Arquiteto de automação: Luíz Gama.





OBJETIVOS

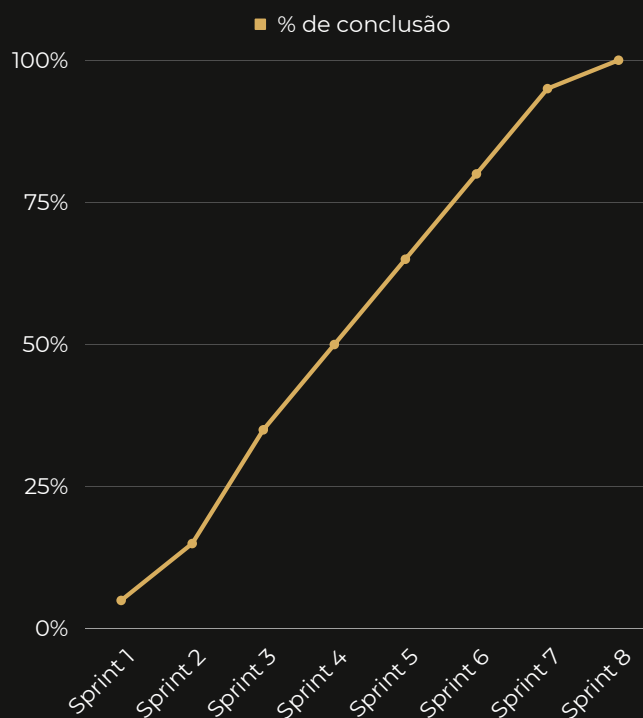
Para testar a aplicação da API ServeRest (e-commerce) foram definidos os seguintes objetivos:

- Implementar e observar o desempenho dos testes.
- Verificar se os fluxos prioritários da aplicação estão funcionando corretamente.
- Assegurar que as estratégias e critérios de aceite estão sendo seguidos.
- Identificar e reportar bugs encontrados.
- Identificar e reportar melhorias.
- Identificar e fornecer uma estimativa dos esforços de teste.



CRONOGRAMA

Para a realização deste projeto foi utilizado a metodologia ágil Scrum, no qual ao final de cada sprint (período de duas semanas) é realizado uma pequena entrega do projeto para o cliente.



Dessa forma é gerado benefício mútuo, pois o cliente conseguirá acompanhar o desenvolvimento do projeto em todo o período, e nós realizaremos entregas com maior valor agregado.



ESCOPO

Este documento apresenta testes de integração e testes do sistema que serão realizados em um ambiente de homologação.

Para tanto, presume-se que o teste de unidade já forneceu uma cobertura extensiva do código fonte e o teste de todas as interfaces do módulo.

ITENS PRESENTES NO ESCOPO DOS TESTES

Área administrativa: cadastro, edição e exclusão de produto.

Cadastro: rota cadastro de usuário ao acessar a url <'https://serverest.dev/usuarios'>.

Login: realizar login acessando a url <'https://serverest.dev/login'>.

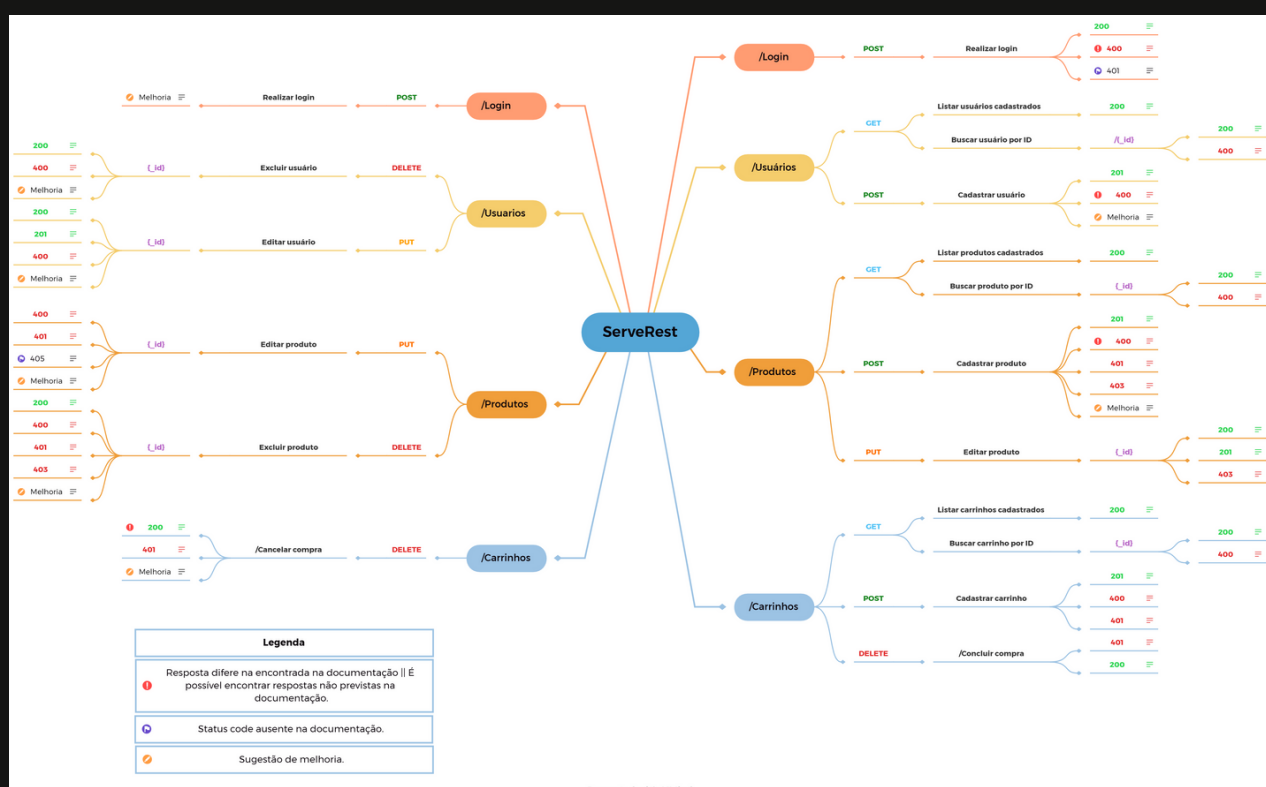
ITENS PRESENTES NO ESCOPO DOS TESTES

Itens fora do fluxo de compra: excluir ou editar usuário, excluir ou editar produto, cancelar compra.

Fluxo de compra: cadastro de usuário, login, listar produtos cadastrados, buscar produto por _id, cadastrar carrinho, concluir compra.

MAPA MENTAL

O mapa mental apresenta todas as rotas disponíveis na aplicação. Do lado tem-se as rotas prioritárias, ou seja, todos os fluxos que possuem altíssimo nível no impacto do fluxo de compras. Do lado esquerdo tem-se os fluxos menos prioritários.



Escaneie o QR Code ao lado para ver melhor a imagem!





SUÍTE DE CASOS DE TESTE

Lista de casos de teste para homologar as rotas de API do back-end.

Código	Objetivo
T01	Validar login com sucesso
T02	Validar login gerando erro 400
T03	Validar tentativa de login sem email
T04	Validar tentativa de login sem senha
T05	Validar tentativa de login com email inválido
T06	Validar tentativa de login com erro interno no servidor
T07	Validar com sucesso a exibição de lista de usuários cadastrados
T08	Validar a busca por usuário por _id com sucesso
T09	Validar a busca por usuário por _id sem sucesso
T10	Validar tentativa de cadastro com email já utilizado
T11	Validar tentativa de cadastro com email inválido
T12	Validar tentativa de cadastro com senha de 1 caractere
T13	Validar tentativa de cadastro sem nome
T14	Validar tentativa de cadastro com administrador incorreto
T15	Validar cadastro de usuário com sucesso
T16	Validar cadastro de usuário administrador com sucesso
T17	Validar exclusão de usuário por _id com sucesso



SUÍTE DE CASOS DE TESTE

Lista de casos de teste para homologar as rotas de API do back-end.

Código	Objetivo
T18	Validar exclusão de usuário por _id sem sucesso
T19	Validar edição de usuário por _id com sucesso
T20	Validar cadastro de usuário por _id com sucesso
T21	Validar tentativa de edição de usuário utilizando um email já cadastrado
T22	Validar com sucesso a exibição de lista de produtos cadastrados
T23	Validar busca de produto por _id com sucesso
T24	Validar busca de produto por _id sem sucesso
T25	Validar tentativa de cadastro de produto com token ausente, inválido ou expirado
T26	Validar tentativa de cadastro de produto já existente
T27	Validar tentativa de cadastro de produto sem ser usuário administrador
T28	Validar cadastro de produto com sucesso
T29	Validar tentativa de cadastro de produto sem nome
T30	Validar edição de produto com sucesso
T31	Validar tentativa de edição de produto sem ser usuário administrador
T32	Validar tentativa de editar o nome de um produto para um nome já existente



SUÍTE DE CASOS DE TESTE

Lista de casos de teste para homologar as rotas de API do back-end.

Código	Objetivo
T33	Validar tentativa de edição com token ausente, inválido ou expirado
T34	Validar exclusão de produto com sucesso
T35	Validar tentativa de cadastro de produto com quantidade menor que 0
T36	Validar tentativa de cadastro de produto com preço negativo
T37	Validar exclusão de produto que faz parte de carrinho
T38	Validar exclusão de produto com token ausente, inválido ou expirado
T39	Validar exclusão de produto sem ser usuário administrador
T40	Validar com sucesso a exibição de lista carrinhos cadastrados
T41	Validar busca de carrinho por _id com sucesso
T42	Validar busca de carrinho por _id sem sucesso
T43	Validar tentativa de cadastro de carrinho com token ausente, inválido ou expirado
T44	Validar cadastro de carrinho com sucesso
T45	Validar tentativa de cadastro de carrinho com produto inexistente
T46	Deve tentar concluir compra com token ausente, inválido ou expirado
T47	Deve concluir compra com sucesso
T48	Deve tentar cancelar compra com token ausente, inválido ou expirado
T49	Deve cancelar compra com sucesso



ESTRATÉGIA DE TESTE

Para este projeto foi utilizado a estratégia de teste analítica de risco. Dessa forma, os testes foram projetados e priorizados com base no nível de risco.

Eis os elementos que compõem a estratégia:

Nível de teste: Teste de sistema.

Objetivo: Validar se o sistema está completo e funcionará como esperado.

Tipo de teste: Funcional, usabilidade, teste dinâmico.

Definição de pronto: Uma intragável de história será considerada pronta quando: for validada em um ambiente de teste; ter todos os bugs de prioridade média/alta/bloqueante vinculados devidamente ajustados; ter sido aceita pelo cliente.

Métricas utilizadas: execução de casos de teste; informações sobre defeitos; conclusão das tarefas; qualidade e eficácia do software.

TESTE END-TO-END

A equipe de testes deverá realizar os testes na aplicação, conforme os cenários de teste previamente elaborados. Dessa maneira será validado o funcionamento dos testes de ponta-a-ponta, sempre levando em consideração o fluxo de compra e também os demais sistemas que suportam a operação.





FERRAMENTAS

Para realizar a realização do projeto foram utilizadas as seguintes ferramentas:

GERENCIAMENTO

Jira Software versão 9.1.

DOCUMENTAÇÃO

Pacote Office.

COMUNICAÇÃO

Microsoft Teams, Outlook.

MAPA MENTAL

XMind.

CAPTURE DE IMAGENS

Captura do Windows.

GRAVAÇÃO DE VÍDEOS

HYFY Screen.

FRAMEWORKS

Mocha versão 10.0.0, Cypress versão 9.7.0.

EDIÇÃO DE CÓDIGOS

VSCode versão 1.71.

LINGUAGEM DE PROGRAMAÇÃO

JavaScript.

VERSIONAMENTO DO CÓDIGO

GitHub.



TESTES CANDIDATOS À AUTOMAÇÃO

Lista de casos de testes aptos para serem automatizados. Esses testes foram escolhidos pois possuem grande impacto no fluxo de compras da API ServeRest.

Código	Objetivo
T01	Validar login com sucesso
T02	Validar login gerando erro 400
T03	Validar tentativa de login sem email
T04	Validar tentativa de login sem senha
T05	Validar tentativa de login com email inválido
T07	Validar com sucesso a exibição de lista de usuários cadastrados
T08	Validar a busca por usuário por _id com sucesso
T09	Validar a busca por usuário por _id sem sucesso
T10	Validar tentativa de cadastro com email já utilizado
T11	Validar tentativa de cadastro com email inválido
T12	Validar tentativa de cadastro com senha de 1 caractere
T13	Validar tentativa de cadastro sem nome
T14	Validar tentativa de cadastro com administrador incorreto
T15	Validar cadastro de usuário com sucesso
T16	Validar cadastro de usuário administrador com sucesso
T22	Validar com sucesso a exibição de lista de produtos cadastrados
T23	Validar busca de produto por _id com sucesso



TESTES CANDIDATOS À AUTOMAÇÃO

Código	Objetivo
T24	Validar busca de produto por _id sem sucesso
T25	Validar tentativa de cadastro de produto com token ausente, inválido ou expirado
T26	Validar tentativa de cadastro de produto já existente
T27	Validar tentativa de cadastro de produto sem ser usuário administrador
T28	Validar cadastro de produto com sucesso
T29	Validar tentativa de cadastro de produto sem nome
T30	Validar edição de produto com sucesso
T31	Validar tentativa de edição de produto sem ser usuário administrador
T35	Validar tentativa de cadastro de produto com quantidade menor que 0
T36	Validar tentativa de cadastro de produto com preço negativo
T40	Validar com sucesso a exibição de lista carrinhos cadastrados
T41	Validar busca de carrinho por _id com sucesso
T42	Validar busca de carrinho por _id sem sucesso
T43	Validar tentativa de cadastro de carrinho com token ausente, inválido ou expirado
T44	Validar cadastro de carrinho com sucesso
T45	Validar tentativa de cadastro de carrinho com produto inexistente
T46	Deve tentar concluir compra com token ausente, inválido ou expirado
T47	Deve concluir compra com sucesso
T48	Deve tentar cancelar compra com token ausente, inválido ou expirado
T49	Deve cancelar compra com sucesso



RELATÓRIO DE BUGS

Segue abaixo o relatório dos problemas identificados pela automação.

ROTA LOGIN

Descrição: Ao tentar realizar login com os campos email e/ou senha incorretos, o status code retornado pela API difere do esperado na documentação. Ademais, é possível encontrar respostas e status code não previstos na documentação.

Gravidade: Baixa.

Passos para reproduzir: Após acessar a URL da ServeRest, clique na rota Login, aperte no botão "Try it out" e altere o campo "senha". A resposta será o status 401 com a mensagem "Email e/ou senha inválidos" (esperava-se status 400, conforme a documentação).

Feito isso, desfaça a alteração no campo "senha". Agora, mude o email retirando o @.com. Será gerado o status 400 com a mensagem "email deve ser um email válido", resposta não prevista na documentação.

ROTA USUÁRIOS

Descrição: É possível encontrar respostas não previstas na documentação ao tentar realizar cadastro com os campos email e administrador incorretos.

Gravidade: Média.

Passos para reproduzir: Após acessar a URL da ServeRest, clique na rota POST Cadastro, aperte no botão "Try it out", retire o @ do campo "email" e deixe o campo "administrador" em branco.

Dessa forma será exibido mensagens não cobertas na documentação (status code 400):

```
{  
  "email": "email deve ser um email  
válido",  
  "administrador": "administrador deve  
ser 'true' ou 'false'"  
}
```



RELATÓRIO DE BUGS

Segue abaixo o relatório dos problemas identificados pela automação.

ROTA PRODUTOS

Descrição: É possível encontrar respostas não previstas na documentação ao tentar cadastrar um produto com o campo "preço" ou "quantidade" com número negativo.

Gravidade: Média.

Passos para reproduzir: Após acessar a URL da ServeRest e realizar login com sucesso, clique na rota POST Produtos, clique em "Try it out" e insira nos campos "preco" e "quantidade" números negativos.

Dessa forma será exibido mensagens não cobertas na documentação (status code 400):

```
{
  "preco": "preco deve ser um número positivo",
  "quantidade": "quantidade deve ser maior ou igual a 0"
}
```

ROTA CARRINHOS

Descrição: É possível encontrar respostas não previstas na documentação ao cancelar uma compra com sucesso.

Gravidade: Baixa.

Passos para reproduzir: Após acessar a URL da ServeRest, realizar login e cadastrar carrinho com sucesso, acesse a rota DELETE Carrinhos/cancelar-compra.

Clique em "Try it out" e execute. Será exibido a mensagem não cobertas na documentação (status code 200):

```
{
  "message": "Registro excluído com sucesso. Estoque dos produtos reabastecido"
}
```



RELATÓRIO DE BUGS

EVIDÊNCIAS - ROTA LOGIN

Requisição 01

```
{  
  "email": "beltrano@qa.com.br",  
  "password": " "  
}
```

Resposta

```
401  
Undocumented Error:  
  
Response body  
{  
  "message": "Email e/ou senha inválidos"  
}
```

Requisição 02

```
{  
  "email": "beltrano@qa",  
  "password": "teste"  
}
```

Resposta

```
400  
  
Error:  
  
Response body  
{  
  "email": "email deve ser um email válido"  
}
```




RELATÓRIO DE BUGS

EVIDÊNCIAS - ROTA USUÁRIOS

Requisição

```
{  
  "nome": "Fulano da Silva",  
  "email": "beltranoqa",  
  "password": "teste",  
  "administrador": " "  
}
```

Resposta

400

Error:

Response body

```
{  
  "email": "email deve ser um email válido",  
  "administrador": "administrador deve ser 'true' ou 'false'"  
}
```



RELATÓRIO DE BUGS

EVIDÊNCIAS - ROTA PRODUTOS

Requisição

```
{  
  "nome": "Mouse",  
  "preco": 0,  
  "descricao": "Mouse",  
  "quantidade": -1  
}
```

Resposta

400

Error:

Response body

```
{  
  "preco": "preco deve ser um número positivo",  
  "quantidade": "quantidade deve ser maior ou igual a 0"  
}
```



RELATÓRIO DE BUGS

EVIDÊNCIAS - ROTA CARRINHOS*

Resposta

200

Response body

```
{  
  "message": "Registro excluído com sucesso. Estoque dos produtos reabastecido"  
}
```

*Não há parâmetros para requisição.



OBSERVAÇÕES COMPLEMENTARES

POSSÍVEIS RISCOS QUE PODEM SER ENCONTRADOS DURANTE O PROJETO:

- O software pode não executar as funções de acordo com a sua especificação;
- O software pode não executar as funções pretendidas de acordo com as necessidades do usuário, do cliente ou dos stakeholders;
- Os tempos de resposta podem ser inadequados; • Atraso na entrega.

SOLUÇÕES:

- Acompanhar o software seja feito regularmente;
- Elaborar um relatório de defeitos;
- Todos da equipe saibam exatamente o que e como precisa ser feito no projeto;
- Seja feito uma análise constante para verificar se o software está de acordo com as regras do negócio;
- O Scrum Master auxilie sempre que necessário;
- Todos os envolvidos no projeto estejam cientes do prazo.



OBSERVAÇÕES COMPLEMENTARES

COBERTURA DE TESTES

Ao realizar o cálculo da cobertura de testes é possível verificar o quanto a API ServeRest está coberta pelos testes.

A seguir é apresentado uma tabela com os seguintes critérios:

- Path Coverage: cálculo baseado na quantidade de endpoints da API;
- Operator Coverage: cálculo baseado na quantidade de métodos existentes na API;
- Parameter Coverage: cálculo baseado na quantidade de parâmetros da API;
- Status Coverage: cálculo baseado na quantidade de status code da API.

Crítérios	API ServeRest	Automação	Cobertura
Path Coverage	9	9	100.00%
Operator Coverage	16	13	81.25%
Parameter Coverage	25	22	88.00%
Status Coverage	38	25	65.79%

CONCLUSÃO

Este documento tem como objetivo apresentar as mínimas atividades para a execução dos testes. É recomendável que, conforme as atividades sejam realizadas, esse documento seja enriquecido com novos processos necessários. É vedado, entretanto, a exclusão ou edição das informações originalmente presentes.

Este plano de testes deve ser disponibilizado para todas as partes interessadas.



API SERVEREST PLANO DE TESTE

Setembro, 2022

Feito por:

Karoline Nascimento D. da Silva
karoline.silva.pb@compasso.com.br