Roteiro de Testes - GOLDILOCKS

Metodologia Escolhida

Nesta abordagem, optei por combinar uma **heurística**, uma **estratégia de priorização** e uma **técnica de teste** com foco em valor. Essa combinação permite estruturar a exploração de forma equilibrada, direcionada aos riscos mais relevantes e com testes eficazes em campos críticos da aplicação.

- **Heurística utilizada: Goldilocks**
  Atua como guia de profundidade durante a exploração, ajudando a evitar tanto testes excessivos quanto testes superficiais. É especialmente útil para entradas com validações ou limites, como campos numéricos e fluxos com restrições de volume.
- **Estratégia: Baseada em Risco**
  Prioriza os pontos mais críticos ou suscetíveis a falhas no sistema. Com isso, o tempo de teste é melhor aproveitado e os feedbacks são gerados com mais relevância para o time de desenvolvimento.
- **Técnica aplicada: Análise de Valor Limite**
  Direciona os testes para os extremos de entrada (mínimo, máximo e inválidos), sendo ideal para validação de campos como CPF, senha, preço, quantidade e nome de produto.

> Obs: A estratégia baseada em risco não é uma técnica de execução, mas sim um critério de escolha que orienta onde concentrar os testes exploratórios com maior impacto.

Charter (Propósito da Exploração):

Explorar limites de entrada no cadastro de usuários

Objetivo:
Avaliar como o sistema lida com entradas mínimas, máximas e inválidas no endpoint de cadastro de usuários (`/usuarios`), especialmente nos campos: nome, email e senha.

Foco:

- Senhas fracas ou curtas podem comprometer a segurança da aplicação.
- Campos com comprimento excessivo podem quebrar layout ou causar falhas de banco.

Critérios de Encerramento do Teste

A sessão será encerrada **quando ocorrer o primeiro dos seguintes eventos**:

- **For encontrado pelo menos um bug ou comportamento inconsistente**, documentado e evidenciado.
- **O tempo limite de 40 minutos for atingido**, mesmo que nenhum bug seja identificado.
- **Todos os valores planejados de limite forem testados** (mínimo, máximo, inválido).

Configuração da Sessão de Testes

Ambiente de Teste

Parâmetro	Valor
Sistema Operacional	Windows 11
Navegador	Google Chrome 135 64 bits
Ferramenta	Postman
API Testada	Serverest
Tipo de Teste	Exploratório
Data e Hora	25/04/2025 - 11:00

Dados da Execução

Parâmetro	Valor
Tester	Anna Santoro
Duração	40 minutos
Técnica aplicada	Valor Limite
Heurística	Goldilocks
Estratégia	Baseada em Risco
Método de registro	Notas manuais + captura de tela

Relatório da Execução

Objetivo:

Explorar o cadastro de usuários na API Serverest, com foco em valores limite para os campos de entrada (nome, e-mail, senha).

Passos Realizados

1. **Teste 1:** Nome vazio (`""`) – Envio de JSON com campo `nome` em branco.
2. **Teste 2:** Nome com 1 caractere (`"A"`) – Teste abaixo do limite inferior válido.
3. **Teste 3:** Nome com 2 caracteres (`"AB"`) – Teste com o valor mínimo válido.

> A sessão foi encerrada após o Teste 3, por atingir o critério de encerramento: **foi identificado um bug de validação divergente entre front e API.**

Comparação Esperado vs. Obtido

Teste	Entrada	Esperado	Obtido	Status
		HTTP 400 – rejeição com mensagem clara	HTTP 400 – `"nome não pode ficar em`	

1	`"nome": ""`		`branco"` + `"administrador é obrigatório"`	✅ Correto
2	`"nome": "A"`	HTTP 400 – rejeição por entrada abaixo do limite	HTTP 400	✅ Correto
3	`"nome": "AB"`	HTTP 201 – cadastro bem-sucedido	✅ Interface: sucesso ❌ Postman: erro `"administrador é obrigatório"`	⚠️ Inconsistência

Observações Técnicas

- **Teste 1:** Embora o foco estivesse na rejeição do campo `nome`, foi retornado também erro para o campo `"administrador"`, que **não deveria ser obrigatório** segundo o comportamento da interface.
- **Teste 2:** Validação mínima de caracteres aplicada corretamente pelo back-end. O sistema rejeitou nomes com apenas 1 caractere, conforme esperado.
- **Teste 3:** Foi identificado um comportamento **inconsistente** entre a interface da aplicação (que realiza o cadastro com sucesso) e a API acessada diretamente via Postman (que exige o campo `"administrador"` explicitamente).

> Isso caracteriza um bug de validação divergente e encerrou a sessão conforme os critérios estabelecidos.

Bug Report: Campo "administrador" considerado obrigatório indevidamente

ID: BUG-GD-001
Título: API retorna erro 400 ao omitir o campo `"administrador"` no cadastro de usuário
Gravidade: Média
Prioridade: Média
Status: Aberto

Descrição do Problema

Durante a execução do caso de teste exploratório **CT03 - Nome com 2 caracteres**, foi identificado que a **API Serverest rejeita requisições de cadastro de usuário** que **não incluam o campo** `"administrador"`, mesmo que este campo seja tratado como **opcional** na interface web da aplicação.

O sistema retorna:

```
{ "administrador": "administrador é obrigatório" }
```

Esse comportamento **não é reproduzido na interface oficial da aplicação**, onde o campo `"administrador"` é opcional (controlado por checkbox) e, quando não marcado, **a submissão ocorre com sucesso**.

Passos para Reproduzir

1. Acesse o endpoint `POST https://serverest.dev/usuarios`
2. Envie o seguinte corpo da requisição:

```
{
```

```
  "nome": "AB",
  "email": "teste3@email.com",
  "password": "123456"
}
```

3. Observe a resposta da API.

Resultado Esperado

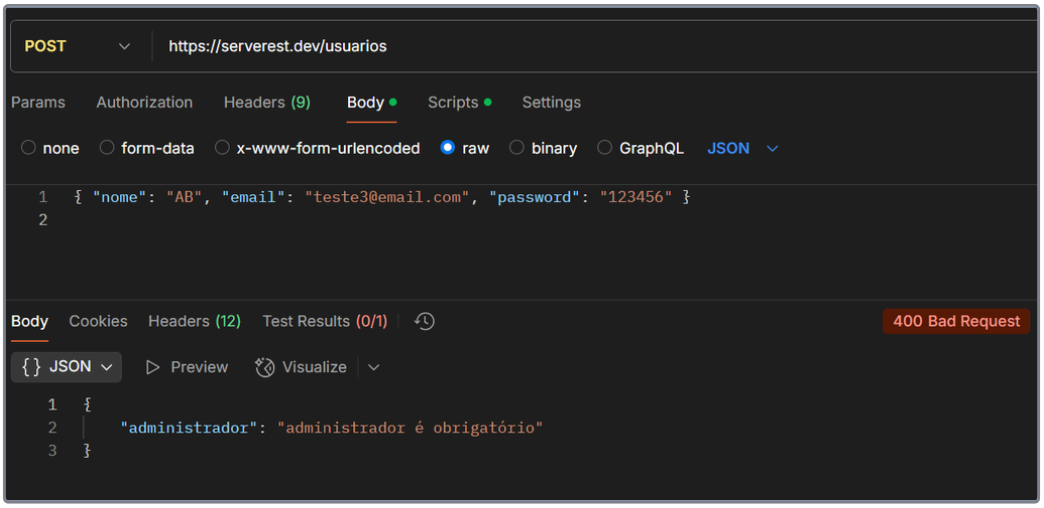
- A API deve considerar o campo `"administrador"` como **opcional**;
- Deve assumir o valor padrão `"false"` quando o campo não for enviado;
- Deve retornar status `201 Created` e realizar o cadastro normalmente.

Resultado Obtido

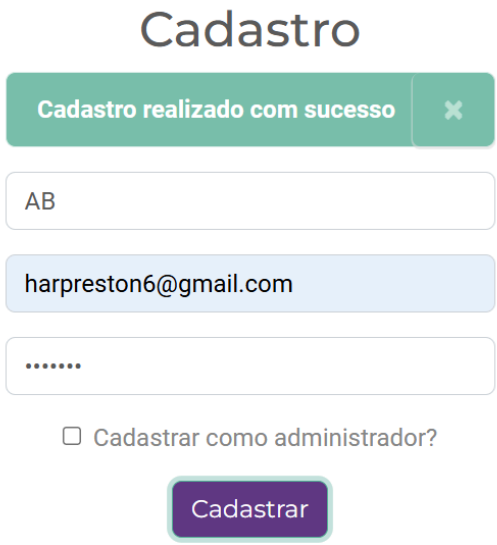
- A API retorna `400 Bad Request`;
- Mensagem: `"administrador é obrigatório"`.

Evidências

- **Postman:** erro 400 ao omitir `"administrador"`



- **Interface Web:** cadastro realizado com sucesso com `"AB"` e sem marcar “administrador”



Impacto

- Inconsistência entre front-end e API
- Testes automatizados falham mesmo com entrada válida
- Desalinhamento com boas práticas REST (campos booleanos opcionais devem ter default)
- Risco de rejeição indevida de dados legítimos via API externa ou scripts

---

Recomendação Técnica

- Alterar a validação do backend para que `"administrador"` seja opcional (`default: false`);
- Atualizar a documentação oficial da API refletindo esse comportamento;
- Adicionar testes automatizados cobrindo este cenário.

Resumo Executivo da Sessão

Item	Valor
Sessão	Cadastro de Usuário
Duração	40 minutos
Total de Bugs	1 (médio)
Pontos Fortes	Comportamento consistente nas rejeições previstas para entradas inválidas

Pontos Negativos	Inconsistência entre front-end e API