✅ 1. Teste de Validação de Entrada

Objetivo: Garantir que entradas como parâmetros da query string estejam sendo validadas corretamente.

Cenários:

- ✅ Entrada válida:
  `GET /dogs?name=Rex` → Deve retornar 200 com dados do cachorro.
- ❌ Entrada inválida:
  `GET /dogs?name=` ou `GET /dogs?name=@!` → Deve retornar 400 com mensagem de erro.

Como testar:

- Validar tipos, presença de campos obrigatórios, e formato adequado (ex: nomes sem caracteres especiais).
- Verifique se os erros são claros.

📊 2. Teste de Cálculo do Percentual de Raças com Maior Preferência de Adoção

Objetivo: Testar se a função que calcula os percentuais de adoção por raça está correta.

Cenário Exemplo:

```
[
  {"raca": "Labrador", "adocoes": 200},
  {"raca": "Poodle", "adocoes": 150},
  {"raca": "Vira-lata", "adocoes": 50}
]
```

Cálculo esperado:

- Total: 400
- Labrador: 50%
- Poodle: 37.5%

- Vira-lata: 12.5%

Testar:

- Com diferentes totais.
- Com total 0 (evitar divisão por zero).
- Com 100% de adoção em uma raça.

🔁 3. Teste de Conversão de Tipos

Objetivo: Garantir que campos numéricos passados como strings sejam corretamente convertidos ou tratados.

Testes:

- `GET /dogs?id=100` → OK.
- `GET /dogs?id="cem"` → Retornar erro (400) com mensagem de tipo inválido.

🔍 4. Teste de Busca de Item em Lista

Objetivo: Verificar se a busca por nome, raça ou ID está funcionando corretamente.

Exemplo:

- Criar lista mock: `[{"id":1,"name":"Rex"},{"id":2,"name":"Toby"}]`
- Buscar por `GET /dogs?name=Rex` → Deve retornar o item com ID 1.
- Buscar por nome inexistente → Deve retornar lista vazia ou 404 com mensagem “não encontrado”.

🧩 Testes de Usabilidade (Interface se houver Frontend)

📝 1. Navegação em Formulário

Objetivo: Verificar se é intuitivo preencher e enviar dados de cadastro/adoção de cachorro.

Critérios:

- Campos bem organizados: nome do cachorro, raça, idade, etc.
- Mensagens de erro claras em campos obrigatórios.
- Labels legíveis.

♿ 2. Acessibilidade

Objetivo: Garantir que pessoas com deficiência visual consigam usar a interface.

Verificar:

- Uso de leitores de tela (NVDA, VoiceOver).
- Contraste adequado nas cores.
- Botões com textos e ícones acessíveis.
- Suporte a navegação via teclado (tab, enter).

📱 3. Navegação Mobile

Objetivo: Avaliar a experiência em smartphones.

Testar:

- Responsividade do layout.
- Facilidade de clicar em botões.
- Legibilidade do conteúdo.

Ferramentas: Chrome DevTools (modo mobile).

🔄 4. Feedback Visual em Ações

Objetivo: Confirmar que o sistema responde visualmente às ações do usuário.

Verificar:

- Botões com efeito de clique.
- Ícones de carregamento ao buscar.
- Mensagens de sucesso ou erro após enviar formulário ou buscar cachorro.

🚀 Testes de Performance

⚡ 1. Tempo de Resposta da Página Web

Objetivo: Medir o tempo de carregamento do frontend.

Testar:

- Acesso via 4G e Wi-Fi.
- Carregamento completo deve ser <3s.

Ferramentas: Lighthouse, GTMetrix, DevTools.

📈 2. Escalabilidade da API

Objetivo: Avaliar desempenho da API com carga.

Etapas:

1. Enviar 50 requisições simultâneas: Verificar se todas respondem rápido (ex: < 500ms).
2. Subir para 100, depois 200.
3. Medir:
    - Tempo médio de resposta.

- Taxa de erros (timeout, 500).
- Estabilidade da API.