

# **PLANO DE TESTES DO SISTEMA**

## **1. Objetivo do Teste**

O objetivo deste plano de testes é garantir que o bot de gerenciamento de estoque funcione corretamente em todas as suas funcionalidades principais, validando cada comando, verificando a integridade dos dados salvos no arquivo JSON, confirmindo o comportamento esperado no Telegram e assegurando que erros e entradas inválidas sejam tratados adequadamente.

## **2. Escopo**

Os testes abrangerão:

- Comandos principais:  
/add, /sub, /set, /remove, /list, /help
- Validação de entradas (quantidades, itens, sintaxe)
- Persistência de dados em inventories.json
- Inventário separado por chat
- Mensagens de erro e respostas informativas
- Comportamento com e sem menção ao bot (caso seja habilitado)

## **3. Abordagem de Teste**

A abordagem utilizada será composta por:

- Testes funcionais: garantir que cada comando execute corretamente.
- Testes de validação: testar limites, formatos incorretos e entradas inválidas.
- Testes de integração: verificar persistência e leitura do JSON.
- Testes de comportamento: checar se o bot responde adequadamente aos usuários.
- Testes regressivos: garantir que novas funções não quebrem funções antigas.
- Testes automatizados (planejados): previstos no backlog e listados na Sprint 1.

# **CASOS DE TESTE**

### **CT-01 – Adicionar item válido (/add)**

**Objetivo:** Verificar se o bot adiciona um item com quantidade válida.

**Pré-condição:** Estoque vazio.

**Passos:** Enviar comando /add papel 3.

**Resultado Esperado:** Mensagem “Adicionou 3 unidades de papel.” e item salvo no JSON.

### **CT-02 – Adicionar item com quantidade inválida**

**Objetivo:** Testar comportamento do bot com quantidade negativa ou inválida.

**Passos:** Enviar comando /add detergente -2.

**Resultado Esperado:** Mensagem “Quantidade inválida.”

### **CT-03 – Subtrair quantidade válida (/sub)**

**Objetivo:** Verificar se o bot subtrai corretamente a quantidade informada.

**Pré-condição:** Item “papel” com quantidade 3.

**Passos:** Enviar comando /sub papel 1.

**Resultado Esperado:** Quantidade atualizada para 2 unidades e mensagem confirmando a operação.

### **CT-04 – Subtrair valor maior que o estoque**

**Objetivo:** Garantir que o sistema impeça valores negativos no inventário.

**Passos:** Enviar comando /sub papel 10.

**Resultado Esperado:** Mensagem “Não é possível subtrair essa quantidade.”

### **CT-05 – Definir quantidade exata (/set)**

**Objetivo:** Testar se o bot atualiza a quantidade exata de um item.

**Pré-condição:** Item já cadastrado no estoque.

**Passos:** Enviar comando /set papel 5.

**Resultado Esperado:** Mensagem “Agora papel possui 5 unidades.”

## **CT-06 – Remover item (/remove)**

**Objetivo:** Verificar se o bot remove um item do estoque.

**Pré-condição:** Item cadastrado no inventário.

**Passos:** Enviar comando /remove papel.

**Resultado Esperado:** Mensagem “Item papel removido.” e remoção no JSON.

## **CT-07 – Listar itens (/list)**

**Objetivo:** Exibir todos os itens cadastrados no estoque daquele chat.

**Passos:** Enviar comando /list.

**Resultado Esperado:** Lista com itens e quantidades; caso vazio, mensagem “O estoque está vazio.”

## **CT-08 – Comando incompleto ou sintaxe inválida**

**Objetivo:** Verificar resposta do bot quando o usuário envia comando incompleto.

**Passos:** Enviar /add sem parâmetros.

**Resultado Esperado:** Mensagem indicando uso correto do comando.

## **CT-09 – Inventário separado por chat**

- **Objetivo:** Garantir que cada chat possua seu próprio inventário independente.

**Pré-condição:** Dois chats distintos.

**Passos:** Adicionar itens diferentes em cada chat.

**Resultado Esperado:** Inventários não se misturam.

## **CT-10 – Persistência de dados (JSON)**

- **Objetivo:** Testar se o bot mantém o inventário após reiniciar.

**Passos:** Adicionar itens → reiniciar o bot.

**Resultado Esperado:** Conteúdo do inventário permanece igual ao salvo no JSON.

### **CT-11 – Exibir ajuda (/help)**

- **Objetivo:** Validar se o comando exibe corretamente todos os comandos e instruções.  
**Passos:** Enviar /help.  
**Resultado Esperado:** Mensagem com a lista completa de comandos e explicações.

### **CT-12 – Ignorar comandos sem menção**

- **Objetivo:** Verificar se o bot ignora comandos enviados sem @menção quando configurado assim.  
**Passos:** Enviar um comando sem mencionar o bot no grupo.  
**Resultado Esperado:** Nenhuma resposta do bot.