

# CATALOGUE

## Caso de Teste: Cadastrar Produto

Para cadastrar um produto, o usuário deverá informar pelo menos o nome e tipo do produto. O nome do produto deve conter pelo menos três caracteres, começando com uma letra. Caso o usuário não informe o fabricante, será utilizado um fabricante padrão de nome “”. Caso o usuário informe o nome do fabricante, este deverá ter pelo menos três caracteres, começando com uma letra. Se o fabricante informado for inexistente, será criado um novo fabricante, caso já exista, o produto será vinculado ao existente. Caso o produto já exista no servidor, será retornado o produto existente, mas se pelo menos 1 dos parâmetros for diferente de algum produto já existente, será criado um novo.

A entrada será um JSON de DtoProduto contendo um DtoTipoProduto e, possivelmente, um DtoFabricante.

Variáveis de Entrada	Condições	Equivalencia Válida	Equivalencia Inválida
dtoProduto	Não pode ser nulo	dtoProduto != null	dtoProduto == null
dtoProduto	Estar no formato JSON de DtoProduto	Sim	Não
dtoChave	Não pode ser nulo	dtoChave != null	dtoChave == null
dtoChave	Estar no formato JSON de DdtoChave	Sim	Não
dtoChave.idUsuario	inteiro maior que 0	idUsuario > 0	idUsuario <= 0
dtoChave.ultimoAcesso	string convertível para DateTime	Sim	Não
dtoChave.token	string com 4 caracteres (t)	t == 4	t != 4
dtoProduto.nome	string com primeiro caractere sendo uma letra	Sim	Não
	string com pelo menos 3 caracteres	t >= 3	t < 3
	string com no máximo 100 caracteres	t <= 100	t > 100
	espaços	Sim	Não

	consecutivos devem ser contados como 1 só		
dtoProduto.codigoDe Barras	string com no máximo 50 caracteres	t <= 50	t > 50
	Espaços são ignorados	Sim	Não
dtoProduto.idTipo	inteiro maior que 0	idTipo > 0	idTipo <= 0
dtoProduto.fabricante	Não pode ser nulo	fabricante != null	fabricante == null
dtoProduto.fabricante.fabricante	string com primeiro caractere sendo uma letra caso possua algum caractere	Sim	Não
	string com nenhum ou pelo menos 3 caracteres	t == 0    t >= 3	t < 3 && != 0
	string com no máximo 50 caracteres	t <= 50	t > 50
	espaços consecutivos devem ser contados como 1 só	Sim	Não

## INDICE

221.PV01: Criar um produto com valores mínimos	4
221.PV02: Criar um produto com valores máximos e fabricante com nome mínimo	5
221.PV03: Espaços consecutivos ignorados	6
221.PV04: Codigo de barras sem espaço e fabricante já cadastrado	7
221.PV05: Cadastrar Produto Repetido	8
<i>221.PI01: DtoChave nulo</i>	9
<i>221.PI02: Chave de acesso em formato diferente de JSON DtoChave</i>	0
<i>221.PI03: Chave de acesso com ID 0</i>	11
<i>221.PI04: Chave de acesso com ID inválido</i>	12
<i>221.PI05: Chave de acesso com token inválido</i>	13
<i>221.PI06: Chave de acesso com string de ultimo acesso não convertível para DateTime</i>	14
<i>221.PI07: DtoProduto nulo</i>	15
<i>221.PI08: DtoProduto em formato diferente de JSON DtoProduto</i>	16
221.PI09: Produto com nome começando com caractere que não seja letra	17
221.PI10: Produto com nome com menos caracteres do que o permitido	18
221.PI11: Produto com nome de tamanho maior do que o permitido	19
221.PI12: Produto com código de barras de tamanho maior que o permitido	20
<i>221.PI13: Produto com tipo inválido</i>	21
<i>221.PI14: Produto sem fabricante</i>	22
221.PI15: Produto com nome de fabricante com primeiro caractere inválido	23
221.PI16: Produto com nome de fabricante menor que o tamanho mínimo não nulo	24
221.PI17: Produto com fabricante com nome maior do que o permitido	25

OBS.: Os testes em itálico não são aplicáveis ao usuário.

## **TESTES DE CAIXA PRETA**

Obs.: Os valores em cinza não são informados pelo usuário.

### **VÁLIDOS**

#### **221.PV01 - Criar um produto com valores mínimos:**

##### **Descrição:**

Criar um produto inédito contendo nome com tamanho mínimo de 3 caracteres, sem código de barras e sem fabricante.

##### **Saída Esperada:**

Dados: Um DtoProduto com os dados enviados, tanto de produto quanto de fabricante, mudando apenas os IDs (IDs atualizados) e um novo DtoChave.

##### **Variaveis de Entrada:**

##### **Chave:**

idUsuario: maior que 0 e pertencente a algum usuário existente;

token:  $\geq 1000$  e  $\leq 9999$  (4 caracteres);

ultimoAcesso: string no padrão "dd/mm/aaaa hh:mm:ss";

OBS.: a chave deve existir no banco de dados.

##### **Produto:**

id: irrelevante;

nome: começando com letra e possuindo apenas 3 caracteres;

codigoDeBarras: vazio;

tipoCodigoDeBarras: irrelevante;

idTipo: valor definido por um enumerator;

tipo: irrelevante;

fabricante.id: irrelevante, possivelmente definido pelo auto-complete;

fabricante.fabricante: vazio;

##### **Pre-condições:**

O usuário deverá estar com o modal de Criar Produto aberto;

Nenhum produto com esses dados deve estar cadastrado no servidor;

##### **Pos-condições:**

Chave atualizada devolvida ao usuário;

Novo produto cadastrado no servidor;

Produto vinculado ao fabricante Nulo ("");

## **221.PV02 - Criar um produto com valores máximos e fabricante com nome mínimo:**

### **Descrição:**

Criar um produto inédito contendo nome com números e letras e tamanho máximo, com código de barras de tamanho máximo e fabricante inédito com tamanho mínimo não nulo.

### **Saída Esperada:**

Dados: Um DtoProduto com os dados enviados, mudando apenas os IDs (IDs atualizados) e um novo DtoChave.

### **Variaveis de Entrada:**

#### **Chave:**

idUsuario: maior que 0 e pertencente a algum usuário existente;

token: >= 1000 e <= 9999 (4 caracteres);

ultimoAcesso: string no padrão "dd/mm/aaaa hh:mm:ss";

OBS.: a chave deve existir no banco de dados.

#### **Produto:**

id: irrelevante;

nome: 100 caracteres com números e começando com letra;

codigoDeBarras: 50 caracteres;

tipoCodigoDeBarras: irrelevante;

idTipo: valor definido por um enumerator;

tipo: irrelevante;

fabricante.id: irrelevante, possivelmente definido pelo auto-complete;

fabricante.fabricante: apenas 3 caracteres começando com letra;

### **Pre-condições:**

O usuário deverá estar com o modal de Criar produto aberto;

Nenhum produto com esses dados deve estar cadastrado no servidor;

Nenhum fabricante com esse nome deve estar cadastrado no servidor;

### **Pos-condições:**

Chave atualizada devolvida ao usuário;

Novo produto cadastrado no servidor;

Novo fabricante cadastrado no servidor;

## 221.PV03 - Espaços consecutivos ignorados:

### Descrição:

Criar um produto inédito contendo nome com números, espaços, isolados e consecutivos, letras e tamanho máximo, com código de barras e fabricante inédito com tamanho máximo contendo espaços.

### Saída Esperada:

Um DtoProduto com os dados enviados, mudando apenas os IDs (IDs atualizados) e um novo DtoChave.

### Variáveis de Entrada:

#### Chave:

idUsuario: maior que 0 e pertencente a algum usuário existente;  
token: >= 1000 e <= 9999 (4 caracteres);  
ultimoAcesso: string no padrão "dd/mm/aaaa hh:mm:ss";  
OBS.: a chave deve existir no banco de dados.

#### Produto:

id: irrelevante;  
nome: contendo espaços consecutivos e, ao todo, 100 caracteres (contando espaços consecutivos apenas como 1) começando com letra;  
codigoDeBarras:  
tipoCodigoDeBarras: irrelevante;  
idTipo: valor definido por um enumerator;  
tipo: irrelevante;  
fabricante.id: irrelevante, possivelmente definido pelo auto-complete;  
fabricante.fabricante: contendo 50 caracteres começando com letra;

### Pre-condições:

O usuário deverá estar com o modal de Criar produto aberto;  
Nenhum produto com esses dados deve estar cadastrado no servidor;  
Nenhum fabricante com esse nome deve estar cadastrado no servidor;

### Pos-condições:

Chave atualizada devolvida ao usuário;  
Novo produto cadastrado, com tamanho máximo, no servidor;  
Novo fabricante cadastrado no servidor;

## **221.PV04 - Codigo de barras sem espaço e fabricante já cadastrado:**

### **Descrição:**

Criar um produto inédito com código de barras contendo espaços e um fabricante já existente

### **Saída Esperada:**

Um DtoProduto com os dados enviados, mudando apenas os IDs (IDs atualizados) e um novo DtoChave.

### **Variaveis de Entrada:**

#### **Chave:**

idUsuario: maior que 0 e pertencente a algum usuário existente;

token:  $\geq 1000$  e  $\leq 9999$  (4 caracteres);

ultimoAcesso: string no padrão "dd/mm/aaaa hh:mm:ss";

OBS.: a chave deve existir no banco de dados.

#### **Produto:**

id: irrelevante;

nome: começando com letra e de tamanho  $\geq 3$  e  $\leq 100$

codigoDeBarras: 50 caracteres (sem contar com os espaços) e contendo espaços;

tipoCodigoDeBarras: irrelevante;

idTipo: valor definido por um enumerator;

tipo: irrelevante;

fabricante.id: irrelevante, possivelmente definido pelo auto-complete;

fabricante.fabricante: nome já cadastrado com espaços, mas possuindo espaços consecutivos.

### **Pre-condições:**

O usuário deverá estar com o modal de Criar produto aberto;

Nenhum produto com esses dados deve estar cadastrado no servidor;

Fabricante com esse nome já deve estar cadastrado no servidor;

### **Pos-condições:**

Chave atualizada devolvida ao usuário;

Novo produto cadastrado, com código de barras sem espaço e vinculado ao fabricante já existente, no servidor;

Nenhum novo fabricante cadastrado no servidor;

## 221.PV05 - Cadastrar Produto Repetido:

### Descrição:

Criar um produto contendo nome com tamanho mínimo, sem código de barras e sem fabricante.

### Saída Esperada:

O DtoProduto do produto já existente no servidor correspondente aos dados enviados, mudando apenas os IDs (IDs atualizados) e um novo DtoChave.

### Variaveis de Entrada:

#### Chave:

idUsuario: maior que 0 e pertencente a algum usuário existente;

token: >= 1000 e <= 9999 (4 caracteres);

ultimoAcesso: string no padrão "dd/mm/aaaa hh:mm:ss";

OBS.: a chave deve existir no banco de dados.

#### Produto:

id: irrelevante;

nome:

codigoDeBarras:

tipoCodigoDeBarras: irrelevante;

idTipo: valor definido por um enumerator;

tipo: irrelevante;

fabricante.id: irrelevante, possivelmente definido pelo auto-complete;

fabricante.fabricante:

OBS.: O produto a ser cadastrado já deve existir no servidor;

### Pre-condições:

O usuário deverá estar com o modal de Criar produto aberto;

Produto com esses dados já deve estar cadastrado no servidor;

Fabricante com esse nome já deve estar cadastrado no servidor;

### Pos-condições:

Chave atualizada devolvida ao usuário;

Nenhum novo produto cadastrado no servidor;

Nenhum novo fabricante cadastrado no servidor;



# INVÁLIDOS

## 221.PI01 - DtoChave nulo:

### Descrição:

Tentar realizar uma ação sem enviar uma chave de acesso.

### Saída Esperada:

Mensagem de erro de código "100": "É necessário realizar login para utilizar o aplicativo!";

### Variáveis de Entrada:

#### Chave:

null

#### Produto:

id: irrelevante;

nome: começando com letra e possuindo apenas 3 caracteres;

codigoDeBarras: vazio;

tipoCodigoDeBarras: irrelevante;

idTipo: valor definido por um enumerador;

tipo: irrelevante;

fabricante.id: irrelevante, possivelmente definido pelo auto-complete;

fabricante.fabricante: vazio;

#### Pre-condições:

Irrelevante;

#### Pos-condições:

Usuário redirecionado para a tela de login;

## 221.PI02 - Chave de acesso em formato diferente de JSON DtoChave:

### Descrição:

Tentar realizar uma ação com uma chave de acesso que não seja um JSON de DtoChave.

### Saída Esperada:

Mensagem de erro de código "100": "É necessário realizar login para utilizar o aplicativo!";

### Variáveis de Entrada:

#### Chave:

{dtoChaveEmFormatoNaoJSON}

#### Produto:

id: irrelevante;

nome: começando com letra e possuindo apenas 3 caracteres;

codigoDeBarras: vazio;

tipoCodigoDeBarras: irrelevante;

idTipo: valor definido por um enumerator;

tipo: irrelevante;

fabricante.id: irrelevante, possivelmente definido pelo auto-complete;

fabricante.fabricante: vazio;

#### Pre-condições:

Irrelevante;

#### Pos-condições:

Usuário redirecionado para a tela de login;

## 221.PI03 - Chave de acesso com ID 0:

### Descrição:

Tentar realizar uma ação com uma chave de acesso inválida: ID < 1.

### Saída Esperada:

Mensagem de erro de código "100": "É necessário realizar login para utilizar o aplicativo!";

### Variáveis de Entrada:

#### Chave:

idUsuario: 0;

token: >= 1000 e <= 9999;

ultimoAcesso: string no padrão "dd/mm/aaaa hh:mm:ss";

#### Produto:

id: irrelevante;

nome: começando com letra e possuindo apenas 3 caracteres;

codigoDeBarras: vazio;

tipoCodigoDeBarras: irrelevante;

idTipo: valor definido por um enumerador;

tipo: irrelevante;

fabricante.id: irrelevante, possivelmente definido pelo auto-complete;

fabricante.fabricante: vazio;

#### Pre-condições:

Irrelevante;

#### Pos-condições:

Usuário redirecionado para a tela de login;

## 221.PI04 - Chave de acesso com ID inválido:

### Descrição:

Tentar realizar uma ação com uma chave de acesso inválida: não pertence a nenhum usuário.

### Saída Esperada:

Mensagem de erro de código "100": "É necessário realizar login para utilizar o aplicativo!";

### Variáveis de Entrada:

#### Chave:

**idUsuario:** ID referenciando usuário inexistente;

token:  $\geq 1000$  e  $\leq 9999$ ;

ultimoAcesso: string no padrão "dd/mm/aaaa hh:mm:ss";

#### Produto:

id: irrelevante;

nome: começando com letra e possuindo apenas 3 caracteres;

codigoDeBarras: vazio;

tipoCodigoDeBarras: irrelevante;

idTipo: valor definido por um enumerator;

tipo: irrelevante;

fabricante.id: irrelevante, possivelmente definido pelo auto-complete;

fabricante.fabricante: vazio;

#### Pre-condições:

Irrelevante;

#### Pos-condições:

Usuário redirecionado para a tela de login;

## 221.PI05 - Chave de acesso com token inválido:

### Descrição:

Tentar realizar uma ação com uma chave de acesso inválida: Token com quantidade de caracteres diferente de 4.

### Saída Esperada:

Mensagem de erro de código "100": "É necessário realizar login para utilizar o aplicativo!";

### Variáveis de Entrada:

#### Chave:

idUsuario: menor ou igual a 0;

token: 10000;

ultimoAcesso: string no padrão "dd/mm/aaaa hh:mm:ss";

#### Produto:

id: irrelevante;

nome: começando com letra e possuindo apenas 3 caracteres;

codigoDeBarras: vazio;

tipoCodigoDeBarras: irrelevante;

idTipo: valor definido por um enumerator;

tipo: irrelevante;

fabricante.id: irrelevante, possivelmente definido pelo auto-complete;

fabricante.fabricante: vazio;

### Pre-condições:

Irrelevante;

### Pos-condições:

Usuário redirecionado para a tela de login;

## 221.PI06 - Chave de acesso com string de ultimo acesso não convertível para DateTime:

### Descrição:

Tentar realizar uma ação com uma chave de acesso inválida: Ultimo Acesso não convertível para DateTime.

### Saída Esperada:

Mensagem de erro de código "100": "É necessário realizar login para utilizar o aplicativo!";

### Variaveis de Entrada:

#### Chave:

idUsuario: menor ou igual a 0;

token: >= 1000 e <= 9999;

ultimoAcesso: string que não esteja dentro do padrão "dd/mm/aaaa hh:mm:ss";

#### Produto:

id: irrelevante;

nome: começando com letra e possuindo apenas 3 caracteres;

codigoDeBarras: vazio;

tipoCodigoDeBarras: irrelevante;

idTipo: valor definido por um enumerator;

tipo: irrelevante;

fabricante.id: irrelevante, possivelmente definido pelo auto-complete;

fabricante.fabricante: vazio;

#### Pre-condições:

Irrelevante;

#### Pos-condições:

Usuário redirecionado para a tela de login;

## 221.PI07 - DtoProduto nulo:

### Descrição:

Tentar cadastrar um produto sem enviar o JSON de DtoProduto.

### Saída Esperada:

Mensagem de erro de código "101": "Dados foram perdidos durante o processo!";

### Variaveis de Entrada:

#### Chave:

idUsuario: menor ou igual a 0;

token: >= 1000 e <= 9999;

ultimoAcesso: string no padrão "dd/mm/aaaa hh:mm:ss";

OBS.: a chave deve existir no banco de dados.

### Produto:

null

### Pre-condições:

Usuário tentando cadastrar um produto;

### Pos-condições:

Mensagem de erro exibida na tela;

Produto não cadastrado;

## 221.PI08 - DtoProduto em formato diferente de JSON DtoProduto:

### Descrição:

Tentar cadastrar um produto sem passar um JSON de DtoProduto válido.

### Saída Esperada:

Mensagem de erro de código "101": "Dados foram perdidos durante o processo!";

### Variaveis de Entrada:

#### Chave:

idUsuario: menor ou igual a 0;

token: >= 1000 e <= 9999;

ultimoAcesso: string no padrão "dd/mm/aaaa hh:mm:ss";

OBS.: a chave deve existir no banco de dados.

### Produto:

{DtoProdutoEmUmFormatoNãoJSON}

### Pre-condições:

Usuário tentando cadastrar um produto;

### Pos-condições:

Mensagem de erro exibida na tela;

Produto não cadastrado;



## **221.PI09 - Produto com nome começando com caractere que não seja letra:**

### **Descrição:**

Usuário tenta cadastrar um produto cujo nome começa com um caractere que não é uma letra.

### **Saída Esperada:**

Mensagem de erro de código "104": "Existem campos com valores inválidos:\nNome do Produto";

### **Variáveis de Entrada:**

#### **Chave:**

idUsuario: menor ou igual a 0;

token: >= 1000 e <= 9999;

ultimoAcesso: string no padrão "dd/mm/aaaa hh:mm:ss";

OBS.: a chave deve existir no banco de dados.

#### **Produto:**

id: irrelevante;

**nome: começando com caractere que não seja letra e possuindo apenas 3 caracteres;**

codigoDeBarras: vazio;

tipoCodigoDeBarras: irrelevante;

idTipo: valor definido por um enumerator;

tipo: irrelevante;

fabricante.id: irrelevante, possivelmente definido pelo auto-complete;

fabricante.fabricante: vazio;

### **Pre-condições:**

Usuário tentando cadastrar um produto;

### **Pos-condições:**

Mensagem de erro exibida na tela;

Produto não cadastrado;

## 221.PI10 - Produto com nome com menos caracteres do que o permitido:

### Descrição:

Usuário tenta cadastrar um produto cujo nome possui menos de 3 caracteres (o mínimo permitido).

### Saída Esperada:

Mensagem de erro de código "104": "Existem campos com valores inválidos:\nNome do Produto";

### Variáveis de Entrada:

#### Chave:

idUsuario: menor ou igual a 0;

token: >= 1000 e <= 9999;

ultimoAcesso: string no padrão "dd/mm/aaaa hh:mm:ss";

OBS.: a chave deve existir no banco de dados.

#### Produto:

id: irrelevante;

**nome: começando com letra e possuindo apenas 2 caracteres;**

codigoDeBarras: vazio;

tipoCodigoDeBarras: irrelevante;

idTipo: valor definido por um enumerator;

tipo: irrelevante;

fabricante.id: irrelevante, possivelmente definido pelo auto-complete;

fabricante.fabricante: vazio;

#### Pre-condições:

Usuário tentando cadastrar um produto;

#### Pos-condições:

Mensagem de erro exibida na tela;

Produto não cadastrado;

## **221.PI11 - Produto com nome de tamanho maior do que o permitido:**

### **Descrição:**

Usuário tenta cadastrar um produto com o nome maior do que o permitido.

### **Saída Esperada:**

Mensagem de erro de código "104": "Existem campos com valores inválidos:\nNome do Produto";

### **Variaveis de Entrada:**

#### **Chave:**

idUsuario: menor ou igual a 0;

token: >= 1000 e <= 9999;

ultimoAcesso: string no padrão "dd/mm/aaaa hh:mm:ss";

OBS.: a chave deve existir no banco de dados.

#### **Produto:**

id: irrelevante;

**nome: começando com letra e possuindo 101 caracteres (espaços consecutivos contam como 1);**

codigoDeBarras: vazio;

tipoCodigoDeBarras: irrelevante;

idTipo: valor definido por um enumerator;

tipo: irrelevante;

fabricante.id: irrelevante, possivelmente definido pelo auto-complete;

fabricante.fabricante: vazio;

### **Pre-condições:**

Usuário tentando cadastrar um produto

### **Pos-condições:**

Mensagem de erro exibida na tela;

Produto não cadastrado;

## 221.PI12 - Produto com código de barras de tamanho maior que o permitido:

### Descrição:

Usuário tenta cadastrar um produto com o código de barras de tamanho maior do que o permitido.

### Saída Esperada:

Mensagem de erro de código "104": "Existem campos com valores inválidos:\nCódigo de Barras";

### Variáveis de Entrada:

#### Chave:

idUsuario: menor ou igual a 0;

token: >= 1000 e <= 9999;

ultimoAcesso: string no padrão "dd/mm/aaaa hh:mm:ss";

OBS.: a chave deve existir no banco de dados.

#### Produto:

id: irrelevante;

nome: começando com letra e possuindo apenas 3 caracteres;

**codigoDeBarras: contendo 51 caracteres;**

tipoCodigoDeBarras: irrelevante;

idTipo: valor definido por um enumerator;

tipo: irrelevante;

fabricante.id: irrelevante, possivelmente definido pelo auto-complete;

fabricante.fabricante: vazio;

### Pre-condições:

Usuário tentando cadastrar um produto

### Pos-condições:

Mensagem de erro exibida na tela;

Produto não cadastrado;

## 221.PI13 - Produto com tipo inválido:

### Descrição:

Usuário tenta cadastrar um produto sem informar o tipo

### Saída Esperada:

Mensagem de erro de código "104": "Existem campos com valores inválidos:\nTipo do Produto";

### Variáveis de Entrada:

#### Chave:

idUsuario: menor ou igual a 0;

token: >= 1000 e <= 9999;

ultimoAcesso: string no padrão "dd/mm/aaaa hh:mm:ss";

OBS.: a chave deve existir no banco de dados.

#### Produto:

id: irrelevante;

nome: começando com letra e possuindo apenas 3 caracteres;

codigoDeBarras: vazio;

tipoCodigoDeBarras: irrelevante;

**idTipo: valor definido por um enumerator, no entanto, fora dos limites min/max;**

tipo: irrelevante;

fabricante.id: irrelevante, possivelmente definido pelo auto-complete;

fabricante.fabricante: vazio;

#### Pre-condições:

Usuário tentando cadastrar um produto

#### Pos-condições:

Mensagem de erro exibida na tela;

Produto não cadastrado;

## 221.PI14 - Produto sem fabricante:

### Descrição:

Usuário tenta criar um produto mas, por algum motivo, o DtoProduto é enviado sem os dados do fabricante.

### Saída Esperada:

Mensagem de erro de código "101": "Dados foram perdidos durante o processo!";

### Variáveis de Entrada:

#### Chave:

idUsuario: menor ou igual a 0;

token: >= 1000 e <= 9999;

ultimoAcesso: string no padrão "dd/mm/aaaa hh:mm:ss";

OBS.: a chave deve existir no banco de dados.

#### Produto:

id: irrelevante;

nome: começando com letra e possuindo apenas 3 caracteres;

codigoDeBarras: vazio;

tipoCodigoDeBarras: irrelevante;

idTipo: valor definido por um enumerator;

tipo: irrelevante;

**fabricante: null;**

#### Pre-condições:

Usuário tentando cadastrar um produto

#### Pos-condições:

Mensagem de erro exibida na tela;

Produto não cadastrado;

## **221.PI15 - Produto com nome de fabricante com primeiro caractere inválido:**

### **Descrição:**

Usuário tenta cadastrar um produto cujo nome do fabricante começa com um caractere que não é uma letra.

### **Saída Esperada:**

Mensagem de erro de código "104": "Existem campos com valores inválidos:\nNome do Fabricante";

### **Variáveis de Entrada:**

#### **Chave:**

idUsuario: menor ou igual a 0;

token: >= 1000 e <= 9999;

ultimoAcesso: string no padrão "dd/mm/aaaa hh:mm:ss";

OBS.: a chave deve existir no banco de dados.

#### **Produto:**

id: irrelevante;

nome: começando com letra e possuindo apenas 3 caracteres;

codigoDeBarras: vazio;

tipoCodigoDeBarras: irrelevante;

idTipo: valor definido por um enumerator;

tipo: irrelevante;

fabricante.id: irrelevante, possivelmente definido pelo auto-complete;

**fabricante.fabricante: composto por apenas 3 caracteres e começando com caractere que não seja letra;**

#### **Pre-condições:**

Usuário tentando cadastrar um produto

#### **Pos-condições:**

Mensagem de erro exibida na tela;

Produto não cadastrado;

Fabricante não cadastrado;

## 221.PI16 - Produto com nome de fabricante menor que o tamanho mínimo não nulo:

### Descrição:

Usuário tenta cadastrar um produto com um fabricante de nome menor do que o tamanho mínimo não nulo.

### Saída Esperada:

Mensagem de erro de código "104": "Existem campos com valores inválidos:\nNome do Fabricante";

### Variáveis de Entrada:

#### Chave:

idUsuario: menor ou igual a 0;

token: >= 1000 e <= 9999;

ultimoAcesso: string no padrão "dd/mm/aaaa hh:mm:ss";

OBS.: a chave deve existir no banco de dados.

#### Produto:

id: irrelevante;

nome: começando com letra e possuindo apenas 3 caracteres;

codigoDeBarras: vazio;

tipoCodigoDeBarras: irrelevante;

idTipo: valor definido por um enumerator;

tipo: irrelevante;

fabricante.id: irrelevante, possivelmente definido pelo auto-complete;

**fabricante.fabricante: contendo 2 caracteres e começando com letra;**

#### Pre-condições:

Usuário tentando cadastrar um produto

#### Pos-condições:

Mensagem de erro exibida na tela;

Produto não cadastrado;

Fabricante não cadastrado;



## 221.PI17 - Produto com fabricante com nome maior do que o permitido:

### Descrição:

Usuário tenta cadastrar um produto com tamanho maior do que o permitido.

### Saída Esperada:

Mensagem de erro de código "104": "Existem campos com valores inválidos:\nNome do Fabricante";

### Variaveis de Entrada:

#### Chave:

idUsuario: menor ou igual a 0;

token: >= 1000 e <= 9999;

ultimoAcesso: string no padrão "dd/mm/aaaa hh:mm:ss";

OBS.: a chave deve existir no banco de dados.

#### Produto:

id: irrelevante;

nome: começando com letra e possuindo apenas 3 caracteres;

codigoDeBarras: vazio;

tipoCodigoDeBarras: irrelevante;

idTipo: valor definido por um enumerator;

tipo: irrelevante;

fabricante.id: irrelevante, possivelmente definido pelo auto-complete;

**fabricante.fabricante: contendo 51 caracteres e começando com letra;**

#### Pre-condições:

Usuário tentando cadastrar um produto;

#### Pos-condições:

Mensagem de erro exibida na tela;

Produto não cadastrado;