

09

## Mão a obra: exceções unchecked

Continuando falando sobre exceções e Faults. Vamos testar uma exceção do tipo *unchecked* (ou seja *unmodeled*).

No projeto já preparamos um validador do item (a classe `ItemValidador`). Use este validador no método `cadastrarItem` da classe `EstoqueWS`:

```
@WebMethod(operationName="CadastrarItem")
public Item cadastrarItem(@WebParam(name="tokenUsuario", header=true) TokenUsuario token, @WebParam(name="item") Item item) {
    System.out.println("Cadastrando " + item + ", " + token);

    if(! new TokenDao().ehValido(token)) {
        throw new AutorizacaoException("Autorizacao falhou");
    }

    //novo
    new ItemValidador(item).validate();

    this.dao.cadastrar(item);
    return item;
}
```

A única linha nova é: `new ItemValidador(item).validate()`. Ao chamar o validador lança uma `ItemValidadorException` caso o item esteja inválido. Essa exceção é *unchecked*, ou seja, não faz parte do WSDL. Faça o teste, envie uma mensagem SOAP com **um token válido mas o item inválido**, por exemplo:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ws="http://ws.estoque.com.br/">
<soapenv:Header>
<ws:tokenUsuario soapenv:mustUnderstand="1">
<token>AAA</token>
<dataValidade>2015-12-31T00:00:00</dataValidade>
</ws:tokenUsuario>
</soapenv:Header>
<soapenv:Body>
<ws:CadastrarItem>
<!--Optional:-->
<item>
<codigo>M</codigo>
<nome>MEAN</nome>
<tipo>Livro</tipo>
<quantidade>5</quantidade>
</item>
</ws:CadastrarItem>
</soapenv:Body>
</soapenv:Envelope>
```

Nessa mensagem SOAP o código está errado pois possui apenas 1 char. Teste agora :)

