****

**ESCOLA SENAI ANTÔNIO PAULO SKAF**

**Relatório de Testes Unitários**

**Murilo Souza Almeida**

**Teste de Controle de Inventário**

**São Caetano do Sul**

**08/2024**

SUMÁRIO

[INTRODUÇÃO 3](#_Toc173768745)

[OBJETIVOS DOS TESTES 4](#_Toc173768746)

[Objetivos Gerais 4](#_Toc173768747)

[Objetivos Específicos: 4](#_Toc173768748)

[AMBIENTE DE TESTES 5](#_Toc173768749)

[METODOLOGIA DE TESTES 6](#_Toc173768750)

[Teste 1: ValidarAAdicaoDoProdutoAoInventario() 6](#_Toc173768751)

[Teste 2: ValidarABuscaDaQuantidadeDoProdutoNoInventario() 7](#_Toc173768752)

[RESULTADO DOS TESTES 9](#_Toc173768753)

[Teste 1: ValidarAAdicaoDoProdutoAoInventario() 9](#_Toc173768754)

[Teste 2: ValidarABuscaDaQuantidadeDoProdutoNoInventario() 9](#_Toc173768755)

[CONCLUSÃO 10](#_Toc173768756)

[ANEXOS 11](#_Toc173768757)

# INTRODUÇÃO

Durante o desenvolvimento de plataformas e projetos complexos, a quantidade necessária de tempo a ser gasto para elaborar funcionalidades, telas, bases de dados e integrações dentro do programa se torna cada vez maior. Tendo isto em mente, fazer os testes de tais funcionalidades uma por uma de forma manual se faz um procedimento inviável.

Muitas das vezes, realizar os testes do projeto, e corrigir seus erros caso precise, acaba sendo mais demorado que o desenvolvimento das funcionalidades do projeto em si, por conta disso uma estruturação mais avançada de realização de testes em programas vem sendo uma técnica cada vez mais importante e utilizada durante o processo de criação de uma nova plataforma.

Existem diversos tipos de testes, cada um com um propósito e uma finalidade específica, neste projeto foi utilizado do teste unitário para a homologação de um sistema de adição e gerenciamento de determinados produtos em um inventário por meio de duas funções simples. As especificações e objetivos dos testes realizados assim como suas metodologias e seus resultados serão abordados neste relatório, por fim será feita uma conclusão a respeito dos resultados esperados e obtidos, assim como uma discussão sobre o uso da ferramenta xUnit com Csharp para realização de testes unitários.

# OBJETIVOS DOS TESTES

Os objetivos dos testes realizados são:

## Objetivos Gerais

Realizar os testes de uma pequena aplicação de gerenciamento de inventário de uma loja, tal aplicação é responsável por: adicionar novos produtos ao inventário da loja; adicionar quantidades ao estoque de um determinado produto já presente no inventário; e trazer informações a respeito da quantidade de determinado produto, em estoque, que esteja presente no inventário [Figura 1].

Para isso foram desenvolvidas funções e fornecidos parâmetros para o teste de dois métodos do projeto. O primeiro método sendo um método para adicionar produtos ao inventário, assim como adicionar quantidades de um produto já existente [Figura 2]. Já o segundo método visa realizar a busca e o gerenciamento das quantidades de determinados produtos que estão presentes no inventário [Figura 3].

## Objetivos Específicos:

* Usar o xUnit na realização de testes unitários em aplicações c#.
* Fazer a homologação de um projeto para gerenciamento de um inventário de produtos de uma loja
* Testar a funcionalidade de adicionar um novo produto ao inventário
* Testar a funcionalidade de adicionar quantidades ao estoque de um produto já presente no inventário da loja.
* Testar a funcionalidade de busca e gerenciamento da quantidade de determinado produto presente no inventário a partir de seu nome.
* Discutir os resultados obtidos, assim como a utilização do xUnit para a realização de tais testes.

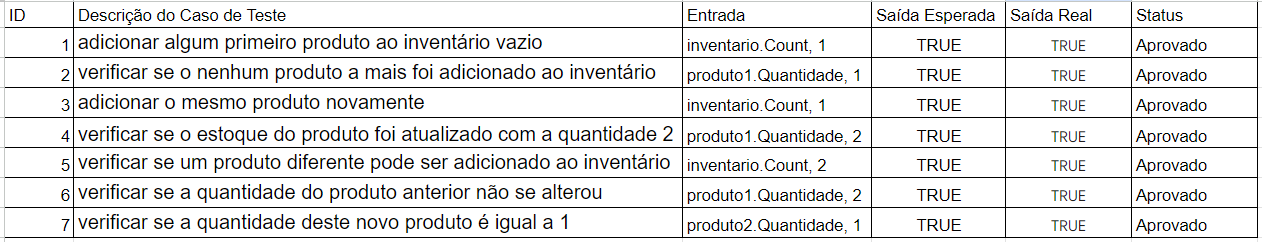
# AMBIENTE DE TESTES

|  |  |
| --- | --- |
| Sistema Operacional | Windows: Edição Windows 11 Pro; Versão: 23H2 |
| Versão do .NET Core/Framework | .NET 8.0 |
| Ferramenta de Teste | xUnit |
| IDE Utilizada | Visual Studio 2022 |

# METODOLOGIA DE TESTES

Para realizar os testes e validar as funções da aplicação em questão, foram criadas duas funções com annotations para indicar que ambas fazem parte de um projeto xUnit, a “ValidarAAdicaoDoProdutoAoInventario()” e a “ValidarABuscaDaQuantidadeDoProdutoNoInventario()” (fotos em anexo).

## Teste 1: ValidarAAdicaoDoProdutoAoInventario()



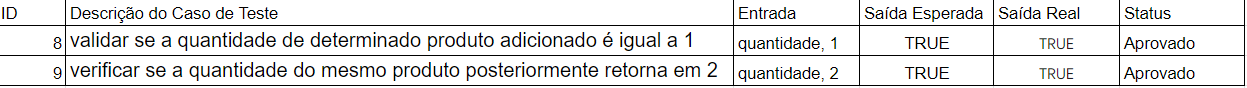
A primeira função tendo como objetivo realizar 7 testes simples: adicionar algum primeiro produto ao inventário vazio; verificar se o nenhum produto a mais foi adicionado ao inventário; adicionar o mesmo produto novamente; verificar se o estoque do produto foi atualizado com a quantidade 2; verificar se um produto diferente pode ser adicionado ao inventário; verificar se a quantidade do produto anterior não se alterou; verificar se a quantidade deste novo produto é igual a 1. [Figura 4]

Para isso, a função instanciará uma lista de produtos vazia para simular o inventário, em seguida irá incluir um novo produto pela função estática AdicionarProdutoAoInventario(string nomeDoProduto, List<Produto> inventário) que recebe como parâmetro uma string com o nome do produto a ser adicionado ã uma lista de produtos, este sendo o inventário e sendo também o segundo parâmetro a ser fornecido. Em seguida é feita uma busca no inventário para acessar ao novo produto e armazená-lo em uma variável, só então os dois primeiros testes são realizados.

O primeiro irá verificar se algum produto foi de fato armazenado na lista, enquanto o segundo irá verificar se a quantidade inicial do produto adicionado é igual a 1. Ao final dos dois códigos, a função continua adicionando mais um produto ao inventário pelo mesmo método anterior, no entanto este com nome o mesmo nome que o anterior, o que abre espaço para os dois testes seguintes, que irão validar se a quantidade do produto aumentou para 2, e se a quantidade de produtos adicionados no inventário não se alterou, indicando que uma unidade do produto foi adicionada ao estoque do mesmo.

Em seguida a sequência de etapas segue em frente adicionando um novo produto ao inventário com o nome diferente do primeiro produto adicionado, e o busca para guardar em uma variável. Por fim são feitos os três últimos testes sendo responsáveis por, respectivamente, verificar se a quantidade de elementos no inventário aumentos, verifica se a quantidade do primeiro produto adicionada se mantém como 2 e verificar se o novo produto adicionado possui quantidade igual a 1. E só assim a primeira função da aplicação está de fato validada.

## Teste 2: ValidarABuscaDaQuantidadeDoProdutoNoInventario()



Já a segunda função é mais simples, seu objetivo é testar a busca da quantidade de determinado produto presente no inventário, para isso foram feitas 2 validações, sendo elas: validar se a quantidade de determinado produto adicionado é igual a 1; verificar se a quantidade do mesmo produto posteriormente retorna em 2. Começando a função, o primeiro passo é instanciar uma nova lista que simulará o inventário. Em seguida, adicionamos um novo produto ao inventário que será instanciado com a quantidade 1. [Figura 5]

Aí entra a primeira validação onde irá utilizar a função “BuscarQuantidadesDeUmProdutoNoInventario(string nomeDoProduto, List<Produto> inventario)” que recebe como parâmetro qual o nome do produto a ser buscado no inventário, e o segundo referencia o inventário em si. Ao chamar a função, a quantidade de produtos presente no inventário é retornada e armazenada em uma variável que será usada para a validação, se a quantidade buscada for 1, a função já estará funcionando satisfatoriamente.

No entanto para confirmar de vez sua funcionalidade, o mesmo produto é adicionado novamente ao inventário aumentando sua quantidade em 1, se tornando agora 2. Em seguida a variável de quantidade do produto é atualizada e caso seja igual a 2 então o primeiro teste é validado completando assim a homologação da aplicação de Controle de Inventário, e sua respectiva aplicação de testes.

# RESULTADO DOS TESTES

## Teste 1: ValidarAAdicaoDoProdutoAoInventario()

|  |  |
| --- | --- |
| Total de Casos de Teste | 7 |
| Casos Aprovados | 7 |
| Casos Reprovados | 0 |
| Percentual de Sucesso | 100% |

## Teste 2: ValidarABuscaDaQuantidadeDoProdutoNoInventario()

|  |  |
| --- | --- |
| Total de Casos de Teste | 2 |
| Casos Aprovados | 2 |
| Casos Reprovados | 0 |
| Percentual de Sucesso | 100% |

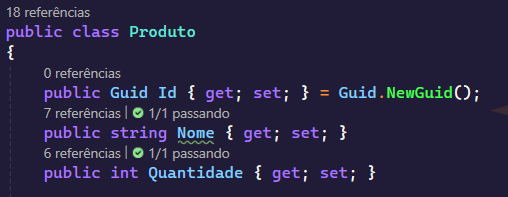
# CONCLUSÃO

Tendo em vista os resultados obtidos e o procedimento necessário para alcança-los, conclui-se que a ferramenta xUnit é uma ótima opção para realizar testes unitários de funções presentes em aplicações simples. Por sua fácil utilização e métodos de comparação de resultado simples, como comparação entre duas variáveis ou comparações com valores lógicos ou valores vazios.

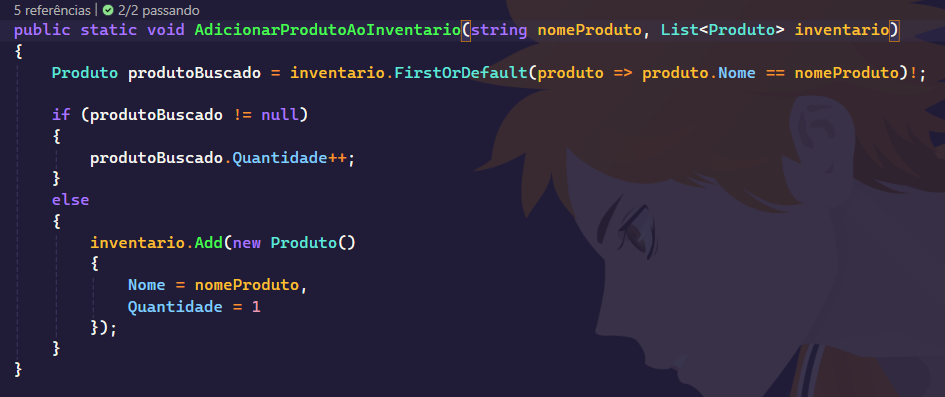
Sua rápida implementação e simples estrutura o torna uma opção muito prática durante os testes rápidos, fazendo com que os testes sejam feitos de maneira muito mais eficiente do que se os realizasse de maneira manual. E por fim há a opção de testes multilinhas para vários casos com as annotations [Theory] e [InLineData] onde há a possibilidade de realizar vários testes com entradas de dados diferentes.

# ANEXOS

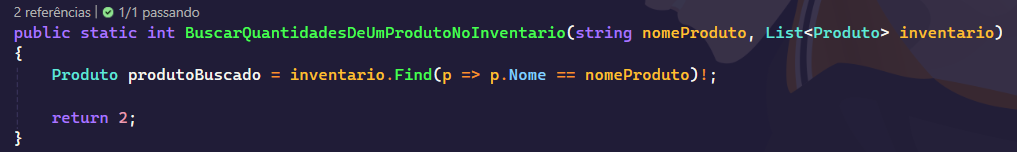
[Figura 1]:



[Figura 2]:



[Figura 3]:



[Figura 4]:



[Figura 5]:

