

Tutorial: .NET Core Test Explorer

O **.NET Core Test Explorer** é uma extensão essencial para Visual Studio Code que revoluciona a forma como desenvolvedores trabalham com testes. Esta ferramenta permite **descobrir, visualizar e executar testes** .NET diretamente na interface do VS Code, eliminando a necessidade de usar a linha de comando e proporcionando uma experiência de desenvolvimento mais fluida e intuitiva.

A extensão oferece suporte abrangente aos frameworks de teste mais populares do ecossistema .NET, incluindo **xUnit** (amplamente utilizado em projetos modernos), **NUnit** (tradicional e robusto) e **MSTest** (nativo da Microsoft). Esta compatibilidade garante que você possa utilizar o Test Explorer independentemente da sua escolha de framework, mantendo a mesma experiência consistente e produtiva em todos os seus projetos de desenvolvimento.

Onde Encontrar o Test Explorer

Painel de Testes Nativo

O VS Code possui um painel de testes integrado que você pode acessar através do ícone de **béquer/frasco** na barra lateral esquerda. Este é o método mais direto e moderno para acessar seus testes.

- Clique no ícone do béquer na barra lateral
- O painel "Testing" será aberto
- Visualize todos os testes em formato de árvore

Test Explorer (Extensão)

Se você instalou a extensão Test Explorer separadamente, encontrará um painel adicional específico para esta ferramenta. Esta opção oferece recursos estendidos e personalizações avançadas.

- Procure por "Test Explorer" na barra lateral
- Acesso a configurações específicas da extensão
- Interface otimizada para projetos complexos

A escolha entre essas opções depende das suas necessidades específicas. O painel nativo é suficiente para a maioria dos cenários, enquanto a extensão Test Explorer oferece funcionalidades adicionais para projetos mais complexos que requerem configurações personalizadas ou workflows específicos de teste.

Estrutura Visual dos Testes

O Test Explorer apresenta uma organização hierárquica clara e intuitiva dos seus testes, facilitando a navegação e compreensão da estrutura do projeto. Esta organização visual permite que você identifique rapidamente a localização de cada teste e compreenda as relações entre diferentes componentes do seu conjunto de testes.

01

Nível do Projeto

Associations.Domain.Tests

– O nível mais alto representa o projeto de testes completo, mostrando o nome do assembly e agrupando todos os testes relacionados.

02

Nível da Classe

MoneySpecs / OrderSpecs

– Cada classe de teste aparece como uma pasta, organizando logicamente os testes por funcionalidade ou entidade testada.

03

Nível do Método

✓

Ctor_ValorNegativo_DeveLancarArgumentOutOfRangeException – Os testes individuais são exibidos como folhas da árvore, com nomes descritivos que indicam o cenário testado.


Esta estrutura hierárquica não apenas facilita a navegação, mas também permite execução granular dos testes. Você pode escolher executar um teste específico, todos os testes de uma classe, ou todos os testes do projeto, proporcionando flexibilidade total no seu processo de desenvolvimento e depuração.

Como Executar Testes

O Test Explorer oferece múltiplas formas de executar seus testes, cada uma adequada para diferentes situações e necessidades de desenvolvimento. Esta flexibilidade permite que você otimize seu workflow de acordo com o contexto específico do seu trabalho, seja durante o desenvolvimento inicial, depuração ou validação final.




Teste Individual

Passe o mouse sobre o nome do teste específico que deseja executar. Um ícone de **Play**  aparecerá ao lado. Clique nele para executar apenas esse teste, ideal para desenvolvimento focado ou depuração específica.




Projeto Inteiro

Use o  no topo do painel para executar todos os testes de todos os arquivos do projeto. Esta opção é ideal para validação completa antes de commits ou releases.




Classe Completa

Clique no  ao lado do nome da classe (como OrderSpecs). Todos os testes dessa classe serão executados em sequência, perfeito para validar uma funcionalidade completa após implementação.



Modo Debug

Utilize o ícone de **Debug**  em vez do Play. Este modo permite colocar breakpoints, inspecionar variáveis e acompanhar o fluxo de execução passo a passo durante os testes.

Interpretando os Resultados

Após a execução dos testes, o Test Explorer fornece feedback visual imediato através de um sistema de ícones intuitivos. Compreender esses indicadores é crucial para interpretar rapidamente o estado dos seus testes e identificar áreas que necessitam atenção. Cada ícone transmite informações específicas sobre o resultado da execução, permitindo uma análise eficiente do status geral da qualidade do código.



Passou

Teste executado com sucesso

O teste foi executado sem erros e todas as asserções foram validadas. Indica que a funcionalidade está funcionando conforme esperado.



Falhou

Teste encontrou um erro

O teste foi executado mas uma ou mais asserções falharam, ou uma exceção não esperada foi lançada. Requer investigação e correção.



Ignorado

Teste marcado para ser pulado

O teste está marcado com atributos como [Skip] ou [Ignore] e foi intencionalmente ignorado durante a execução.



Não Executado

Teste ainda não foi rodado

O teste foi descoberto mas ainda não foi executado. Estado inicial de todos os testes após carregamento do projeto.



Executando

Teste em andamento

Indica que o teste está sendo executado no momento. Estado transitório durante o processo de execução.

Recursos Úteis do Test Explorer



Atualizar Lista

Clique no ícone de **refresh** (🔄) no topo do painel quando adicionar novos testes e eles não aparecerem automaticamente. Esta funcionalidade força a redescoberta de todos os testes no projeto.



Filtrar Testes

Utilize a caixa de busca no topo do painel para localizar testes específicos. Digite parte do nome do teste, da classe, ou do namespace para filtrar e encontrar rapidamente o que procura.



Detalhes do Erro

Quando um teste falha (❌), clique nele para ver informações detalhadas no painel inferior, incluindo mensagem de erro, stack trace e comparação entre valores esperados e recebidos.



CodeLens

Diretamente no arquivo de código, aparecem links **Run | Debug** acima de cada método de teste, permitindo execução imediata sem sair do contexto do código.

Dica: O recurso CodeLens é especialmente útil durante o desenvolvimento, pois permite executar testes específicos sem interromper o fluxo de codificação. Configure-o adequadamente para maximizar sua produtividade.

Configurações Recomendadas

Para otimizar sua experiência com o Test Explorer, é fundamental configurar adequadamente as preferências do VS Code. Essas configurações garantem que a extensão funcione de forma eficiente e atenda às suas necessidades específicas de desenvolvimento. As configurações apropriadas podem significativamente melhorar sua produtividade e reduzir atritos no workflow de testes.

Adicione as seguintes configurações ao seu `settings.json`. Acesse através de **Cmd+Shift+P** → *"Preferences: Open User Settings (JSON)"*.

```
{
  // Padrão para encontrar projetos de teste
  "dotnet-test-explorer.testProjectPath": "**/*Tests.csproj",

  // Observa mudanças e atualiza automaticamente
  "dotnet-test-explorer.autoWatch": true,

  // Expande automaticamente a árvore de testes
  "dotnet-test-explorer.autoExpandTree": true,

  // Mostra ícones CodeLens nos arquivos de teste
  "dotnet-test-explorer.codeLens": true,

  // Executa testes após build
  "dotnet-test-explorer.runAfterBuild": false
}
```

testProjectPath

Define o padrão de busca para localizar projetos de teste automaticamente. O padrão `**/*Tests.csproj` funciona para a maioria das convenções de nomenclatura.

autoWatch

Habilita o monitoramento automático de mudanças nos arquivos, atualizando a lista de testes conforme você modifica o código, mantendo sempre sincronizado.

codeLens

Ativa os links **"Run | Debug"** diretamente no código, proporcionando acesso instantâneo à execução de testes sem sair do contexto de desenvolvimento.

Workflow Recomendado (TDD)

O Test-Driven Development (TDD) é uma metodologia poderosa que se beneficia enormemente das funcionalidades do Test Explorer. Este approach promove código mais robusto, design mais limpo e maior confiança nas modificações. O Test Explorer facilita a implementação do TDD fornecendo feedback visual imediato e execução rápida de testes.

● RED

Escreva um teste que falha

Comece sempre escrevendo um teste para a funcionalidade que ainda não existe. Este teste deve falhar inicialmente, confirmando que está testando algo novo.

```
[Fact]
public void NovoTeste_Cenario_Comportamento()
{
    // Escreva o teste primeiro
    // Este teste deve falhar
}
```

● GREEN

Implemente o código mínimo

Execute o teste (▶) e veja ele falhar (✖).

Em seguida, escreva apenas o código necessário para fazer o teste passar (✔).

Não se preocupe com elegância nesta etapa.

● REFACTOR

Melhore o código

Com o teste passando, refatore o código para melhorar sua qualidade, mantendo os testes executando continuamente para garantir que não quebrou nada.

📌 **Vantagem do Test Explorer no TDD:** A execução visual e instantânea permite ver imediatamente o resultado de cada etapa do ciclo, tornando o processo mais intuitivo e produtivo.

Atalhos de Teclado Úteis

Dominar os atalhos de teclado é essencial para maximizar sua produtividade com o Test Explorer. Estes comandos permitem navegar e executar testes rapidamente sem interromper seu fluxo de desenvolvimento. A memorização destes atalhos pode significativamente acelerar seu processo de desenvolvimento orientado por testes.

Ação	macOS	Windows/Linux
Abrir painel de testes	Cmd+; ou Cmd+T	Ctrl+; ou Ctrl+T
Executar teste no cursor	Usar CodeLens	Usar CodeLens
Reexecutar último teste	Configurável	Configurável
Executar todos os testes	Cmd+Shift+;	Ctrl+Shift+;

Personalização de Atalhos

Acesse **Cmd+Shift+P** → "Preferences: Open Keyboard Shortcuts" para customizar atalhos específicos para suas necessidades de desenvolvimento.

Atalhos Context-Aware

Muitos atalhos funcionam de forma contextual. Quando o cursor está em um método de teste, os comandos de execução aplicam-se automaticamente a esse teste específico.

Pratique estes atalhos regularmente para desenvolver memória muscular. Com o tempo, você executará testes sem quebrar a concentração, mantendo o foco no desenvolvimento de código de qualidade.

Problemas Comuns e Soluções

Durante o uso do Test Explorer, é comum encontrar alguns obstáculos que podem prejudicar a produtividade. Conhecer as soluções para estes problemas frequentes permite resolver rapidamente as situações e manter o foco no desenvolvimento. A maioria dos problemas tem origem em configurações inadequadas ou sincronização entre o VS Code e o projeto .NET.

? Testes não aparecem?

Sintoma: O painel Test Explorer está vazio ou não mostra todos os testes esperados.

✓ Soluções:

1. Compile o projeto: `dotnet build`
2. Clique no botão "Refresh" no painel
3. Recarregue a janela: **Cmd+Shift+P** → *"Developer: Reload Window"*
4. Verifique se o `.csproj` possui as dependências de teste corretas
5. Confirme se os arquivos de teste seguem a convenção de nomenclatura configurada

? Teste aparece mas não executa?

Sintoma: Os testes são visíveis na árvore, mas não executam quando clicados ou ficam em estado permanente de "executando".

✓ Soluções:

1. Verifique se não há erros de compilação no projeto
2. Confirme que os atributos `[Fact]` ou `[Theory]` estão presentes
3. Execute via terminal para identificar erros: `dotnet test`
4. Verifique se as dependências do framework de teste estão atualizadas

? CodeLens não aparece?

Sintoma: Os links "Run | Debug" não aparecem acima dos métodos de teste no código.

✓ Solução:

- Ative nas configurações: `"dotnet-test-explorer.codeLens": true`
- Reinicie o VS Code após alterar a configuração
- Verifique se a extensão está ativa e atualizada

Benefícios de Usar o Test Explorer

A adoção do Test Explorer transforma completamente a experiência de desenvolvimento orientado por testes, oferecendo vantagens significativas que impactam diretamente na produtividade e qualidade do código. Estes benefícios se estendem desde melhorias na experiência do desenvolvedor até impactos positivos na manutenibilidade e confiabilidade dos projetos de software.



Feedback Visual Imediato

Veja instantaneamente quais testes passam, falham ou estão sendo executados através de indicadores visuais claros e intuitivos. Esta retroalimentação visual elimina a necessidade de interpretar saídas textuais complexas.



Integração Completa

Execute testes diretamente do editor sem necessidade de alternar para terminal ou outras ferramentas. Mantenha o contexto de desenvolvimento e concentração no código.



Debug Integrado

Coloque breakpoints, inspecione variáveis e investigue problemas diretamente durante a execução dos testes. Identifique e resolva problemas mais rapidamente.



Organização Clara

Visualize a estrutura completa dos testes em formato hierárquico, facilitando navegação e compreensão da cobertura de testes do projeto.



Produtividade Elevada

Execute seletivamente apenas os testes relevantes para o desenvolvimento atual, economizando tempo e focando no que importa no momento específico.



TDD Facilitado

Acelere o ciclo de desenvolvimento guiado por testes com execução rápida e feedback imediato, tornando o processo Red-Green-Refactor mais fluido e intuitivo.

Dica para Alunos

O Test Explorer representa uma evolução significativa na forma como estudantes e novos desenvolvedores interagem com testes de software. Esta ferramenta transforma uma atividade tradicionalmente baseada em comandos de terminal e saídas textuais em uma experiência **visual e interativa** que facilita enormemente o processo de aprendizado.



Interface Visual

As **cores intuitivas** substituem texto complexo do terminal. Verde para sucesso, vermelho para falha, azul para não executado. Esta codificação visual acelera a compreensão e reduz a curva de aprendizado.



Estrutura Organizada

A **árvore hierárquica** organizada por namespace e classe oferece uma visão clara da arquitetura de testes, ajudando estudantes a compreender boas práticas de organização de código.



⚡ Execução Seletiva

Execute **testes específicos** com um clique, permitindo foco em funcionalidades individuais durante o aprendizado. Ideal para estudar o comportamento de métodos específicos sem ruído de outros testes.



🔍 Detalhes Completos

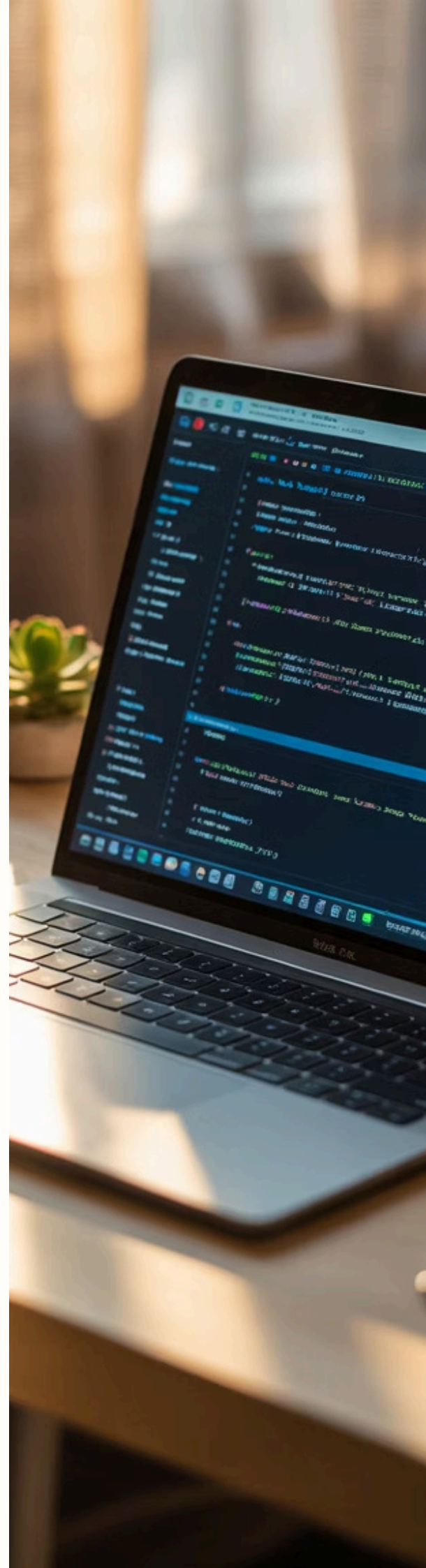
Mensagens de erro **claras e detalhadas**, stack traces organizados e comparações de valores esperados vs recebidos facilitam a identificação e correção de problemas durante o aprendizado.

Transformação do Aprendizado: Em vez de memorizar comandos de terminal e interpretar saídas textuais complexas, estudantes podem focar no que realmente importa: escrever código de qualidade e compreender os princípios de teste de software.

Esta abordagem visual torna o processo de aprendizado mais intuitivo e o desenvolvimento mais ágil, preparando estudantes para um workflow profissional moderno desde o início de sua jornada de aprendizado.

Próximos Passos


Agora que você domina os conceitos fundamentais do Test Explorer, é hora de colocar esse conhecimento em prática! A melhor forma de consolidar o aprendizado é através da experiência hands-on com as ferramentas e workflows apresentados neste tutorial.





Experimente Agora

Abra seu projeto no VS Code e localize o teste

`AddItem_Skulnexistente_DeveAdicionarNovoItem`. Execute-o diretamente pelo Test Explorer usando o botão  e observe o feedback visual em ação.

Configure Seu Ambiente

Implemente as configurações recomendadas no seu `settings.json` para otimizar a experiência com CodeLens e auto-descoberta de testes.

Pratique TDD

Crie um novo teste seguindo o ciclo Red-Green-Refactor, utilizando o Test Explorer para acompanhar visualmente cada etapa do processo de desenvolvimento.

Explore Recursos Avançados

Experimente o modo debug, filtros de teste e execução em lote para descobrir como essas funcionalidades podem acelerar seu desenvolvimento diário.



Próxima Meta: Execute o teste `AddItem_Skulnexistente_DeveAdicionarNovoItem` diretamente pelo Test Explorer e observe como a experiência visual transforma seu workflow de desenvolvimento! 🚀

Lembre-se: a maestria vem com a prática consistente. Utilize o Test Explorer em todos os seus projetos .NET e, em pouco tempo, essa ferramenta se tornará uma extensão natural do seu processo de desenvolvimento, aumentando significativamente sua produtividade e confiança no código que você escreve.