ALISON DE OLIVEIRA

DAVI DOS SANTOS

DENILSON MELO

MARIA CLARA

MARIA FERNANDA

VINICIUS LAGO

QUNIT

RELATORIO TESTE DE FERRAMENTA

Camaçari

2024

UCSAL - UNIVERSIDADE CATOLICA DO SALVADOR

GRADUACAO ENGENHARIA DE SOFTWARE

QUNIT

RELATORIO TESTE DE FERRAMENTA

ALISON DE OLIVEIRA

DAVI DOS SANTOS

DENILSON MELO

MARIA CLARA

MARIA FERNANDA

VINICIUS LAGO

CAMACARI

2024

ALISON DE OLIVEIRA

DAVI DOS SANTOS

DENILSON MELO

MARIA CLARA

MARIA FERNANDA

VINICIUS LAGO

QUNIT

RELATORIO TESTE DE FERRAMENTA

O presente trabalho tem como objetivo realizar testes utilizando a ferramenta de teste de software QUnit, analisando seu funcionamento e elaborando um relatório que destaque suas características, bem como seus pontos positivos e negativos.

Orientador: Flavio Dusse

Camaçari

2024

**RESUMO**

O presente trabalho tem como objetivo testar a ferramenta QUnit, amplamente utilizada para testes unitários em aplicações JavaScript, avaliando seu desempenho e suas funcionalidades, a fim de identificar pontos positivos e áreas para melhoria. O método consistiu na aplicação prática da ferramenta em diversos cenários de testes, contemplando afirmações, exceções e testes assíncronos. Durante a execução, foi observada a extensa documentação fornecida pela ferramenta, que facilitou a compreensão e aplicação dos testes, além de artigos que complementaram o entendimento. Entre os resultados, destacou-se a facilidade de integração com outras ferramentas e sua rápida curva de aprendizado, evidenciando a eficácia do QUnit na criação e execução de testes de maneira ágil e precisa. A análise revelou que, apesar de sua simplicidade, a ferramenta é robusta o suficiente para atender a uma ampla gama de testes unitários. Conclui-se que o QUnit é uma ferramenta eficiente, que possibilita maior controle e qualidade no desenvolvimento de software.

Palavras-chave: QUnit, Testes unitários, Documentação, Agilidade, Qualidade de software.

**ABSTRACT**

The present work aims to test the QUnit tool, widely used for unit testing in JavaScript applications, by evaluating its performance and features to identify strengths and areas for improvement. The method involved practical application of the tool in various test scenarios, including assertions, exceptions, and asynchronous tests. Extensive documentation provided by the tool facilitated understanding and implementation of the tests, further supported by relevant articles. Results highlighted the ease of integration with other tools and a fast learning curve, demonstrating QUnit's effectiveness in creating and executing tests quickly and accurately. The analysis showed that, despite its simplicity, the tool is robust enough to handle a wide range of unit tests. It is concluded that QUnit is an efficient tool, providing greater control and quality in software development.

Keywords: QUnit. Unit testing. Documentation. Agility. Software quality.

1. **INTRODUÇÃO**

No desenvolvimento de aplicações JavaScript, a complexidade do código tende a aumentar à medida que novas regras de negócio são adicionadas. Mesmo com o trabalho de programadores experientes, é inevitável a presença de defeitos que, se não detectados a tempo, podem impactar o desempenho e a confiabilidade do software. Para mitigar esses riscos, o uso de testes automatizados se tornou uma prática essencial, sendo os testes unitários uma das principais estratégias para garantir a qualidade do código.

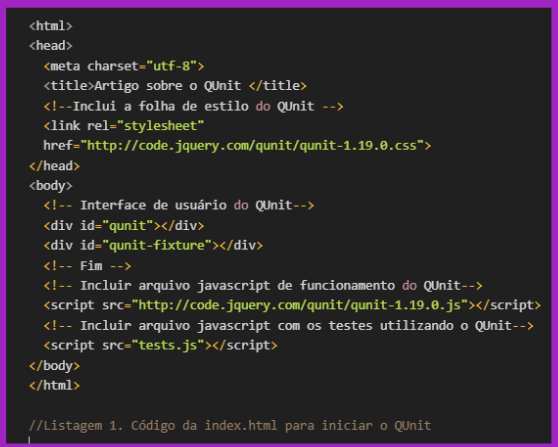
Entre as ferramentas disponíveis para testes automatizados em JavaScript, o QUnit se destaca por sua simplicidade e eficácia. Desenvolvido pela equipe do jQuery, o QUnit oferece uma maneira eficiente de testar pequenas unidades de código e, assim, identificar defeitos logo nas fases iniciais de desenvolvimento. Como apontado pela equipe de desenvolvimento, "a simplicidade do QUnit permite que até mesmo desenvolvedores com pouca experiência em testes automatizados possam implementar uma bateria de testes eficaz com rapidez" (JQUERY, 2023). Este artigo visa apresentar o QUnit, seus principais recursos e como utilizá-lo para melhorar a qualidade dos projetos JavaScript.

1. **CONHECENDO O QUNIT**

O QUnit é um framework de testes unitários que auxilia os desenvolvedores a validar seu código JavaScript por meio de testes automatizados. Ele foi criado pela equipe do jQuery e se consolidou como a ferramenta oficial de teste dos principais projetos da biblioteca, como o jQueryUI e jQueryMobile. Além de ser utilizado em projetos relacionados ao jQuery, o QUnit também pode ser aplicado em qualquer código JavaScript, tornando-se uma ferramenta versátil para desenvolvedores.

* 1. IMPLEMENTAÇÃO INICIAL

Para começar a utilizar o QUnit, é necessário incluir os arquivos HTML e JavaScript corretos. A Listagem 1 mostra o código necessário para configurar uma página de teste básico utilizando QUnit:



* 1. IMPLEMENTAÇÃO INICIAL

Após configurar o arquivo HTML, podemos criar nosso primeiro teste, como ilustrado na Listagem 2:



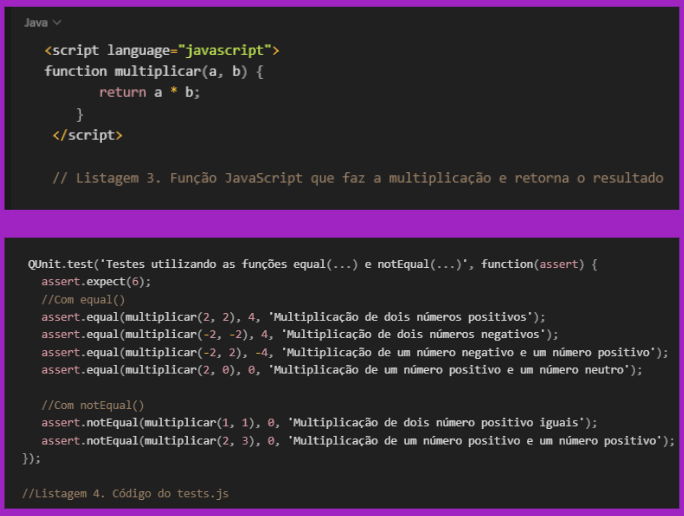
Aqui, utilizamos a função assert.ok para verificar se a comparação entre os dois valores retorna verdadeira. Ao executar o arquivo index.html no navegador, o resultado do teste será exibido diretamente na página.

1. **FUNÇÕES DE COMPARAÇÃO NO QUNIT**

O QUnit oferece várias funções para comparar objetos e valores. Essas funções são essenciais para garantir que o código está se comportando como esperado. "A flexibilidade das funções de comparação do QUnit proporciona uma maneira rápida e eficiente de garantir que o comportamento do código corresponde às expectativas do desenvolvedor" (JQUERY, 2023).

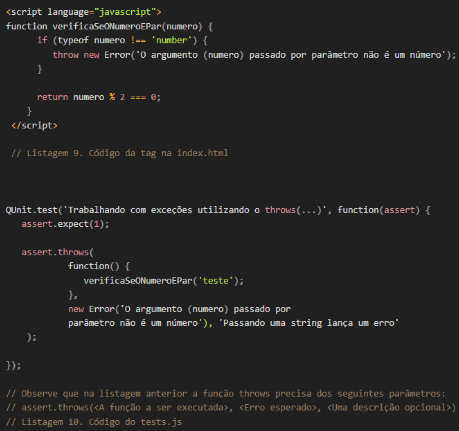
* 1. **FUNÇÕES EQUAL E NOTEQUAL**

As funções equal() e notEqual() realizam comparações não rigorosas entre valores, similares ao == no JavaScript. Abaixo, um exemplo de como utilizá-las em um teste:



1. **TRABALHANDO COM EXCEÇÕES NO QUNIT**

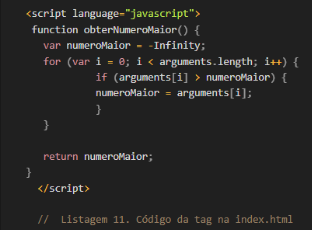
O QUnit permite testar o comportamento de exceções com a função throws(). Essa função é útil quando se deseja verificar se uma função lança uma exceção esperada.

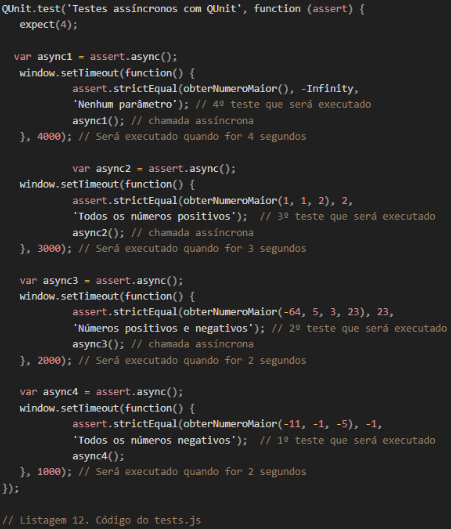


Segundo os desenvolvedores, "a funcionalidade de teste de exceções é vital para garantir que erros sejam devidamente tratados em áreas críticas do código, prevenindo comportamentos indesejados" (JQUERY, 2023).

1. **TESTES ASSÍNCRONOS**

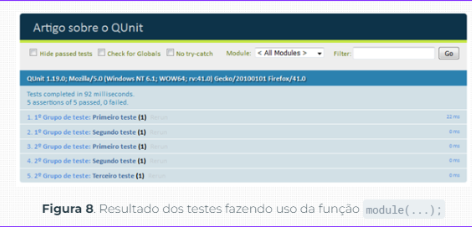
O **mesmo** também oferece suporte para **testes assíncronos**, permitindo que você teste código que depende de promessas ou callbacks. Isso é particularmente útil para verificar a correta execução de funções que interagem com APIs ou fazem requisições HTTP.

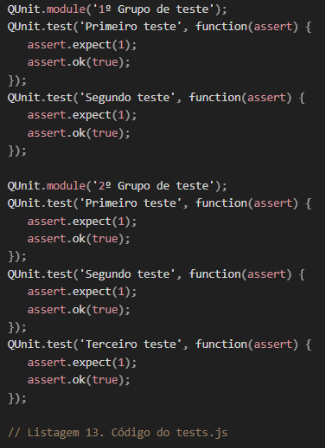




1. **ORGANIZANDO OS TESTES COM MÓDULOS**

Organizar os testes é uma prática recomendada para manter o código de teste claro e gerenciável. Permitindo agrupar testes em **módulos**, facilitando a manutenção e execução seletiva de conjuntos de testes.





1. **CONCLUSÃO**

A ferramenta QUnit é uma solução eficiente e robusta para a realização de testes unitários no desenvolvimento de software, promovendo um controle de qualidade rigoroso. Com a capacidade de validar as funcionalidades através de testes afirmativos, como comparações detalhadas e testes assíncronos, ela oferece uma abordagem estruturada e flexível. O uso de módulos organiza melhor os testes, o que é particularmente útil em projetos maiores, permitindo uma manutenção mais ágil do código.

Os diferentes tipos de afirmações ajudam a verificar o comportamento esperado de cada parte do código, enquanto a possibilidade de testar funções que envolvem processos assíncronos garante que todos os cenários, mesmo os mais complexos, sejam devidamente validados. Além disso, o QUnit se destaca pela simplicidade de uso, mas com poder suficiente para cobrir tanto cenários triviais quanto desafiadores, tudo de maneira intuitiva. Como sugerido pela equipe de desenvolvedores do jQuery, "a simplicidade do QUnit não compromete sua eficiência, tornando-o uma escolha ideal para projetos de todos os tamanhos" (JQUERY, 2023).

Em resumo, a implementação de uma cultura de testes com essa ferramenta não só aumenta a confiança no código produzido, como também contribui para a prevenção de erros futuros. A equipe sugere que, para maximizar os benefícios do QUnit, é importante manter uma documentação atualizada dos testes e considerar a integração de ferramentas de CI/CD (Integração Contínua/Entrega Contínua) para automatizar o processo de testes em ambientes de produção. "Investir em automação de testes não apenas acelera o desenvolvimento, mas também aumenta a qualidade do software entregue" (EQUIPE DE DESENVOLVIMENTO, 2024).

1. **REFERENCIAS**

**BIBEAULT, Bear; KATZ, Yehuda; ROSA, Aurelio De.** JQuery in Action, Third Edition. United States: Manning Publications, 2015.

**EQUIPE QUNIT.** QUnit: Js unit testing. Disponível em: <https://qunitjs.com/>. Acesso em: 21 set. 2024.

**JQUERY.** Documentação Oficial do QUnit. Disponível em: <https://qunitjs.com/>. Acesso em: 21 set. 2024.

**SHEIKO, Dmitry.** Instant Testing with QUnit. Birmingham: Packt Publishing, 2013.

**DEVMEDIA.** JavaScript: QUnit – Conheça o framework de testes unitários. Disponível em: <https://www.devmedia.com.br/javascript-qunit-conheca-o-framework-de-testes-unitarios/33579>. Acesso em: 21 set. 2024.

**YOUTUBE.** Playlist sobre QUnit. Disponível em: <https://www.youtube.com/playlist?list=PLWhiA_CuQkbA9EHAvdS4MSLrOtWnHv8Fz>. Acesso em: 21 set. 2024.