FACULDADE CATÓLICA SALESIANA DO ESPÍRITO SANTO

SILAS STOFFEL

ESTUDO COMPARATIVO ENTRE AS TECNOLOGIAS DE DESENVOLVIMENTO PARA WEB JAVASCRIPT E JQUERY FRAMEWORK

VITÓRIA

2013

SILAS STOFFEL

ESTUDO COMPARATIVO ENTRE AS TECNOLOGIAS DE DESENVOLVIMENTO PARA WEB JAVASCRIPT E JQUERY FRAMEWORK

Trabalho de Conclusão de Curso apresentado à Faculdade Católica Salesiana do Espírito Santo, como requisito obrigatório para obtenção de título de Bacharel em Sistemas de Informação.

Orientador: Prof. Msc. Rodrigo Alves Sarmento

VITÓRIA
2013

SILAS STOFFEL

ESTUDO COMPARATIVO ENTRE AS TECNOLOGIAS DE DESENVOLVIMENTO PARA WEB JAVASCRIPT E JQUERY FRAMEWORK

Trabalho de Conclusão de Curso apresentado à Faculdade Católica Salesiana do Espírito Santo, como requisito obrigatório para obtenção de título de Bacharel em Sistemas de Informação.

Aprovado em 03 de dezembro de 2013, por:

_____

Prof. MSc. Rodrigo Alves Sarmento, FCSES – Orientador

_____

Prof. MSc. André Cypriano Monteiro Costa, FCSES

_____

Prof. MSc. Romulo Ferreira Douro, FCSES

Dedico este trabalho aos meus pais que tanto me ajudaram diretamente e indiretamente na concretização da minha formação acadêmica.

AGRADECIMENTOS

Primeiramente agradeço a Deus por ter me proporcionado esta rica oportunidade de alcançar mais um objetivo em minha vida.

Agradeço a meu pai José Veríssimo, minha mãe Leida Stoffel e, minha tia Léia Inácio por participarem diretamente ou indiretamente na concretização do sonho.

Agradeço também a minha namorada Betânia Souza por me ajudar nas partes do trabalho que estavam ao seu alcance e tolerar minha ausência nos momentos de elaboração deste trabalho.

Agradeço ao meu orientador Rodrigo Sarmento pelos ensinamentos, paciência e dedicação que me permitiram concluir este trabalho.

Por fim agradeço às pessoas que colaboraram diretamente ou indiretamente na fase de pesquisa, sendo: Douglas Carlini, Luan Lima, Jeferson Soares, Carlos Barbosa, Clayton Cardoso, Welder Alvarenga, Thiago Bucsan, Elismar Lobão e todas outras que colaboram por meio destas. Agradeço também aos demais professores da instituição pelo conhecimento transmitido.

“A sabedoria é a coisa principal; adquire, pois,
a sabedoria; sim, com tudo o que possues
adquire o conhecimento”.

Provérbios, 4.7 - Bíblia Sagrada

RESUMO

A utilização dos recursos de tecnologia da informação (TI) para o desenvolvimento de páginas web proporciona a aceitação e ganho de interatividade na utilização de páginas. Partindo do ponto de interatividade das páginas web existem linguagens de scripts com tal finalidade. O JavaScript por exemplo, é uma das linguagens de scripts mais utilizadas para proporcionar interatividade, validações de formulário, detecção de objetos, cálculos matemáticos, fechamento e abertura de janelas, controle do comportamento do navegador, apresentação de mensagens, caixas de alerta, confirmação e uma série de outros recursos no desenvolvimento de páginas web. Mesmo o JavaScript sendo uma das linguagens de scripts mais utilizadas no desenvolvimento web, existem fatores complicadores para sua utilização como: incompatibilidades entre os navegadores, grande quantidade linhas de código escritas para alcançar simples resultados, codificação não uniforme para os principais navegadores, manutenção, tempo para adquirir aprendizado suficiente para desenvolver soluções e etc. Na tentativa de soluções destes problemas surgem os frameworks. Dentre os frameworks, o JQUERY destaca-se pela simplicidade e a grande quantidade de recursos. O presente trabalho, apresenta um estudo comparativo entre JavaScript nativo e o framework JQUERY. O estudo inicialmente resgatará informações baseando-se em literaturas e documentação oficial dos produtores das tecnologias. Com intuito de obter a referência de mercado foram aplicados e analisados questionários de pesquisa quantitativa e qualitativa realizada na região metropolitana da Grande Vitória, estado do Espírito Santo. O questionário coleta informações referente à utilização das tecnologias JavaScript e JQUERY no mercado de trabalho com foco nos indicadores de utilização dos recursos, performance, produtividade, curva de aprendizado, manutenibilidade e portabilidade. Ao final serão apresentados os resultados dos estudos realizados.

Palavras-chave: JavaScript. Framework. JQUERY.

ABSTRACT

The use of information technology's resources for the development of web pages provides the acceptation and the gain of interactivity in the use of pages. From the point of interactivity on the use of web pages there is scripting languages for such purpose. The JavaScript for example, is one of the most used scripting languages to provide interactivity, forms validation, objects detection, mathematical calculations, closing and opening of windows, control the behavior of the browser, display messages, alert boxes, confirmation and a number of other resources in the development of web pages. Although JavaScript being one of the most used languages on the development of web pages, there are complicating factors for its use such as: incompatibilities between browsers, large quantities of lines of code written to achieve simple results, no uniform coding for the major browsers, maintenance, time to acquire sufficient learning to develop solutions, etc. In an attempt to solutions of these problems arise frameworks. Among the frameworks, JQUERY is distinguished by its simplicity and the large amount of resources. This paper presents a comparative study between native JavaScript and JQUERY framework. The study initially retrieves information based on literature and official documentation of the technology producers. In order to get the market benchmark were applied questionnaires and analyzed quantitative and qualitative research conducted in the metropolitan region of Vitória, state of Espírito Santo. The questionnaire collects information regarding the use of JavaScript and JQUERY technologies in the labor market with a focus on indicators of resource utilization, performance, productivity, learning curve, maintainability and portability. At the end we will present the results of studies.

Keywords: JavaScript. Framework. JQUERY.

LISTA DE FIGURAS

Figura 1 - Exemplo da solicitação HTTP 29
Figura 2 - Funcionamento do protocolo.....30
Figura 3 - Processo de requisição ao servidor web.....32
Figura 4 - Incluindo o JQUERY em uma página HTML. 42
Figura 5 - Codificação básica em HTML para construção de simples página HTML 47
Figura 6 - Como é exibida a página da figura 6 por um navegador.....48
Figura 7 - Incluindo o JavaScript diretamente em página HTML.....49
Figura 8 - Chamada do arquivo do framework local.....51
Figura 9 - Fazendo chamada do arquivo do framework em servidor externo 51
Figura 10 - Parte da codificação do JQUERY Framework 1.10.2 – Versão não compactada.....53
Figura 11 - Codificação do JQUERY Framework 1.10.2 – Versão compactada. 53
Figura 12 - Medindo o tamanho em KB do JQUERY Framework 1.10.2 (Não Compactado) com a ferramenta de desenvolvedor do navegador Google Chrome 54
Figura 13 - Medindo o tamanho em KB do JQUERY Framework 1.10.2 (Compactado) com a ferramenta de desenvolvedor do navegador Google Chrome. 55
Figura 14 - Codificação utilizando JavaScript para mesurar produtividade. 57
Figura 15 - Codificação utilizando JQUERY para mesurar produtividade. 58
Figura 16 - Resultado padrão da solução codificada tanto com JavaScript e JQUERY 59
Figura 17 - Código base de comparativo de compatibilidade entre navegadores utilizando JavaScript 63
Figura 18 - Teste de manipulação de evento click no Internet Explorer 7 módulo de compatibilidade 64
Figura 19 - Teste de manipulação de evento click no Internet Explorer 9. 65
Figura 20 - Teste de manipulação de evento click no Mozilla Firefox. 65
Figura 21 - Código base de comparativo de compatibilidade entre navegadores com utilização do JQUERY. 66
Figura 22 - Informações do suporte do JQUERY aos navegadores.....67
Figura 23 - Página HTML para comparativo de recursos.....70

Figura 24 - Codificação com JavaScript para o comparativo de recursos 70
Figura 25 - Codificação com JQUERY para o comparativo de recursos 71

LISTA DE GRÁFICOS

Gráfico 1 - Avaliação de utilização dos recursos.....73
Gráfico 2 - Avaliação de performance. 74
Gráfico 3 - Avaliação de produtividade.....76
Gráfico 4 - Avaliação da curva de aprendizado. 77
Gráfico 5 - Avaliação de manutenibilidade 79
Gráfico 6 - Avaliação de portabilidade.....81
Gráfico 7 - Resultado da avaliação consolidada com todos indicadores.....85
Gráfico 8 - Resultado da avaliação final JavaScript e JQUERY.....86

LISTA DE SIGLAS

CSS - Cascading Style Sheet

HyTime - Hypermedia/Time-based Document Structuring Language

HTML - Hypertext Markup Language

HTTP - Hypertext Transfer Protocol

SGML - Standard Generalized Markup Language

XHTML - Extensible Hypertext Markup Language

XML - Extensible Markup Language

SUMÁRIO

1 INTRODUÇÃO 23

2 REFERÊNCIAL TEÓRICO 27

2.1 DEFINIÇÃO DE DESENVOLVIMENTO DE SISTEMAS 27

2.2 INTERNET, O QUE É? PARA QUE SERVE? 27

2.3 NAVEGADOR DE INTERNET 30

2.4 HTML 33

2.5 CSS 35

2.6 JAVASCRIPT 36

2.7 TECNOLOGIA CLIENTE SERVIDOR 38

2.8 FRAMEWORK 39

2.9 JQUERY 40

3 METODOLOGIA 43

4 RESULTADO E DISCUSSÃO DA PESQUISA 45

4.1 ESTUDO COMPARATIVO 45

4.1.1 Fundamentos 45

4.1.2 Iniciando Estruturação de Página com HTML Básico 45

4.1.3 Ambiente para Desenvolvimento em JavaScript 48

4.1.4 Iniciando Desenvolvimento com JavaScript Básico 49

4.1.5 Começando a Desenvolver com JQUERY 50

4.1.6 Comparativo de Performance 51

4.1.7 Comparativo de Produtividade 56

4.1.8 Comparativo de Curva de Aprendizado 59

4.1.9 Comparativo de Portabilidade 62

4.1.10 Comparativo de Manutenibilidade 67

4.1.11 Comparativo de Recursos 69

4.2 RESULTADO DA PESQUISA 71

4.2.1 Resultado da Avaliação de Utilização dos Recursos 73

4.2.2 Resultado da Avaliação de Performance 74

4.2.3 Resultado da Avaliação de Produtividade 76

4.2.4 Resultado Avaliação de Curva de Aprendizado 77

4.2.5 Resultado da Avaliação de Manutenibilidade 79

4.2.6 Resultado da Avaliação de Portabilidade 80

4.2.7 Resultado da Avaliação dos Pontos Fortes 82

4.2.8 Resultado da Avaliação dos Pontos Fracos 83

4.2.9 Consolidação dos Indicadores da Pesquisa 84

5 CONSIDERAÇÕES FINAIS 87

REFERÊNCIAS 90

APÊNDICE A – Questionário de Pesquisa 95

APÊNDICE B – Estrutura HTML da página para comparativo de produtividade 101

APÊNDICE C – Metodologia de confecção dos gráficos 102

1 INTRODUÇÃO

Mediante a rapidez da evolução das tecnologias, surge a Internet, com objetivo de interconectar milhares de dispositivos computacionais ao redor do mundo (KUROSE; ROSS, 2011).

Anteriormente à popularidade da Internet o objetivo era interligar redes militares de comunicação, sobre isto, Ramires; Cardoso citam Tanembaum (2009, p.1), afirma:

> A primeira Internet foi chamada de ARPANET e foi criada inicialmente com fins militares. Seu objetivo principal era formar uma arquitetura de rede robusta, que mesmo com a perda de algum equipamento, mantivesse seu funcionamento com os computadores restantes. Passados alguns anos, seu uso deixou de ser apenas militar e passou a conectar universidade e laboratórios.

A utilização da internet inicialmente estava limitada apenas às bases militares, universidades e instituições de pesquisa. Por fim, o governo decidiu permitir acesso à internet para propósito comercial (DEITEL; DEITEL; NIETO, 2003).

Com o passar do tempo a Internet passa ser usada no meio corporativo. A Internet rapidamente tem se tornado peça chave na infraestrutura do comércio eletrônico, pois oferece um meio mais fácil de comunicação entre empresas e indivíduos a um baixo custo. A Internet não é apenas usada no comércio eletrônico, também pode ser usada em serviços baseados em informação como software, música e vídeos podem ser divulgados fisicamente via Internet. Vendedores de outro tipo de produto ou serviço podem utilizar a internet para distribuir informações importantes para divulgação (LAUDON; LAUDON, 2004).

A melhor descrição para Internet é a informação. Por meio dela encontram-se serviços e facilidades, notícias e atualidades, ou se preferir como é o caso de muitas pessoas, um excelente local para encontrar amigos, jogos, bate papo e lojas virtuais (SILVA; VARGAS, 2007).

Além da infraestrutura para o funcionamento da Internet como servidores físicos, servidores DNS, domínios e etc., também é necessária a existência do serviço de distribuição de conteúdo na Internet. Este serviço baseia-se na criação de páginas que armazenam o conteúdo a ser distribuído na rede mundial de computadores. Uma das formas mais populares para a criação de páginas na internet é utilizando linguagens de marcação como HTML e estilização ou formatação como o CSS. O

HTML é uma linguagem de marcação utilizada para produzir páginas web (SILVA; VARGAS, 2007).

A criação de páginas construídas apenas com HTML e CSS não proporcionam recursos sofisticados de interatividade com o internauta nem com o navegador de internet. O termo internauta pode ser entendido como usuário de Internet ou da rede mundial de computadores que ocupa parte do seu tempo explorando os recursos por ela oferecidos (FERREIRA, 2000). Mediante a necessidade de interatividade surgiram as linguagens de scripts, dentre elas o JavaScript destaca-se pelos recursos por ela oferecidos.

O JavaScript é uma linguagem que tem a finalidade de fornecer um meio de adicionar interatividade às páginas de Internet, seu funcionamento é dependente das funcionalidades hospedadas no navegador do internauta devido ao fato de ser uma linguagem interpretada e fazer a utilização do interpretador instalado no navegador (SILVA, 2010). Os principais recursos do JavaScript são: ler e escrever elementos na página, realização de cálculos matemáticos, manipular elementos, gerenciar o tamanho da tela do usuário, executar ações condicionais, alertar usuário e manipulação de formulário entre outros (DUCKETT, 2010).

JavaScript não é uma linguagem simples para se programar. A programação em JavaScript requer esforços como: produção de inúmeras linhas codificas, etapas de testes e documentação. Diante do cenário apresentado surgiram os frameworks para codificação em JavaScript. O JQUERY, por exemplo, é um dos frameworks gratuitos mais utilizados no desenvolvimento web.

O framework JQUERY tem por objetivo a simplicidade e visa não submeter programadores ou designers a produzir grandes quantidades em linhas de código para alcançar simples efeitos (SILVA, 2008). Outro ponto positivo do JQUERY é a compatibilidade com os principais navegadores de mercado. Isso significa que ao desenvolver com JQUERY não será necessário escrever blocos de codificação específico para um navegador.

O presente trabalho tem o objetivo de elaborar um estudo comparativo do desenvolvimento de páginas para internet, onde serão comparados os fatores do desenvolvimento utilizando recursos nativos da linguagem JavaScript e o desenvolvimento com framework JQUERY.

Os quesitos abordados na elaboração do trabalho são:

- Estudar os conceitos e definições nas literaturas para melhor compreensão e entendimento sobre a Internet, funcionamento básico de aplicação web, conceitos de programação para o lado cliente e lado servidor, CSS, JavaScript, e o framework JQUERY.
- Descrever as principais características básicas do framework JQUERY.
- Fazer um estudo comparativo entre o JavaScript e o JQUERY, onde neste comparativo os indicadores abordados serão: recursos, performance, produtividade, curva de aprendizado, portabilidade e manutenção.
- Fazer uma pesquisa de campo para obter a referência de mercado quanto à utilização do JavaScript nativo e o JQUERY.

O trabalho foi confeccionado e dividido em cinco capítulos. No primeiro capítulo é realizada a apresentação do conteúdo introdutório do trabalho. O segundo capítulo apresenta base teórica, conceitos e definições que serão utilizados nos próximos capítulos. Todo o estudo dirigido neste capítulo foi embasado em documentação oficial e órgãos mantenedores das tecnologias utilizadas no trabalho, literaturas e artigos.

O terceiro capítulo apresenta informações referentes à maneira de como o estudo foi dirigido, elaborado e apresentado. O quarto capítulo, primeiramente apresenta um estudo comparativo entre as tecnologias JavaScript e o framework JQUERY baseado em referências bibliográficas. Na segunda parte do capítulo são apresentados resultados obtidos da pesquisa de campo realizada com objetivo de resgatar a referência de mercado das tecnologias comparadas neste trabalho.

O quinto e último capítulo, por sua vez apresenta os resultados, conclusões finais da pesquisa e proposta de continuação do estudo.

Este trabalho foi elaborado com intuito de fornecer um estudo teórico-prático a leitores leigos e profissionais de tecnologia com pouco ou nenhum conhecimento sobre a utilização das tecnologias JavaScript nativo e do framework JQUERY, apresentando resultados de avaliações comparativas realizadas utilizando fontes teóricas bibliográficas, práticas e estudos de mercado com profissionais utilizadores das tecnologias avaliadas.

Baseado nos resultados apresentados, o intuito é apresentar aos leitores uma base para que possa ser tomada decisão de qual tecnologia que pode ser usada nos projetos.

2 REFERÊNCIAL TEÓRICO

2.1 DEFINIÇÃO DE DESENVOLVIMENTO DE SISTEMAS

O desenvolvimento de sistemas na computação pode ser entendido como ato de elaboração e implantação de sistema em modelo computacional, isto é, transformação da necessidade de utilizador em um produto de software.

Um computador é uma máquina que executa operações. Um conjunto de operações forma um programa para o computador. O programa de computador é chamado software. Geralmente o software é desenvolvido por programadores que utilizam linguagens de programação para construí-lo Gesac (acesso em 8 abr. 2013). Conforme o 1º artigo da lei 9.609 de 19 de fevereiro de 1998, diz:

> Programa de computador é a expressão de um conjunto organizado de instruções em linguagem natural ou codificada, contida em suporte físico de qualquer natureza, de emprego necessário em máquinas automáticas de tratamento da informação, dispositivos, instrumentos ou equipamentos periféricos, baseados em técnica digital ou análoga, para fazê-los funcionar de modo e para fins determinados.

O processo de construção do sistema é constituído por etapas, uma das etapas envolvidas é a de análise do problema antes de propor uma solução computacional, logo após é projetada a solução, que é codificada computacionalmente e então são elaborados os planos e execução de testes para consequente entrega.

2.2 INTERNET, O QUE É? PARA QUE SERVE?

A internet nada mais é que uma rede mundial de computadores, que tem papel de interligar milhões de dispositivos ao redor do mundo. Sobre rede de computadores, Guizzo (2000, p.30) afirma:

> Uma rede de computadores é um conjunto de computadores interligados capazes de compartilhar informações e recursos. Essas redes são chamadas de redes locais, ou LANs (Local Área Networks).
>
> No mundo todo existem milhares de redes de computadores. Empresas, bancos, escolas, universidades, institutos de pesquisa, hospitais, órgão de governo, etc., a maioria deles tem sua própria rede. Mas antes da Internet, cada rede estava restrita aos limites da própria organização. Assim, por

> exemplo, dados armazenados num computador de uma universidade só podiam ser acessados no próprio computador ou através da rede dessa universidade. Pessoas de fora não tinham como acessá-lo. Essas redes existiam isoladas umas das outras. Ás vezes havia interligação entre universidades, ou entre empresas, de uma mesma cidade ou região. Formavam redes metropolitanas ou redes regionais. Com a Internet, todas essas redes se ligaram criando uma rede global. As redes locais não deixaram de existir, mas foram interligadas entre si, formando uma enorme teia. É por isso que a Internet não é apenas uma rede de computadores, mas uma rede de redes ou um emaranhado de redes.

Uma peça chave para o armazenamento e divulgação de informações por meio da Internet são os computadores definidos como servidor. Estes computadores são chamados de servidor de Internet ou servidor web. Sobre os tipos de computador servidor, o autor Guizzo (2000, p.32) diz:

> Na Internet, os computadores que disponibilizam informações, serviços e outras facilidades são chamados de servidores (eles “servem” dados). Os servidores são em geral, computadores poderosos, permanentemente ligados à Internet. Os usuários, por sua vez, rodam em seus computadores aplicações que requisitam dados dos servidores. Essas aplicações são chamados de clientes.

Para que seja distribuído conteúdo na Internet de forma interativa e dinâmica é necessário um protocolo. O protocolo utilizado é o HTTP que é conhecido com HypertText Transfer Protocol (Protocolo de Transferência de Hipertexto). Em outras palavras, é um método acordado para que sejam transferidos documentos baseados em hipertexto pela web. Enquanto que documento de hipertexto são páginas HTML, o protocolo também pode ser usado para transferir imagens ou qualquer outro arquivo que venha a ser necessário a uma página da web (FREEMAN; FREEMAN, 2008).

O acesso a uma página web na rede mundial de computadores pode ser feita pelo navegador web, basta apenas informar o endereço desejado na barra de endereço do navegador e aguardar que navegador se encarrega de exibir o retorno da solicitação HTTP.

Figura 1 - Exemplo da solicitação HTTP

Fonte: Freeman; Freeman (2008, p.115).

> A ideia por trás do hipertexto é que em vez de ler o texto em uma estrutura rígida e linear (como em um livro), como você pode passar facilmente de ponto a outro. Você poderá obter mais informações, voltar, ir para outros tópicos e navegar pelo texto com base nos que lhe interessa (LEMAY, 2002, p.6).

No mundo existem companhias de telecomunicações públicas e privadas que controlam as conexões de internet. As conexões administradas por estas companhias são de altíssima velocidade e interligam cidades, países e até continentes, por sua vez são conhecidos como backbones. As empresas conhecidas como provedores estão ligados aos backbones que oferecem o acesso à internet e empresas menores ou usuário domésticos contratam o acesso à internet dos provedores (GUIZZO, 2000).

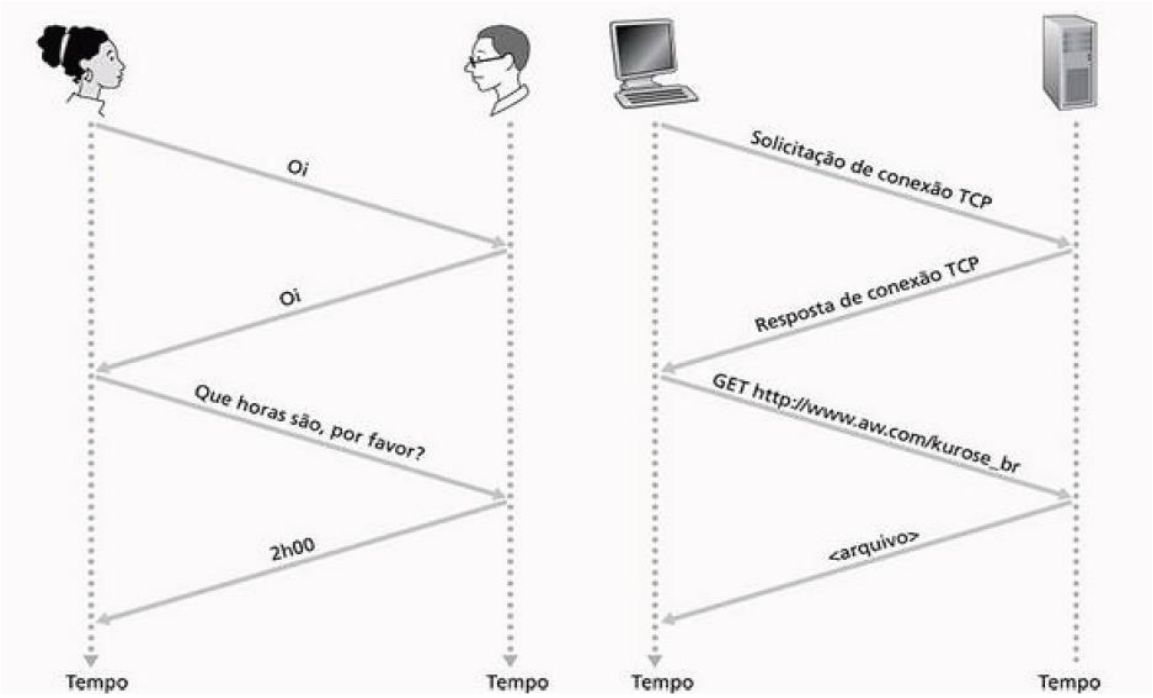
> As companhias de telecomunicações cobram desses provedores pelas suas conexões com os backbones, e os provedores por sua vez cobram das empresas e dos usuários finais pelo acesso. Assim a internet auto sustenta e os grandes custos dessa imensa rede são diluídos até o usuário final, que paga mensamente uma taxa (GUIZZO, 2000, p.36).

Portanto a internet serve para que você acesse computadores e se comunique com qualquer pessoa no mundo. Para o funcionamento de comunicação e troca de informações com outros computadores é necessária à utilização de um protocolo. Segundo (KUROSE; ROSS, 2011) pode-se entender protocolo com a seguinte analogia humana:

> Provavelmente é mais fácil entender a ideia de um protocolo de rede computadores considerando primeiramente algumas analogias humanas, já que executamos protocolo ao tempo todo. Considere o que você faz quando quer perguntar as horas a alguém.

> O protocolo humano (ou as boas maneiras, ao menos) ordena que, ao iniciarmos a comunicação com outras pessoas, primeiramente a cumprimentemos. A resposta comum para um “oi” é um outro “oi”. Implicitamente, tomamos a resposta cordial “oi” com a indicação que podemos prosseguir e perguntar as horas. Uma resposta diferente ao “oi” (tal como “Não me pertube”, “I don’t speak portuguese!“ ou alguma resposta impublicável) poderia indicar falta de vontade ou incapacidade de comunicação.

Figura 2 - Funcionamento do protocolo



Fonte: Kurose; Ross (2010, p.6)

Na computação, o funcionamento dos protocolos é semelhante à analogia citada. O que difere baseado na analogia é que a comunicação ou a troca de mensagens é feita por meio dos componentes de hardware e software de um determinado dispositivo. Os dispositivos podem ser PDA, computadores, celulares e outro equipamento habilitado para rede.

2.3 NAVEGADOR DE INTERNET

Para que as publicações na Internet sejam visualizadas pelos internautas é necessário um software fundamental que permite a visualização do conteúdo publicado. O navegador de internet ou browser ou web browser é um software que

permite acessar e visualizar o conteúdo das páginas de Internet. O termo navegador para Lemay (2002, p.11) pode ser compreendido da seguinte forma:

> [...] um programa que você usa para visualizar e navegar na World Wide Web. Os navegadores da web são chamados, algumas vezes de cliente da web ou em outros nomes da moda (ferramentas de navegação na internet), mas navegador da web é o termo, mas comumente utilizado.

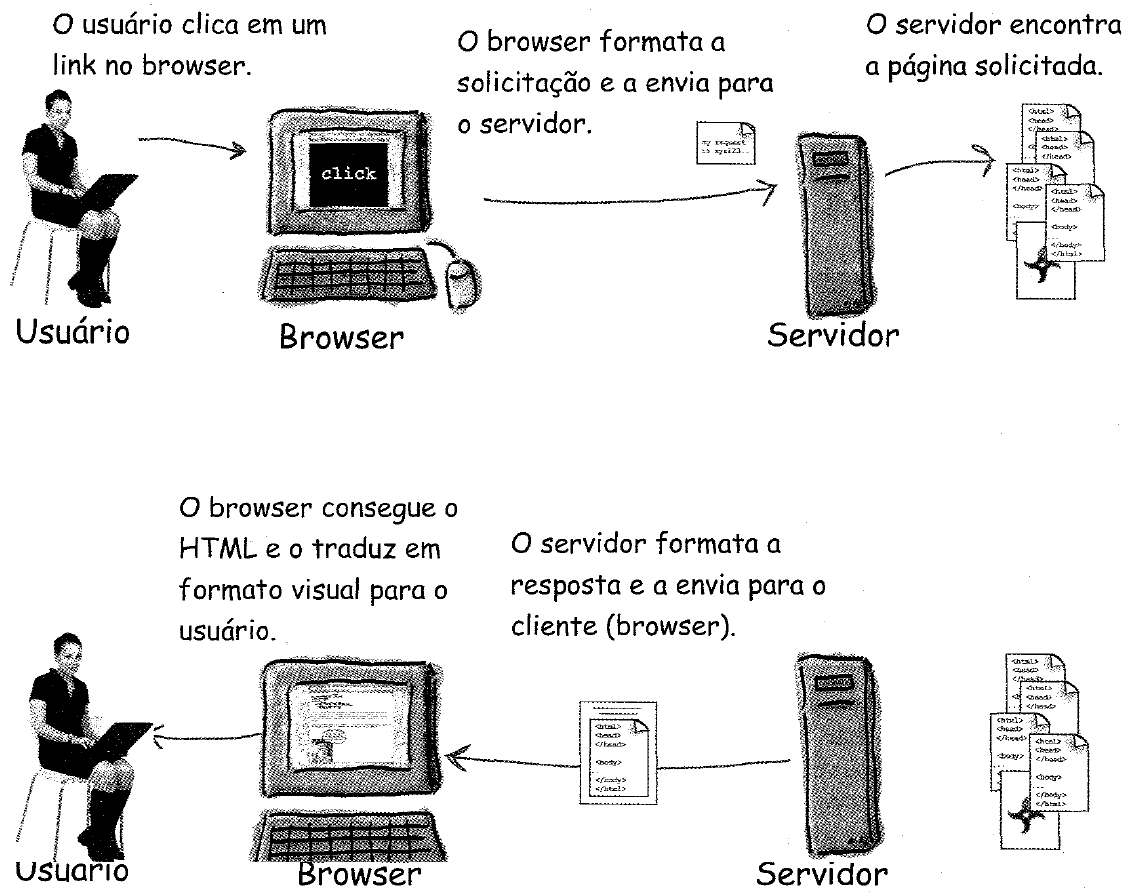
A principal função do navegador é permitir que o internauta acesse e visualize conteúdo de forma amigável e interpretável, sobre isto, os autores Freemann; Fremeean (2008, p.5) afirmam:

> Quando o browser lê seu HTML, ele interpreta todas as tags que envolvem seu texto. As tags são apenas palavras ou caracteres entre os sinais de maior e menor, como <head>, <p>, <h1> e assim por diante. As tags informam o browser a estrutura e o significado do seu texto. Assim, ao invés de apenas entulhar seu browser com um monte de texto, com o HTML você pode usar as tags para dizer qual texto é o título, qual texto é um parágrafo, qual texto precisa ser destacado e até mesmo onde as imagens devem ser colocadas.

Um navegador web também é conhecido como cliente, sobre isto, os autores Basham; Sierra; Bates (2007, p.5) afirmam:

> Quando falamos em clientes, no entanto, geralmente queremos dizer ao mesmo tempo o usuário e o browser (ou um dos dois).
>
> O browser é o software (como o Netscape ou o Mozilla) que sabe se comunicar com o servidor. A outra grande tarefa do browser é interpretar o código HTML e exibir a página ao usuário.
>
> Então, a partir de agora, quando usarmos o termo cliente, geralmente não nos preocuparemos se estamos falando do usuário ou do browser. Em outras palavras, o cliente é o browser fazendo aquilo que o usuário solicitou.

Figura 3 - Processo de requisição ao servidor web



Fonte: Basham; Sierra; Bates (2007, p.5)

Os navegadores mais conhecidos até o presente momento da elaboração deste trabalho são: Firefox, mantido pela fundação Mozilla; o Google Chrome, mantido pela Google; o Internet Explorer, mantido pela Microsoft; o Safari, mantido pela Apple e o Opera, mantido pela Opera Software ASA. Os navegadores Firefox, Google Chrome, Safari e Opera podem ser adquiridos sem nenhum custo. O navegador Internet Explorer está disponível apenas para o sistema operacional Microsoft Windows, no entanto, para utilizar o sistema operacional da Microsoft é necessário adquirir a licença de uso. Quando adquirido a licença do sistema operacional o navegador já está incluso no sistema operacional, em outras palavras, é necessário adquirir a licença do sistema operacional para utilizar o navegador.

2.4 HTML

Hypertext Markup Language conhecido como HTML é uma linguagem utilizada para criação de páginas web para serem visualizadas em um navegador. HTML surgiu em 1992, baseado/derivado do SGML, Standard Generalized Markup Language (Linguagem de Marcação Generalizada Padrão) e o HyTime (Hypermedia/Time-based Document Structuring Language) (MACEDO, 2004).

> A HTML, então, é uma linguagem de marcação que descreve a estrutura, o conteúdo e apresentação de um documento e sua relação com os outros documentos. Com ela podemos representar uma informação e vinculá-la a outros tipos de recursos, como texto, áudio vídeo, gráficos, entre outros, permitindo que diferentes tipos de informação sejam exibidos de forma simultânea e que esses recursos diferentes se complementem (MACEDO, 2004).

O HTML é um padrão internacional cujas especificações são mantidas pela World Wide Web Consortium também conhecido como W3C. A W3C fundada em 1994 é dedicada à criação de tecnologias interoperáveis não proprietárias. O objetivo principal da W3C é tornar a web universalmente conhecida e a padronização na criação e publicação de conteúdo web.

> O W3C é também uma organização de padronização. As tecnologias de padronizadas pelo W3C são chamadas de Recommendations incluem a extensible HyperText Markup Language (XHTML), Cascading Style Sheet (CSS), HyperText Markup Language (HTML; agora considerada uma tecnologia comum) e a Extensible Markup Language (XML) (DEITEL; DEITEL; NIETO, 2003, p.63)

O HTML é uma linguagem de marcação, estruturada por tags, estas tags são instruções de marcação apropriadas indicando como o navegador deve estruturar ou apresentar o documento. Na prática, tags, são caracteres que contêm sinais de maior e menor que delimitam o início e fim do conteúdo entre a tag declarada, tal como “texto desejado” que ao ser lido pelo navegador apenas apresenta o texto “texto desejado” em negrito (POWELL, 2010). Os comandos em HTML são chamados de tags (ou rótulos), são estes comandos que dizem ao navegador como o texto e as imagens serão formatadas e exibidas (SILVA; VARGAS, 2007).

> [...] a HTML geralmente é vista como nada mais do que uma linguagem de formatação de documento, ou marcação. As tags (dentro de caracteres delimitadores < >) instruem um programa visualizador (o browser ou, mais genericamente, o cliente) a exibir os pedaços de texto ou imagens (GOODMAN, 2001, p.4).

O HTML permite que os autores criem páginas web com títulos, parágrafos, textos, tabelas, listas, fotos, recuperem informações através de links de hipertexto, com o clique de um botão, realizem transações com serviços remotos, para uso na busca de informações, reservas, encomenda de produto, incluir folhas de difusão, videoclipes, clipes de som e outras aplicações diretamente em seus documentos.

Uma estrutura básica de página web é composta pelas seguintes tags “head” que armazena o conteúdo do cabeçalho e a tag “body” que define o corpo da página que será apresentado.

> O HTML define um conjunto de estilos em comum para as páginas web: cabeçalhos, parágrafos, listas e tabelas. Ele também define os estilos de caracteres como negrito e exemplos de código. Cada elemento possui um nome e está contido no que se convencionou chamar de tag (marca). Quando você escreve uma página da web no HTML, etiqueta os diferentes elementos da sua página com essas tags que dizem “este é um cabeçalho” ou este é um “item” (LEMAY, 2002, p.44).

HTML fornece apenas a maneira de descrever a estrutura de conteúdo das páginas, sua estrutura contém os recursos básicos para a formatação de textos como: definir parágrafos, tamanho e estilos como: negrito, sublinhado, cores entre outros. As páginas com os recursos mais avançados e agradáveis visualmente são escritas com o auxílio de CSS, que contém recursos de descrever como o conteúdo deve ser apresentado (FREEMAN; FREEMAN, 2008).

Após a criação e com a utilização dos recursos do HTML foram identificadas algumas limitações do HTML. Diante deste cenário surgiram novas linguagens e reformulações do HTML, o qual o objetivo era solucionar as limitações contidas no HTML. Dentre o processo de reformulação surgiram o XML e o XHTML. Macedo (2004, p.4), sobre o XML, afirma:

> A XML (eXtensible Markup Language) foi desenvolvida para solucionar limitações da HTML, ela é uma linguagem de marcação para descrição, captura, processamento e publicação de informações em diferentes tipos de mídias [...].

XHTML (Extensible HyperText Markup Language) é uma reformulação da HTML versão 4.01 baseada na linguagem XML. Neste processo de reformulação os recursos existentes na HTML 4.01 não deixaram existir, o que significa que XHTML não tornou os recursos da HTML 4.01 obsoletos (MACEDO, 2004).

XHTML na verdade é XML, e possui vantagens interessantes comparadas ao HTML 4.01. Uma das primeiras mudanças foi na mudança de sintaxe que está mais rigorosa. Também o XHTML trouxe benefícios, tais como:

- Permite que os leitores de tela e outros browsers para deficientes visuais consumam com mais facilidades os conteúdo da Web.
- Aproveitamento de páginas em todos os últimos browsers que estão por vir e possibilidade funcionarem em dispositivos móveis.
- Possibilidade de ser estendido para incluir novas marcações.
- Possibilidade de ser lido por aplicações de software que fazem leitura de XML.

Mesmo sendo semelhante ao HTML, o XHTML possui algumas diferenças e benefícios, conforme citados acima (FREEMAN; FREEMAN, 2008, p.232).

2.5 CSS

O Cascading Style Sheets, também conhecido como CSS, em tradução, folha de estilo em cascata, é uma linguagem independente do HTML e pode ser usado em qualquer linguagem de marcação que seja baseado em XML. Os autores Pfaffenberger; Scharfer; White; Bill Karow (2004) definem que “o CSS foi projetado para auxiliar na formatação fora de especificação HTML”.

O CSS serve para descrever a aparência das páginas web, o CSS permite a estilização de apresentação através dos recursos de cores, layouts e fontes, permite adaptar a apresentação em diferentes tipos de dispositivos, como grandes telas, pequenas telas e apresentação para serem impressas. Sobre os recursos do CSS, Macedo (2004, p.6) afirma:

> [...] permite uma maior versatilidade no desenvolvimento de design para sites sem aumentar o seu tamanho. Basicamente, o CSS permite ao designer um controle maior sobre os atributos tipográficos de um site, como tamanho e cor das fontes, espaçamento entre linhas e caracteres, margem de texto, entre outros. Introduziu também a utilização de layers, permitindo a sobreposição de texto sobre outro texto ou texto sobre imagens.

2.6 JAVASCRIPT

O JavaScript, também chamado de JS ou linguagem de scripts da web. O JavaScript é uma linguagem de programação interpretada, orientada a objetos e baseada em protótipos. Mesmo com sua popularidade, sendo conhecida como a linguagem de scripts da web ela também pode ser usada em outros ambientes como o Couchbase e Node JS (MOZILLA DEVELOPER NETWORK, acesso em 17 abr. 2013).

Inicialmente, o JavaScript foi lançado juntamente com o navegador Netscape Navigator 2.0. Os autores Pereira; Rehder (2003, p.94) afirmam: “[...] O primeiro browser a suportar JavaScript foi o Netscape Navigator 2.0 [...]”. Anteriormente ao lançamento do JavaScript a navegação web não era interativa, caso houvesse necessidade de desenvolver algo interativo a maneira existente era criar a interatividade diretamente na programação usando scripts CGI. Segundo (LEMAY, 2002, p.568), “[...] não havia como adicionar conteúdo interativo às páginas da Web. Se quiséssemos criar um aplicativo interativo, tínhamos que fazer tudo usando scripts CGI ou outras técnicas de programação lado servidor [...]”. Com a linguagem JavaScript é possível criar funcionalidades que executam no computador do internauta melhorando a interatividade. Goodman (2001, p.3) define:

> A linguagem JavaScript é uma tecnologia para melhoria web. Quando empregada no computador cliente, a linguagem pode ajudar a transformar uma página de conteúdo estático em uma experiência atraente, interativa e inteligente. As aplicações podem ser tão sutis quanto receber o visitante de um site com saudação “Bom dia!” quando for de manhã no fuso horário do computador cliente – embora seja a hora do jantar onde o servidor está localizado. Ou então as aplicações podem ser muito mais óbvias, como ao oferecer um conteúdo de um show de slides em um download de uma página enquanto o JavaScript controla a sequência de ocultar, exibir e “flutuar o slide” enquanto o navega pela apresentação.

O paradigma de programação com orientação a objetos está presente no JavaScript. Orientação a objetos não é apenas um recurso exclusivo do JavaScript, o paradigma de orientação a objeto surgiu e já era usado desde a década de 1960. Os autores Melo; Nascimento (2008, p.98) afirmam: “[...] Desde a década de 1960 já existiram linguagem que davam suporte a essa metodologia; contudo essas linguagens bem como a metodologia de orientação a objetos eram geralmente restritas ao meio acadêmico [...]”. Com o passar do tempo vendo as evoluções das formas de

linguagem de programação muitas das linguagens começaram a evoluir criando os recursos de programação orientada a objetos.

> A orientação a objetos é um paradigma que representa toda filosofia para construção de sistemas. Em vez de construir um sistema formado por um conjunto de procedimentos e variáveis nem sempre agrupadas de acordo com o contexto, como se fazia em linguagens estruturadas (Cobol, Clipper, Pascal), na orientação a objetos utilizamos a ótica mais próxima do mundo real. Lidamos com objetos, estruturas que já conhecemos do nosso dia-a-dia e sobre as quais possuímos maior compreensão (DALL'OGLIO, 2007, p.87).

O JavaScript é uma linguagem de programação interpretada e utiliza o interpretador acoplado nos navegadores. Um programa que faz utilização de um interpretador ou compilador difere-se na maneira de execução. Schild (1996, p.9), define os termos interpretadores e compiladores podem ser entendidos como:

> Os termos compiladores e interpretadores referem-se à maneira como um programa é executado. Em teoria, qualquer linguagem de programação pode ser compilada ou interpretada, mas algumas linguagens são executadas de uma maneira ou de outra.

Interpretadores e compiladores, nada mais são que programas com sofisticação que operam diretamente no código fonte escritos em uma determinada linguagem de programação. Definindo a função de interpretador de um código fonte seria: o interpretador faz a leitura de linha por linha do programa escrito em uma linguagem de programação e executa a instrução definida na linha, o que acarreta atrasos no processo de conclusão do programa, diferentemente de uma linguagem compilada que faz a leitura execução de todas as linhas em uma leitura apenas e gera código de máquina, com isso o programa não precisa ser recompilado para usá-lo novamente. Já uma linguagem interpretada precisa do interpretador todas as vezes que o programa precisar ser executado (SCHILD, 1996).

Outra característica do JavaScript é que uma linguagem que é executada no lado cliente, ou seja, não é preciso necessariamente de um servidor web para o seu funcionamento, no caso do JavaScript é preciso de um interpretador para o funcionamento, e este interpretador já está embutido no próprio navegador. McFarland (2011, p.23) afirma que: "[...] JavaScript is a client-side language, which (in English) means that it works inside a web browser [...]". O JavaScript é uma linguagem cliente, isso significa que ele funciona dentro do navegador (MCFARLAND, 2011).

Toda vez que é executado um comando não necessariamente será disparado uma requisição no servidor web para executar e obter o retorno da execução, em programa onde o requisito funcional é clicar em botão e esperar uma mensagem identificando que foi clicado naquele botão específico, então este requisito pode ser atendido com o JavaScript através do evento onclick sem enviar a requisição para que o servidor processe e de a resposta. Os autores Deitel; Deitel; Nieto (2003, p.648) completa sobre scripts do lado cliente, da seguinte forma:

> A criação de scripts do lado cliente permite validar a entrada do usuário, acessar o navegador e aprimorar as páginas web com controle ActiveX, Dynamic HTML e os miniaplicativos Java (isto é, programas em Java do lado cliente que rodam em navegadores). A validação do lado cliente reduz o número de solicitações que precisam ser passadas ao servidor. A interatividade permite aos usuários tomar decisões, clicar em botões, brincar com jogos, etc. – tornando mais interessante a experiência de navegar em um site Web.

2.7 TECNOLOGIA CLIENTE SERVIDOR

A tecnologia cliente/servidor possui uma arquitetura na qual o processamento da informação é distribuído em módulos ou processos distintos. Existe um processo que é responsável pela manutenção da informação (servidores) e outros responsáveis pela obtenção dos dados (os clientes). Os processos cliente fazem requisição ou enviam pedidos para o processo servidor, e este por sua vez processa e envia os resultados dos pedidos (TAROUCO, 2013).

No desenvolvimento de páginas web as tecnologias do lado cliente são incorporadas ao HTML.

> As adições mais comuns à HTML simples estão no lado cliente. Esses suplementos incluem extensões de formatação como Cascading Style Sheets e Dymamic HTML; Linguagem de script do lado cliente como JavaScript; Applets Java e Flash. O suporte para todas essas tecnologias é ou não dependendo do caso, integrado ao navegador (CONVERSE; PARK, 2003, p.18).

Quadro 1 - Definições das principais tecnologias do lado cliente

Tecnologia Lado Cliente	Utilização Principal	Efeitos de exemplo
Cascading Style Sheets, Dynamic HTML.	Formatar páginas: Controlar tamanho, layout e sincronização de elementos	Sobrepor fontes diferentes cores/tamanho. Camadas, posicionamento exato
Script do lado cliente (JavaScript, VBScript)	Tratamento de evento: consequência de eventos definidos	Link que muda a cor no mouseover. Calculadora de hipoteca
Applets Java	Distribuir pequenas aplicações independentes	Logotipo móvel Palavras cruzadas
Animações em Flash	Animação	Desenhos animados de curta metragem

Fonte: Converse; Park (2003).

As tecnologias do lado cliente na internet também tem seu lado ruim na história, que é a inteira dependência do navegador web. Devido esta dependência, os usuários podem habilitar e desabilitar configurações no navegador que causam consequências desagradáveis. Um exemplo, seria o ato de desabilitar o JavaScript que impede o funcionamento/visualização normal da página web. Outro ponto negativo é que através do recurso de visualização de código fonte contido no navegador o internauta tem acesso aos códigos fontes (CONVERSE; PARK, 2003).

Em contraposição às tecnologias lado cliente, que é parte mais atraente e fascinante, existe a tecnologia para o lado do servidor. No desenvolvimento de páginas web as tecnologias para lado servidor ou também conhecido como scripts lado servidor, são os responsáveis por tratar conexões das páginas aos servidores back-end, como os bancos de dados. Um exemplo comum que se pode demostrar a interação entre cliente e servidor são os formulários on-line e algumas listas suspensas (normalmente aquelas que exigem que se clique em um botão) que o script monta dinamicamente no servidor (CONVERSE; PARK, 2003).

2.8 FRAMEWORK

O termo framework é muito comentado e utilizado por empresas, desenvolvedores de sistemas e os profissionais de tecnologia da informação. Primeiramente, framework é uma palavra que possui vários significados e ainda refere a algo que

não é novo, em um contexto de desenvolvimento de sistemas, um framework, pode ser compreendido como sub pacotes ou subsistemas que agilizam nas atividades relacionadas ao desenvolvimento ou apenas parte delas. Os autores Melo; Nascimento (2007, p.236) afirmam que:

> Framework nada mais é que um tipo especial de aplicativo, que oferece um conjunto básico de ferramentas e subsistemas que automatizam grande parte dos serviços que precisam ser implantados nos sistemas para usuários finais. Alguns exemplos práticos são: um cadastro de clientes, um blog na internet, um site de notícia etc.

Segundo Lisboa, citado por Larman (2009, p.21) framework pode ser compreendido como: “[...] um conjunto coeso de interfaces e classes que colaboram para fornecer serviços para parte básica e constante de um sistema lógico [...]”.

Muitos desenvolvedores associam um framework com uma biblioteca de classes. Essas classes fornecem o serviço básico e logico para um aplicativo ou sistema (LISBOA, 2009).

A ideia de adotar um framework para o desenvolvimento de codificação de software é que a equipe de desenvolvimento não precise se preocupar em iniciar o projeto do ponto zero, pois o framework fornece uma infraestrutura completa para sustentação do sistema. Com isso, a equipe pode dedicar maior parte do tempo de trabalho nas regras de negócios (MELO; NASCIMENTO, 2007)

2.9 JQUERY

A palavra ideal que defini o framework JQUERY é simplicidade nas atividades de codificação em JavaScript. JQUERY é um framework conhecido por muitos como uma biblioteca de JavaScript. Seu código é livre e aberto, isto resumidamente seria que o framework pode ser obtido gratuitamente e ser utilizado para fazer o uso em projetos pessoais ou comerciais (SILVA, 2008).

O JQUERY foi criado por John Resig. Seu trabalho no momento de criação do framework foi simplificar a codificação em JavaScript para os utilizadores e com diferenciais que é permitir tanto designers e programadores com poucos conhecimentos em JavaScripts utilizarem o framework (SILVA, 2008).

O site corporativo do framework JQUERY afirma que:

> JQUERY is a fast, small, and feature-rich JavaScript library. It makes things like HTML document traversal and manipulation, event handling, animation, and Ajax much simpler with an easy-to-use API that works across a multitude of browsers. With a combination of versatility and extensibility, JQUERY has changed the way that millions of people write JavaScript.

Em tradução da definição exposta pelo site corporativo do framework JQUERY, o framework é para codificação em JavaScript e possui características como: rapidez, é pequeno e rico em recursos. Ele faz coisas como geração de conteúdo HTML em páginas, manipulação de eventos, animação e AJAX muito mais simples com uma API fácil de usar, que funcionam em vários de navegadores. Possui combinação de versatilidade e capacidade de expansão, JQUERY mudou a maneira que milhões de pessoas escrevem JavaScript.

Sobre definição e as características do framework JQUERY, o autor McFarland (2011, p.3) afirma:

> That's where JQUERY comes in. JQUERY is a JavaScript library intended to make JavaScript programming easier and more fun. A JavaScript library is a complex JavaScript program that both simplifies difficult tasks and solves cross-browser problems. In other words, JQUERY solves the two biggest headaches with JavaScript — complexity and the finicky nature of different web browsers.
>
> JQUERY is a web designer's secret weapon in the battle of JavaScript programming. With JQUERY, you can accomplish tasks in a single line of code that would otherwise take hundreds of lines of programming and many hours of browser testing to achieve with your own JavaScript code […].

JQUERY, além pregar a simplicidade para codificação em JavaScript, também fornece recursos de interatividade e dinamismo as páginas web. Segundo Silva (2008, p.22) o framework JQUERY serve para:

> Adicionar efeitos visuais e animações;
>
> Acessar e manipular DOM;
>
> Buscar informações no servidor sem necessidade de recarregar a página;
>
> Prover interatividade;
>
> Alterar conteúdos;
>
> Simplificar apresentação e estilização;
>
> Simplificar tarefas específicas de JavaScript;
>
> Realizar outras tarefas às descritas.

Dentre a diversidade de recursos fornecidos pelo JQUERY, um dos interessantes é a conformidade com os padrões web definidos pela W3C, quando se fala em conformidade em web significa o funcionamento em qualquer sistema operacional e navegador. O JQUERY também fornece suporte completo à terceira versão CSS.

Outra coisa que vale lembrar mesmo o JQUERY fornecendo suporte aos padrões web não significa que usando JQUERY a aplicação fica inteiramente dentro dos padrões web definidos pela W3C, cabe ao desenvolvedor entender e aplicar as técnicas das outras linguagens e tecnologias usadas na aplicação.

As principais características do framework JQUERY são:

- Utiliza recursos para selecionar e/ou localizar elementos por seletores CSS nas páginas HTML.
- Foi escrito de maneira que permite compatibilidade com instalação ou integração de plug-ins e extensões.
- Possui suporte para os principais navegadores.
- Silva (2008, p.29) “[...] é capaz de interação implícita, isto é, não há necessidade de construção de loops para localização de elementos no documento “[...]”.
- Possui uma estrutura que permite a modalidade de programação encadeada, ou seja, cada método retorna um objeto.
- É extensível, permite que a criação e inclusão de funcionalidades na biblioteca existente.

Os passos para começar a desenvolver com JQUERY são simples e objetivos. Basta apenas acessar o site do produto, que é http://www.jquery.com, procurar pelo link download. Até o desenvolvimento deste trabalho o framework JQUERY está codificado dentro de um único arquivo. Após obtido uma cópia do framework basta apenas fazer a chamada na página para começar a utilizá-lo. O exemplo de como fazer a chamada do arquivo pode ser visto na figura 4.

Figura 4 - Incluindo o JQUERY em uma página HTML.

```
...
<head>
...
<script type="text/javascript" src="/caminho/jquery-1.2.6.js"></script> <!-- esta
   linha chama a biblioteca jQuery -->
</head>
...
```

O valor do atributo `src` indica o caminho (diretório) no qual se encontra gravado o arquivo da biblioteca.

Fonte: Silva (2008, p.30).

3 METODOLOGIA

Visando obter as respostas aos problemas e colaboração com o conhecimento científico, usa-se a pesquisa. A pesquisa para Yin (2001): “[...] a pesquisa deve ser conduzida por um projeto de pesquisa que visa conectar os dados empíricos às questões iniciais de estudo através de uma sequência lógica que permitirá chegar às conclusões. [...]”.

Para alcançar os objetivos geral e específico deste trabalho, o método de pesquisa escolhido foi pesquisa exploratória. Sobre a pesquisa exploratória, (MARCONI; LAKATOS, 2003, p.188) afirmam que:

> [...] são investigações de pesquisa empírica cujo objetivo é a formulação de questões ou de um problema, com tripla finalidade: desenvolver hipóteses, aumentar a familiaridade do pesquisador com um ambiente, fato ou fenômeno, para a realização de uma pesquisa futura mais precisa ou modificar e clarificar conceitos.

Na primeira parte deste trabalho foi realizada uma revisão bibliografia dentre as diversas literaturas existentes, publicações de artigos científicos, sites dos produtores ou mantenedores da tecnologia, manuais e documentações das tecnologias abordadas, criando assim um embasamento teórico (MORESI, 2003).

> A investigação exploratória é realizada em área na qual há pouco conhecimento acumulado e sistematizado. Por sua natureza de sondagem, não comporta hipóteses que, todavia, poderão surgir durante ou ao final da pesquisa. É, normalmente, o primeiro passo para quem não conhece suficientemente o campo que pretende abordar (MORESI, 2003, p.9).

Depois de formado um embasamento teórico, a segunda parte do trabalho será realizado um estudo comparativo entre o desenvolvimento com JavaScript nativo e com o framework JQUERY.

Para poder realizar o estudo comparativo da utilização das tecnologias e obter a referência de mercado será aplicada uma pesquisa de campo através de um questionário (APÊNDICE A) de caráter quantitativo e qualitativo. Este questionário será aplicado a um grupo de pessoas que utilizam as tecnologias comparadas neste trabalho. O questionário foi aplicado às pessoas que assumem no mínimo um cargo em empresas de qualquer porte localizadas na região metropolitana de Vitória, estado de Espirito Santo.

Os resultados obtidos na pesquisa serão apresentados em formato de quadros ou gráficos. A metodologia utilizada para confeccionar os gráficos está documentada no APÊNDICE C. Em determinadas partes do estudo serão apresentados pequenos trechos de codificação de linguagem de programação como JavaScript e linguagem de marcação como HTML.

Os itens necessários para concretização do estudo comparativo utilizados foram:

- Netbeans versão 7.3.1, para codificação utilizando as linguagens JavaScript, HTML e CSS.
- Internet Explorer versão 9 com modo de compatibilidade para versões 7 e 8, Mozilla Firefox versão 24 e Google Chrome versão 29.
- Microsoft Office Excel, utilizado para criar os gráficos com a interpretação dos dados do questionário de pesquisa aplicado.
- Cópia do código fonte do framework JQUERY versão 1.10.2.

4 RESULTADO E DISCUSSÃO DA PESQUISA

4.1 ESTUDO COMPARATIVO

Esta seção abordará o estudo comparativo entre as linguagens e tecnologias definidas no objetivo deste trabalho.

4.1.1 Fundamentos

O presente trabalho utilizará as seguintes linguagens:

- HTML, para estruturação de páginas, os recursos da HTML usados serão básicos devido o foco do trabalho não ser HTML.
- CSS básico para formatação de conteúdo.
- JavaScript, que é a linguagem foco do trabalho.
- Framework JQUERY versão 1.10.2.

Todas as codificações desenvolvidas neste trabalho serão testadas nos seguintes navegadores:

- Mozilla Firefox, versão 24.0.
- Google Chrome, versão 30.0.1599.69 m.
- Internet Explorer, versão 9. Nos testes de portabilidade será usado o modo de compatibilidade que já está incluído nativamente no navegador, tais testes em modo de compatibilidade são para poder simular ambientes com versões anteriores do navegador Internet Explorer.

4.1.2 Iniciando Estruturação de Página com HTML Básico

A Linguagem HTML é uma linguagem de marcação que descreve a estrutura de uma página web, o conteúdo e a apresentação de um documento e sua relação com os outros documentos (MACEDO, 2004).

Visando alcançar os objetivos específicos deste trabalho, será utilizado a linguagem HTML para estruturação de páginas e incorporar componentes JavaScript e CSS. Inicialmente, serão abordados os recursos básicos para estruturação de páginas com a Linguagem HTML.

Quando for utilizar a linguagem de marcação HTML, a primeira coisa que deverá ser codificada são as informações de que tipo de linguagem de marcação será utilizado para exibição do conteúdo (MACEDO, 2004). Na maioria das páginas existentes é utilizada a seguinte declaração: “<DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">”, adicionado esta linha no documento fica definido que será utilizado XHTML com tipo de documento Transitional, o que significa que podem ser utilizados recursos do HTML que estão classificados como obsoletos na nova versão, mas ainda são uteis por questão de compatibilidade se comparando com versões anteriores (MACEDO, 2004).

O próximo passo é definir a tag “<html></html>” que representa o início e fim de um documento HTML. Quando o documento for usar o XHTML na declaração deverá ser adicionado um atributo conforme mostrado a seguir: “<html xmlns="http://www.w3.org/1999/xhtml"></html>” (MACEDO, 2004).

Depois de incluir a tag “<html></html>” a próxima tag a ser incluída é a “<head></head>” que armazena um conjunto de tags com informações do cabeçalho de uma página. A utilização da tag pode ser exemplificada da seguinte maneira: “<head><title>Título da página</title></head>”. Neste exemplo dentro da tag “<head></head>” foi adicionada outra tag, sendo “<title></title>” que define o título da página.

Para finalizar a estrutura básica de uma página HTML a nova tag a ser adicionada é a tag “<body></body>”, que é o corpo, ou melhor, dizendo tudo que será apresentado no browser para o internauta. Dentro da tag “<body></body>” podem ser adicionadas outras tags ou apenas texto.

Figura 5 - Codificação básica em HTML para construção de simples página HTML

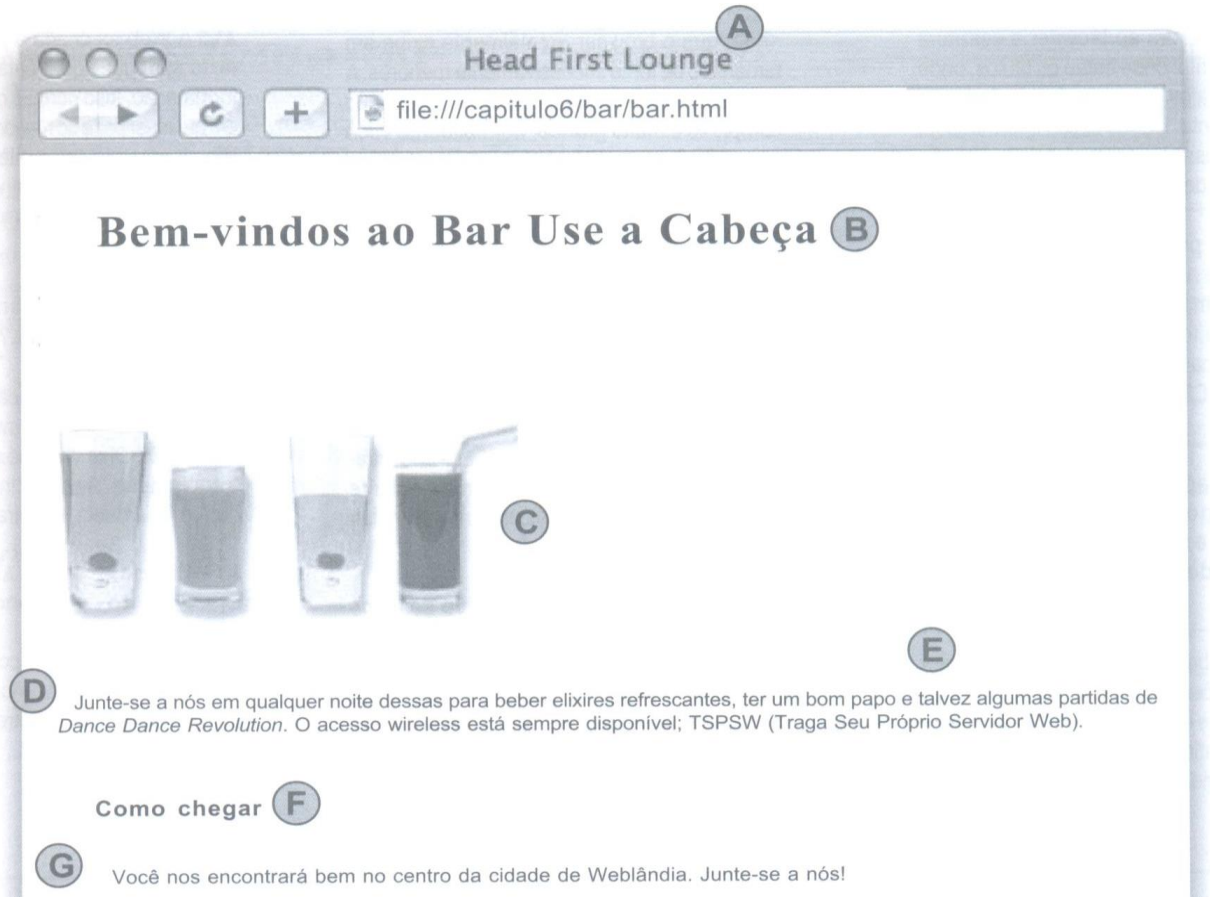
```
<html>
  <head>
    <title>Head First Lounge</title>
  </head>
  <body>
    <h1>Bem-vindos ao Bar Use a Cabeça</h1>
    <img src="imagens/drinks.gif" alt="Drinques" />
    <p>
      Junte-se a nós qualquer noite dessas para beber<a ref="bebidas/
      elixir.html">elixires</a> refrescantes, ter um bom papo
      e talvez algumas partidas de <em>Dance Dance Revolution</em>.
      O acesso wireless está sempre disponível;
      TSPSW(Traga Seu Próprio Servidor Web).
    </p>
    <h2>Como chegar</h2>
    <p>
      Você nos encontrará bem no centro da cidade
      de Weblândia. Junte-se a nós!
    </p>
  </body>
</html>
```

Fonte: Freeman; Freeman (2008, p.5)

A figura 5 ilustra a codificação utilizada para gerar uma página utilizando a linguagem HTML. A imagem foi mapeada por letras do alfabeto iniciando em A e finalizando em G, seguindo uma ordem crescente. Cada uma das letras em destaque (círculo para a respectiva letra) representa uma tag ou conjunto delas.

A figura 6 ilustra como o mapeamento da figura 5 é exibido por um navegador de internet. Cada ponto mapeado na figura 5 é representado pelo mesmo mapeamento na figura 6.

Figura 6 - Como é exibida a página da figura 6 por um navegador



Fonte: Freeman; Freeman (2008, p.5)

4.1.3 Ambiente para Desenvolvimento em JavaScript

Os componentes básicos necessários para o desenvolvimento utilizando JavaScript são: um editor de texto e um navegador de internet.

Os ambientes de configuração para o desenvolvimento em JavaScript podem variar devido aos comportamentos dos sistemas operacionais de computadores pessoais. Em sua maior parte, a experiência na construção de recursos em JavaScript é idêntica, independente de plataforma de sistema operacional. Embora possam existir ligeiras diferenças na codificação, dependendo do navegador ou do sistema operacional, mas a informação permanece a mesma (GOODMAN, 2001).

O editor de texto ou IDE é um componente importante para codificação em JavaScript, o editor de texto permite ao desenvolvedor digitar palavras exclusivas da linguagem JavaScript para futuramente serem interpretadas pelos navegadores.

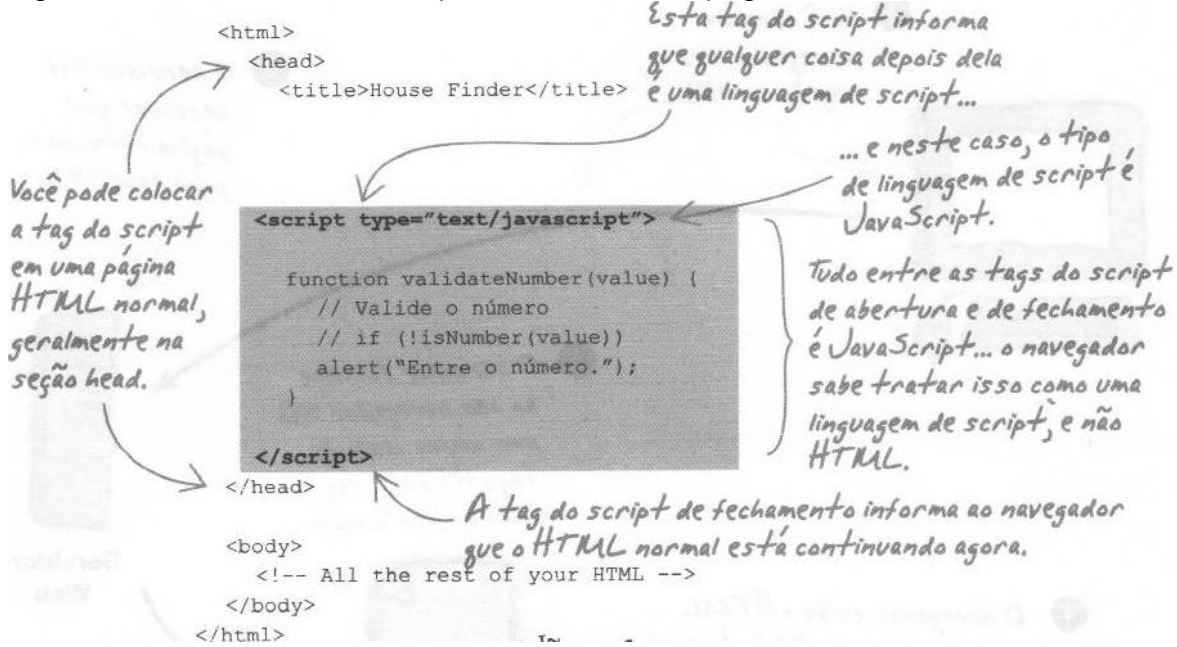
Para interpretação e/ou execução de arquivo fonte codificado em JavaScript é necessário um navegador (cliente web). Para testar e/ou executar um fonte em JavaScript não é preciso estar conectado à Internet (GOODMAN, 2001).

4.1.4 Iniciando Desenvolvimento com JavaScript Básico

Para iniciar o desenvolvimento com JavaScript é necessário integrá-lo com uma página que utiliza a linguagem de marcação HTML. Para isso, o JavaScript pode ser incluído diretamente em uma página HTML ou associado a ela através de um arquivo externo.

Para incluir o JavaScript diretamente na página é necessário informar ao navegador web que será adicionada codificação com JavaScript. O procedimento para informar o navegador que será incluso um código JavaScript é a declaração da tag “<script>” em qualquer lugar da página HTML, porém é recomendado colocá-la dentro da tag “<head>” (MORRISON, 2008).

Figura 7 - Incluindo o JavaScript diretamente em página HTML



Fonte: Morrison (2008, p.11)

Sobre a inclusão do JavaScript em página HTML, Rocha (1999, p.5) afirma:

> Há três maneiras de incluir JavaScript em uma página Web:
>
> Dentro de blocos HTML <SCRIPT> ... </SCRIPT> em várias partes da página: para definir funções usadas pela página, gerar HTML em novas páginas ou alterar o procedimento normal de interpretação do HTML da página pelo browser.
>
> Em um arquivo externo, importado pela página: para definir funções que serão usadas por várias páginas de um site.
>
> Dentro de descritores HTML sensíveis a eventos: para tratar eventos do usuário em links, botões e componentes de entrada de dados, durante a exibição da página.

A partir do momento em que o JavaScript já estiver incluído na página sejam por referência a um arquivo externo ou declaração dentro da própria página HTML os seus recursos podem ser interpretados pelo navegador no momento de execução da página web.

4.1.5 Começando a Desenvolver com JQUERY

O JQUERY é um framework escrito utilizando a linguagem JavaScript. Até a data de elaboração deste trabalho, a codificação do JQUERY é disponibilizada para download em um único arquivo.

O download pode ser feito diretamente pelo site do fabricante que é o www.jquery.com. Até o momento da elaboração deste trabalho, existe a possibilidade de baixar cópias compactadas e não compactadas.

Cópias não compactadas, ou popularmente tratadas como uncompressed são usadas no momento de desenvolvimento e depuração. Segundo o site corporativo do JQUERY a versão compactada, ou compressed é sugerida para ser usada em ambiente de produção, pois, economiza em largura de banda e consequentemente o desempenho tende a ser melhor (JQUERY, acesso: 29 ago. 2013).

Neste trabalho foi utilizado o framework JQUERY versão 1.10.2, não compactado. Feito a escolha da versão basta apenas fazer a chamada do JQUERY conforme demostra a figura 8, a qual ilustra a chamada do arquivo do framework local, ou seja, os arquivos de código fonte da aplicação e o arquivo do framework estão na mesma máquina e/ou servidor. Para entender de fato onde está sendo feita a chamada do framework basta observar que na imagem as linhas de código fonte são

enumeradas em ordem crescente e na linha de número nove (9) é feita a chamada do arquivo fonte do framework.

Figura 8 - Chamada do arquivo do framework local

```
1  <!doctype html>
2  <html>
3  <head>
4      <meta charset="utf-8" />
5      <title>Demo</title>
6  </head>
7  <body>
8      <a href="http://jquery.com/">jQuery</a>
9      <script src="jquery.js"></script>
10     <script>
11
12     // Your code goes here.
13
14     </script>
15 </body>
16 </html>
```

Fonte: <http://learn.jquery.com/about-jquery/how-jquery-works/>

O atributo src da tag script permite fazer uma chamada do arquivo do framework em outro servidor, mas, para isso é necessário que a máquina onde esteja o arquivo que faz referência a um servidor externo tenha acesso à Internet.

Figura 9 - Fazendo chamada do arquivo do framework em servidor externo

```
1  <!DOCTYPE html>
2  <html>
3    <head>
4      <title>jQuery Remoto</title>
5      <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
6      <!-- Fazendo a chamada do arquivo remoto -->
7      <script type="text/javascript" src="http://code.jquery.com/jquery-1.10.2.min.js"></script>
8    </head>
9    <body>
10   </body>
11 </html>
```

Fonte: Própria

Depois de escolhida a versão do framework, se compactado ou descompactado e o tipo de chamada, já está concluído o passo inicial com framework JQUERY, o que resta são as adaptações às necessidades da aplicação.

4.1.6 Comparativo de Performance

A escolha em utilizar um framework abstrai muito o esforço que precisa ser feito operacionalmente, optando pela utilização de um framework torna-se necessário

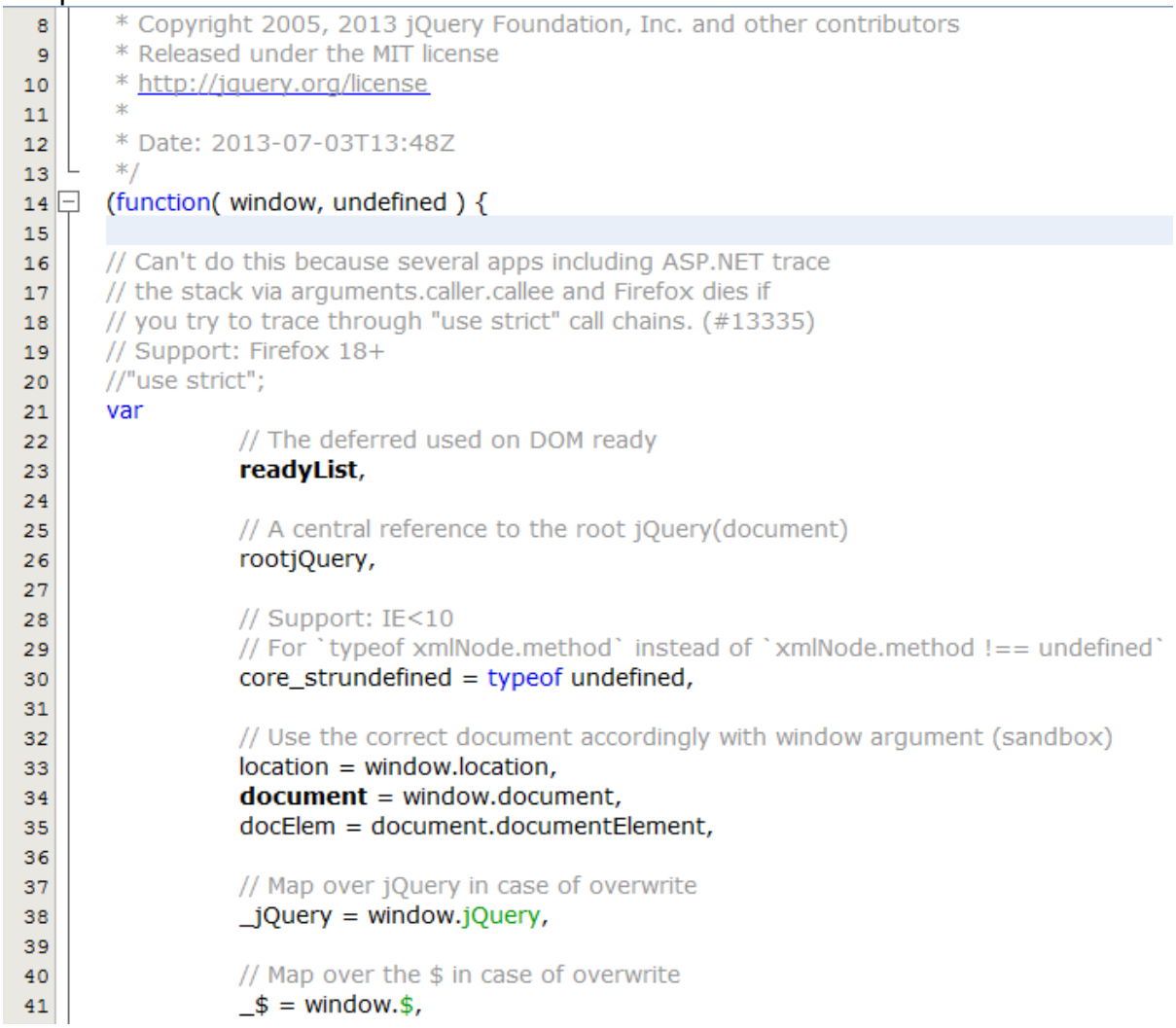
fazer a inclusão e/ou importação do arquivo de código fonte onde ficam codificados os recursos do framework.

Quando é incluído o framework em página HTML ou até mesmo outro arquivo que tenha a dependência do framework, consequentemente torna a página mais pesada devido o framework possuir inúmeros recursos e estes recursos são todos carregados no momento que as páginas são também carregadas.

A utilização do JavaScript nativo não requer a inclusão de arquivos externos. Os únicos arquivos que podem ser incluídos são arquivos codificados e preparados para reutilização. Para exemplificar imagine um recurso com a função de formatar uma data do formato americano para o formato brasileiro, ou seja, o programador codifica a solução apenas uma vez e faz a reutilização onde precisar, para isso basta incluir a referência onde está arquivo.

Visando reduzir o tamanho em kbytes do arquivo do framework JQUERY existe a opção de utilizar a versão compactada ou não compactada.

Figura 10 - Parte da codificação do JQUERY Framework 1.10.2 – Versão não compactada.

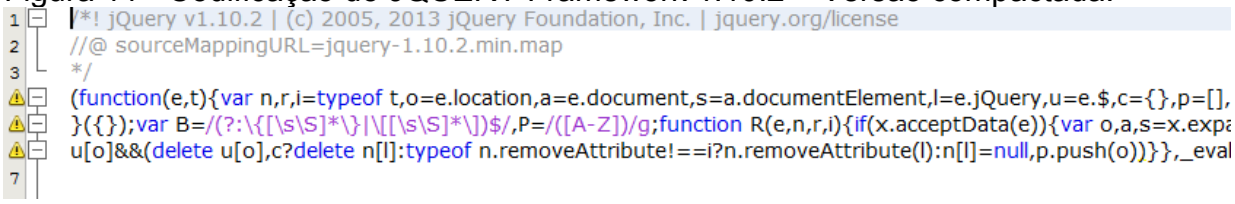


```
 8    * Copyright 2005, 2013 jQuery Foundation, Inc. and other contributors
 9    * Released under the MIT license
10    * http://jquery.org/license
11    *
12    * Date: 2013-07-03T13:48Z
13    */
14   (function( window, undefined ) {
15
16   // Can't do this because several apps including ASP.NET trace
17   // the stack via arguments.caller.callee and Firefox dies if
18   // you try to trace through "use strict" call chains. (#13335)
19   // Support: Firefox 18+
20   //"use strict";
21   var
22               // The deferred used on DOM ready
23               readyList,
24
25               // A central reference to the root jQuery(document)
26               rootjQuery,
27
28               // Support: IE<10
29               // For `typeof xmlNode.method` instead of `xmlNode.method !== undefined`
30               core_strundefined = typeof undefined,
31
32               // Use the correct document accordingly with window argument (sandbox)
33               location = window.location,
34               document = window.document,
35               docElem = document.documentElement,
36
37               // Map over jQuery in case of overwrite
38               _jQuery = window.jQuery,
39
40               // Map over the $ in case of overwrite
41               _$ = window.$,
```

Fonte: <http://code.jquery.com/jquery-1.10.2.js>

A figura 10 ilustra versão não compactada, nesta podem ser observada os comentários detalhados dos recursos do framework e espaçamento e endentação da codificação.

Figura 11 - Codificação do JQUERY Framework 1.10.2 – Versão compactada.



```
1  /*! jQuery v1.10.2 | (c) 2005, 2013 jQuery Foundation, Inc. | jquery.org/license
2  //@ sourceMappingURL=jquery-1.10.2.min.map
3  */
   (function(e,t){var n,r,i=typeof t,o=e.location,a=e.document,s=a.documentElement,l=e.jQuery,u=e.$,c={},p=[],
   }({});var B=/(?:\{[\s\S]*\}|\[[\s\S]*\])$/,P=/([A-Z])/g;function R(e,n,r,i){if(x.acceptData(e)){var o,a,s=x.expa
   u[o]&&(delete u[o],c?delete n[l]:typeof n.removeAttribute!==i?n.removeAttribute(l):n[l]=null,p.push(o))}},_eval
7
```

Fonte: <http://code.jquery.com/jquery-1.10.2.min.js>

A figura 11 ilustra versão compactada, nesta, pode ser observada que os comentários e endentação foram removidos da codificação. Utilizando o recurso de compactação o framework fica todo escrito em apenas três linhas, ignorando os

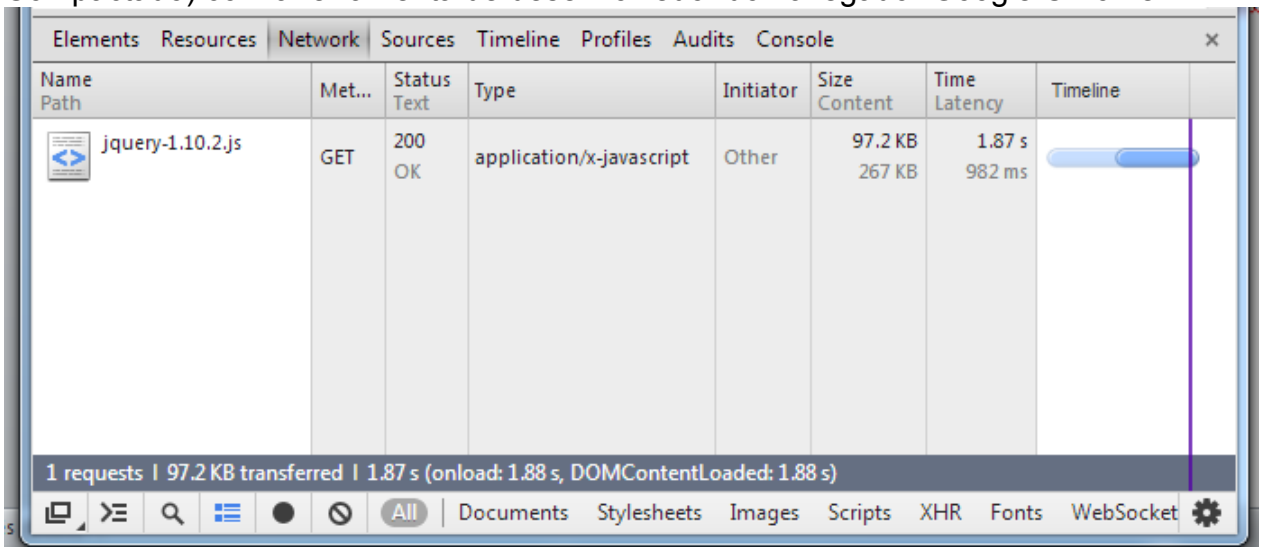
comentários que utilizam três linhas. Todo o processo de compactação e/ou compressão é para reduzir o tamanho do arquivo.

A versão compactada possui características como remoção dos comentários, espaçamento entre linhas e declarações, tornando o código difícil de ler para humanos e mais compacto, pois se transforma em uma sequência corrida de código, sem quebra de linhas (SILVA, 2008).

Utilizando ferramentas com objetivo de comparar o tamanho em bytes da versão compactada e não compactada, conclui-se que a versão não compactada tem tamanho de duzentos e sessenta e sete kbytes (267 KB), enquanto a versão compactada tem o tamanho inferior a noventa e um kbytes (91KB).

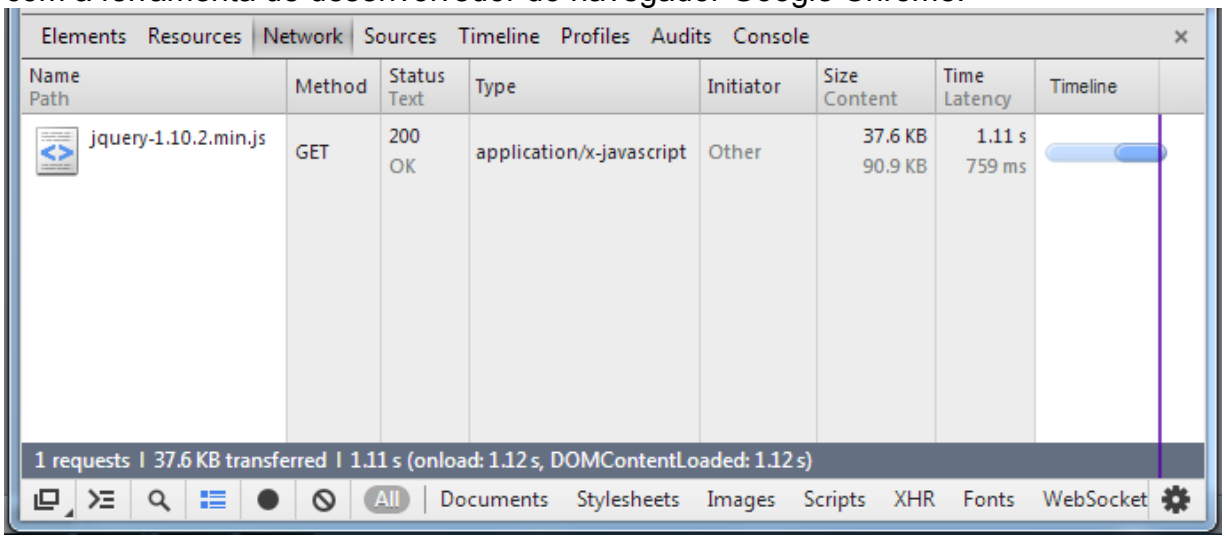
Os indicadores apresentados foram gerados com ferramenta de desenvolvedor inclusa no navegador de internet Google Chrome.

Figura 12 - Medindo o tamanho em KB do JQUERY framework 1.10.2 (Não Compactado) com a ferramenta de desenvolvedor do navegador Google Chrome



Fonte: <http://code.jquery.com/jquery-1.10.2.js>

Figura 13 - Medindo o tamanho em KB do JQUERY framework 1.10.2 (Compactado) com a ferramenta de desenvolvedor do navegador Google Chrome.



Fonte: <http://code.jquery.com/jquery-1.10.2.min.js>

O tamanho da versão compactada é relativamente pequeno, seu tamanho é aproximadamente noventa kbytes (90KB). Dependendo do servidor web que está sendo utilizando ainda existe o recurso de compressão para o formato gzip, utilizando este recurso o tamanho do arquivo pode chegar aproximadamente a trinta kbytes (30KB) (MCFARLAND, 2011).

Outra boa prática para aumentar a performance é fazer utilização do JQUERY páginas onde realmente é feita utilização do framework, ou seja, imagine uma funcionalidade em página onde o requisito seja apenas fazer a impressão de um texto ou a exibição de uma janela com um alerta. Neste caso é melhor a utilização do JavaScript nativo ao invés de incluir um arquivo de framework para fazer a chamada de um recurso que atenda ao requisito.

Sobre o comparativo, conclui-se:

- Utilização do JavaScript nativo não requer inclusão de arquivos para funcionamento padrão.
- O arquivo do framework JQUERY contém diversos recursos e na maioria das vezes nem todos os recursos são utilizados de uma única vez.
- Existem duas versões do arquivo do framework, sendo compactada e não compactada.

- O arquivo não compactado tem um tamanho de duzentos e sessenta e sete kbytes (267 KB), enquanto a versão compactada tem o tamanho inferior a noventa e um kbytes (90.9KB).

4.1.7 Comparativo de Produtividade

Segundo Jones (1996) a produtividade no desenvolvimento de sistemas é essencial tanto para empresas como para o próprio desenvolvedor. Em economia, a produtividade, de um modo geral, pode ser entendida como a relação entre a quantidade de bens ou serviços produzidos e a despesa ou trabalho necessário para produzí-los (JONES, 1996).

Um dos fatores que podem ser considerados como produtivo no desenvolvimento software é a utilização de um framework. Um framework possui recursos que valem a pena serem investigados, sendo eles: fácil aprendizado, desenvolvimento simplificado, desenvolvimento rápido e documentação completa (ALLEN; LO; BROWN, 2009).

Um framework, como o JQUERY, por exemplo, elimina muitas linhas de programação em JavaScript e muitas horas necessárias para testá-las, tudo isso apenas fazendo a chamada da função ou método (MCFARLAND, 2011).

A produtividade na utilização de framework pode ser detectada nos seguintes pontos:

- Quantidade de linhas de codificação produzida.
- Fato de não precisar desenvolver uma documentação para os recursos desenvolvidos.
- Simplicidade na utilização.
- Desenvolvimento mais rápido comparado ao desenvolvimento sem o framework.

O JavaScript nativo não é tão produtivo se comparado a um framework JavaScript. As soluções desenvolvidas com recursos nativos apenas utilizam os recursos do JavaScript, ou seja, com isso todas as soluções sejam elas fáceis ou não devem ser criadas a partir do marco zero.

A produtividade na codificação JavaScript pode ser quantificada inicialmente na quantidade de linhas produzidas para alcançar um determinado objetivo. Linhas e mais linhas de programação JavaScript escritas para obter simples efeitos em um objeto são substituídas por apenas algumas, escritas com a sintaxe do framework JQUERY (SILVA, 2008).

Para exemplificar parte da produtividade em números de linhas, o seguinte requisito é lançado e logo após são apresentadas as soluções codificadas em JavaScript e JQUERY. O requisito é construir um recurso seja ele um botão ou um link que ao ser clicado automaticamente marque todas as opções do formulário. As opções do formulário são elementos em HTML definidos como checkbox.

Para construir este exemplo será usada uma página em formato HTML. Esta página pode ser vista e/ou encontrada no APENDICE B. As figuras 14 e 15 ilustram os códigos em JavaScript que devem ser incluídos no APENDICE B. Seguindo recomendação da W3C os códigos devem ser incluídos dentro da tag "<head>".

Figura 14 - Codificação utilizando JavaScript para mesurar produtividade.

```
<script type="text/javascript">
   window.onload = function() {
      var linkMarcarTodos = document.getElementById("marcarTodos");
      if (linkMarcarTodos !== null) {
         linkMarcarTodos.onclick = function() {
            var opcoes = document.getElementsByName("opcoes");
            var quantidade = opcoes.length;
            var i = 0;
            for (i = 0; i < quantidade; i++) {
               opcoes[i].setAttribute("checked", "checked");
            }
         };
      }
   };
</script>
```

Fonte: Própria

Figura 15 - Codificação utilizando JQUERY para mesurar produtividade.

```
<script type="text/javascript" src="../js/jquery-1.10.2.js"></script>
<script type="text/javascript">
    $(document).ready(function() {
        $("#marcarTodos").click(function() {
            $("input[name=opcoes]").attr("checked", "checked");
        });
    });
</script>
```

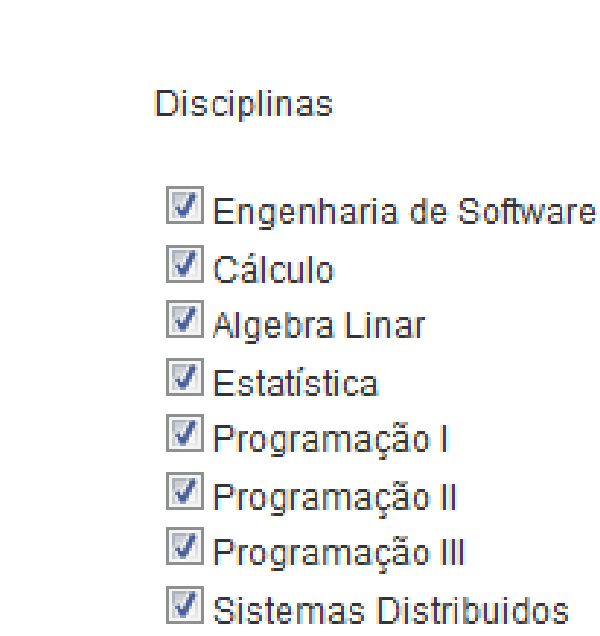
Fonte: Própria

As duas soluções codificadas e apresentadas pelas figuras 14 e 15 alcançam o mesmo objetivo. A principal diferença é quantidade de linhas e abstração de recursos aplicados com JQUERY.

Na solução codificada com JavaScript houver necessidade de aplicar esforços para localizar todos os elementos do tipo checkbox, verificar se existe no mínimo um elemento do tipo checkbox, caso exista, é disparado um recurso para percorrer todos os itens encontrados para que individualmente sejam marcados.

Na solução codificada com JQUERY foram utilizados os recursos de seletores para selecionar os itens do tipo checkbox. Logo após, foi utilizado um recurso que permite modificar e/ou atribuir valor aos atributos dos itens selecionados. Desta forma, para todos os componentes do tipo checkbox selecionados foram modificados e/ou atribuídos valores para o atributo checked.

Figura 16 - Resultado padrão da solução codificada tanto com JavaScript e JQUERY



Fonte: Própria

4.1.8 Comparativo de Curva de Aprendizado

JavaScript não é uma linguagem de fácil de aprendizado. JavaScript tem um embaraçoso segredo que é a dificuldade de escrevê-lo. Mesmo sendo mais simples comparando com outras linguagens, o JavaScript ainda é uma linguagem de programação (MCFARLAND, 2011).

Sobre o processo de aprendizagem os autores Bransford; Brown; Cocking (2007, p. 20) define:

> [...] que toda aprendizagem acontece em cenários que apresentam conjuntos específicos de normas e expectativas culturais e sociais, e que esses cenários influenciam a aprendizagem e a transferência de maneira marcante.

Há diversas possibilidades de aprendizagem, ou seja, há diversos fatores que levam o indivíduo apresentar comportamento que anteriormente não era apresentado, como o crescimento físico, descobertas tentativas e erros, ensino etc. (BOCK; FURTADO; TEIXEIRA, 2002).

O framework JQUERY é uma maneira simplificada e fácil de escrever JavaScript, sua simplicidade está ao alcance de programadores experientes, web designers e programadores com pouco conhecimento de programação (SILVA, 2008).

JQUERY é amigável para web designers e não assume que apenas um cientista da computação seja capaz de utilizá-lo. JQUERY aproveita o conhecimento CSS que a maioria dos web designers já têm (MCFARLAND, 2011).

Em uma análise do aprendizado da sintaxe com JavaScript nativo e com utilização do framework JQUERY, nota-se que o JavaScript utiliza sintaxe com escrita mais extensa e com dificuldade maior de aprendizado comparado com o JQUERY. Por outro lado o JQUERY procura padronizar sua sintaxe e apresenta recursos de utilização baseado em caracteres especiais e/ou chaves, para reduzir o tamanho das palavras chaves do framework. Veja no quadro 2 os esforços necessários aplicados para atender os requisitos com a utilização do JQUERY e o JavaScript nativo.

Quadro 2 - Apresentação do comparativo de sintaxe básicas em JavaScript nativo e JQUERY

Requisito	Sintaxe JavaScript	Sintaxe JQUERY
Selecionar um objeto pelo id	document.getElementyId("id")	$("#id")
Selecionar objeto(s) pela class	document.getElementsByClassName("classe")	$(".classe")
Selecionar objeto(s) pelo nome	document.getElementsByName("nome")	$("[name=nome]")
Selecionar objeto pelo elemento	document.getElementsByTagName("elemento")	$("elemento")
Evento ao clicar em elemento	document.getElementById("elemento").onclick = function(){}	$("elemento").click(function(){})

Fonte: Silva(2008)

Para os exemplos apresentados no quadro 2, observa-se que JQUERY utiliza os mesmos seletores do CSS. Quando for necessário selecionar um elemento por id com JQUERY é utilizado o caracter tralha “#”. Para combinar algum estilo utilizando identidificadores você usa o caracter tralha “#” no seletor (PFAFFENBERGER; SCHAFER; WHITE; KAROW, 2004).

Utilizando o seletor de classe, tanto em JQUERY quanto no CSS é utilizado o caractere especial ponto “.”. No CSS, para especificar uma classe para combinar com um seletor é necessário acrescentar um ponto e o nome da classe (PFAFFENBERGER; SCHAFER; WHITE; KAROW, 2004).

> […] If you're familiar with CSS, you're already well aware of the power of selectors, which describe groups of elements by their attributes or placement within the document. With JQUERY, you'll be able to leverage your knowledge and that degree of power to vastly simplify your JavaScript (BIBEAULT; KATZ, 2008, p.5).

Para desenvolver utilizando JavaScript e JQUERY é necessário ter no mínimo um conhecimento básico em HTML. Para que o aprendizado se torne ainda mais eficiente deve-se ampliar o conhecimento em tecnologias como XML e CSS.

Fazendo uma análise comparativa das tecnologias para a curva de aprendizado, os seguintes cenários devem ser levados em consideração:

- O JQUERY possui características como nomes dos métodos com sintaxe curta, nomes intuitivos para os métodos.
- JQUERY utiliza os mesmos seletores aplicados no CSS, por isso os web designers estão aptos a utilizar o framework.
- JQUERY simplifica a codificação por abstrair o esforço operacional que é feito com JavaScript.
- JavaScript tem sintaxe mais extensa, possui hierarquia de objetos para acessar métodos.
- JavaScript requer mais esforço na codificação.

Os pontos até aqui abordados são características que auxiliam no processo de aprendizado das tecnologias. Nesta lista é evidente que o JQUERY é superior pelos seus recursos de padronização e resgate de alguns recursos das tecnologias de outras linguagens como os seletores da linguagem de estilização CSS. O JavaScript por sua vez segue uma estrutura mais rígida para a codificação e utilização dos seus recursos.

Pelo fato do JQUERY ser um framework codificado em JavaScript aconselha-se que também tenha-se uma base de codificação com JavaScript, pois uma tecnologia complementa a outra, além disso proporciona uma aceleração da curva de aprendizado.

4.1.9 Comparativo de Portabilidade

Portabilidade na computação é um tópico de extrema importância e garantia para o funcionamento do componente a ser portável em seu determinado ambiente.

Portabilidade neste contexto pode ser entendida como a capacidade do produto de software de ser transferido de um ambiente para outro. Capacidade do produto de software de ser adaptado para diferentes ambientes especificados, sem necessidade de aplicação de outras ações ou meios além daqueles fornecidos para essa finalidade pelo software considerado (NBR ISO/IEC 9126-1, 2003).

Em uma definição computacional, a portabilidade pode ser entendida como qualidade de um componente ou de um programa informático que o torna capaz de ser usado em diferentes sistemas e computadores (DICIONARIO PRIBERAM, acesso em 16 out. 2013).

No desenvolvimento para internet utilizando as tecnologias de scripts uma das dificuldades encontradas é compatibilidade de funcionalidades em ambientes distintos, ambientes neste caso pode ser entendido como navegadores. O funcionamento dos recursos não são semelhantes em todos os navegadores, a diferença está em como é realizada a codificação das funcionalidades. Existe diferenças entre os diversos fabricantes de navegadores e até mesmo diferenças entre versões de navegadores do mesmo fabricante. Para exemplificar variação entre navegadores do mesmo fabricante, Silva (2010, p.439) define:

> O navegador Internet Explorer 9 suporta a manipulação de eventos com uso do DOM 2, contudo as versões anteriores desse navegador usam um método proprietário para a manipulação [...].

Mesmo existindo um órgão com a competência de desenvolver padrões para web, neste caso o W3C, ainda existem proprietários de navegadores que ainda implementam padrões próprios. Tal implementação, obriga aos programadores JavaScript desenvolverem soluções exclusivas que atendam os padrões proprietários.

Com o passar do tempo surgem novas versões e atualizações dos navegadores, juntamente surge também padronização e interpretação do JavaScript entre os

navegadores. Da lista dos principais e mais utilizados navegadores apenas o Internet Explorer, mantido pela Microsoft que ainda implementa padrão proprietário.

A preocupação do desenvolvedor com a portabilidade surge quando a codificação de uma determinada funcionalidade for fazer a utilização do JavaScript nativo. Algumas medidas devem ser tomadas para o controle do ambiente onde está sendo interpretado o JavaScript. Em alguns casos, para a manipulação de eventos, por exemplo, deve-se ter preocupação na codificação para que funcione em todos os navegadores. Silva (2010, p.441) afirma: “[...] para atrelar e desatrelar eventos de maneira que atenda a todos os navegadores é preciso servir códigos totalmente diferentes para a mesma função”.

Para exemplificar o contexto de evento apresentado anteriormente foi criada uma simples página preparada para manipulação de evento. Para este exemplo será utilizado o evento “click”, ou seja, quando for clicado no botão o evento “click” será acionado. Este exemplo foi testado nos navegadores Internet Explorer e no Mozilla Firefox. Para o Internet Explorer foi testado nas versões 7 (com módulo de compatibilidade) e 9, no Mozilla Firefox testado na versão 23.

Figura 17 - Código base de comparativo de compatibilidade entre navegadores utilizando JavaScript

```
<!DOCTYPE html>
<html>
    <head>
        <title>Comparativo</title>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <script type="text/javascript">
            window.onload = function(event) {
                var btn = document.getElementById('btn');
                if (btn.addEventListener) {
                    btn.addEventListener("click", function(){alert('Isto é addEventListener');});
                } else if (btn.attachEvent) {
                    btn.attachEvent("onclick", function() {alert('Isto é attachEvent');});
                }
            };
        </script>
    </head>
    <body>
        <button type="button" id="btn">Identificar</button>
    </body>
</html>
```

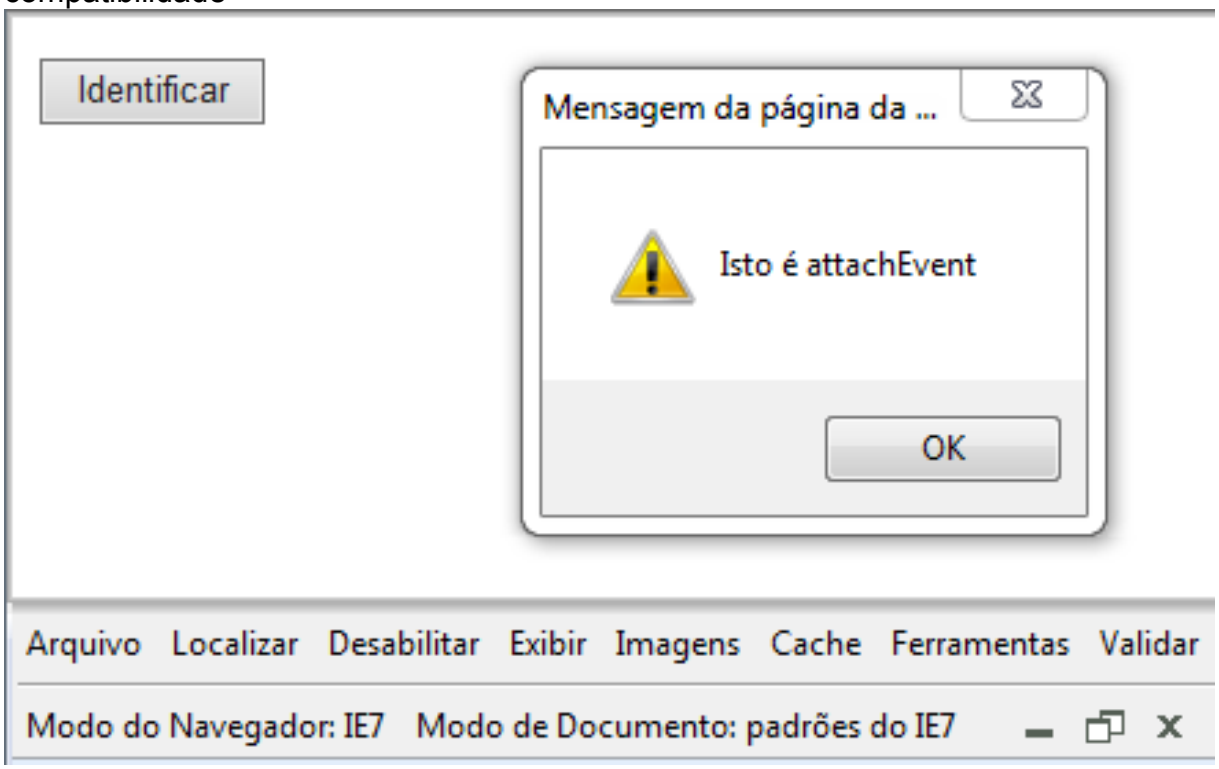
Fonte: Própria.

A codificação representada pela figura 17 terá os seguintes comportamentos:

- Se navegador for Internet Explorer versão 8 ou inferior será utilizado o comando “attachEvent”, para isso o sistema exibe uma alerta com o seguinte texto: “Isto é attachEvent”.
- Se navegador for Internet Explorer versão 9 ou superior será utilizado o comando “addEventListener”, para isso o sistema exibe uma alerta com o seguinte texto: “Isto é addEventListener”. Lembrando que versão mais atual até a data de elaboração deste trabalho é o Internet Explorer 10.
- Para os demais navegadores como Mozilla Firefox, Google Chrome, Safari e Opera será exibido uma alerta com o seguinte texto: “Isto é addEventListener”.

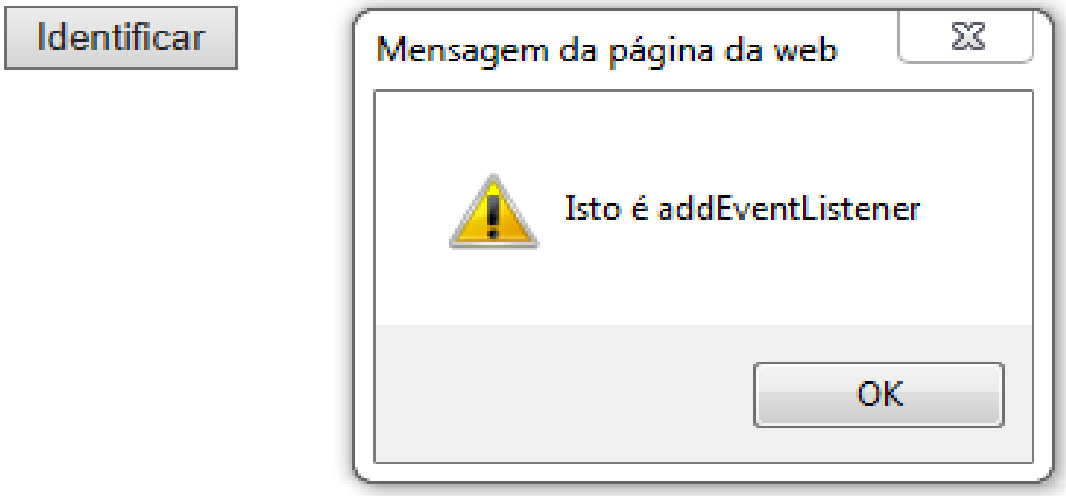
O que muda no exemplo apresentado é como é chamado o recurso que faz o vínculo objeto com o evento, podendo ser “addEventListener” ou “attachEvent”.

Figura 18 - Teste de manipulação de evento click no Internet Explorer 7 módulo de compatibilidade



Fonte: Própria

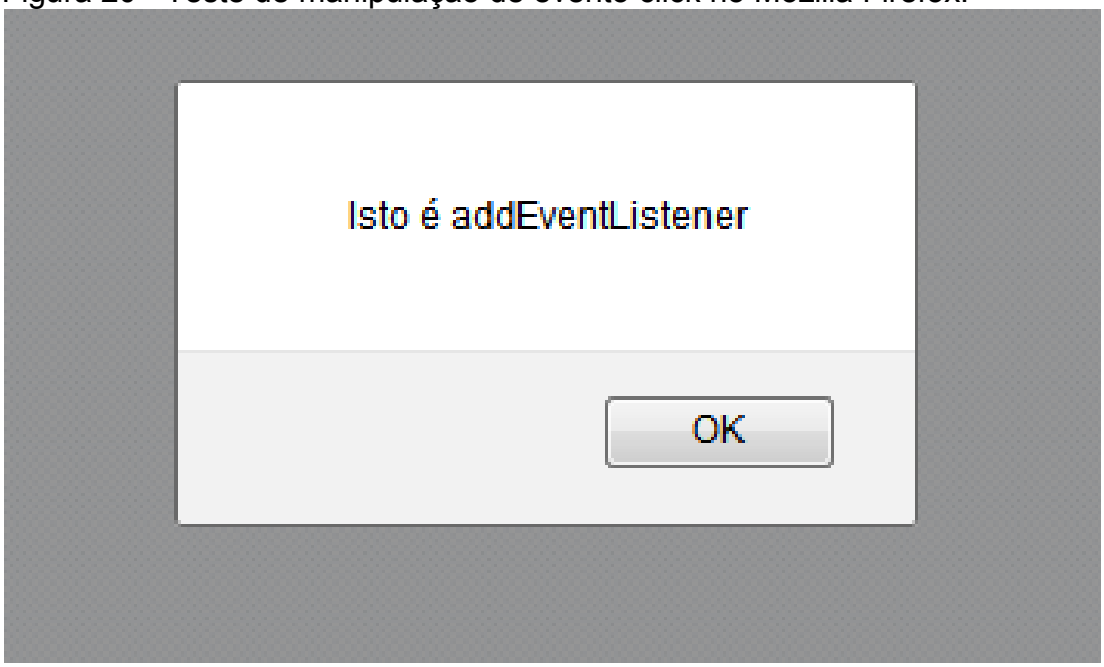
Figura 19 - Teste de manipulação de evento click no Internet Explorer 9.



Fonte: Própria

As figuras 18 e 19 apresentam os resultados obtidos no navegador Internet Explorer versões 7 e 9. Mesmo sendo navegadores do mesmo fabricante foi necessário codificar uma funcionalidade xclusiva para versões do Internet Explorer.

Figura 20 - Teste de manipulação de evento click no Mozilla Firefox.



Fonte: Própria

A figura 20 apresenta o resultado obtido do teste com o navegador Firefox, o qual não implementa padrões proprietários.

Os problemas de incompatibilidade apresentados anteriormente possuem soluções. Uma delas, é a preocupação em desenvolver para todos os navegadores ou utilizar um framework.

O JQUERY, por exemplo, é um framework que abstrai toda a preocupação de portabilidade entre navegadores que um desenvolvedor precisa ter ao desenvolver com JavaScript nativo. Sobre a portabilidade do framework JQUERY, Silva (2010, p.29) define: “[...] é indiferente às inconsistências de renderização entre navegadores”.

Para codificar uma solução sem que não necessite preocupar com portabilidade entre diferentes navegadores ou navegadores do mesmo fabricante é recomendada a utilização de um framework. Silva (2010, p.441) “[...] Alguns desenvolvedores optam pelo uso de bibliotecas JavaScript, tal como JQUERY, que cumprem essa finalidade [...]”. A figura 21 demostra a codificação com o mesmo objetivo representado figura 17 com um diferencial, que permite o desenvolvedor codificar sem preocupar com qual navegador irá interpretar a solução.

Figura 21 - Código base de comparativo de compatibilidade entre navegadores com utilização do JQUERY.

```
1  <!DOCTYPE html>
2  <html>
3     <head>
4        <title>Comparativo</title>
5        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
6        <script type="text/javascript" src="../js/jquery-1.10.2.js"></script>
7        <script type="text/javascript">
8           $(function() {
9              $("#btn").click(function(){alert('Solução com jQuery');});
10          });
11       </script>
12    </head>
13    <body>
14       <button type="button" id="btn">Identificar</button>
15    </body>
16 </html>
```

Fonte: Própria.

O framework JQUERY, abstrai conjuntos de esforços operacionais do JavaScript na manipulação dos recursos e possui um alto índice de compatibilidade. Mesmo com estas características ainda existem recursos que não possuem suporte a todas as

versões dos navegadores. A causa pela falta de suporte é a evolução das tecnologias dos navegadores e também a utilização de navegadores muito antigos.

O JQUERY, por exemplo, também possui limitações no suporte a todos os navegadores. No site do produtor do JQUERY está documentado o suporte aos navegadores por versão do navegador e versão do JQUERY.

Figura 22 - Informações do suporte do JQUERY aos navegadores

	Internet Explorer	Chrome	Firefox	Safari	Opera
jQuery 1.x	6+	(Current - 1) or Current	(Current - 1) or Current	5.1+	12.1x, (Current - 1) or Current
jQuery 2.x	9+				

Fonte: <http://jquery.com/browser-support/>

A figura 22 informa a respeito do suporte a versões, baseado nestas informações conclui-se que o JQUERY possui recursos que comparado ao JavaScript nativo torna-o com maior índice de portabilidade. Também é possível tornar o JavaScript portável apesar de ser aplicado com maior esforço. Outro fator que vale a pena lembrar é sobre a segunda versão do JQUERY que passa a fornecer suporte no navegador Internet Explorer apenas para versões nove ou superior.

4.1.10 Comparativo de Manutenibilidade

Manutenibilidade é a capacidade de manter um sistema (ENGHOLM JR, 2010). Entende-se manutenibilidade como capacidade do produto de software de ser modificado. As modificações podem incluir correções, melhorias ou adaptações do software devido a mudanças no ambiente e nos seus requisitos ou especificações funcionais (NBR ISO/IEC 9126-1, 2003).

A capacidade de manutenção de software enquadra-se dentro da lista de requisitos não funcionais definidos pela engenharia de software.

Na engenharia de software, a manutenção de software está associada a motivos diferentes, originando também diferentes tipos de manutenção. O tipo de manutenção corretiva será utilizada quando houver a necessidade de realizar modificação do software com objetivo de fazer correções de falhas. A manutenção evolutiva é utilizada na inclusão de novas funcionalidades ou ao desempenho do

software. E pôr fim a manutenção adaptativa refere-se à adaptação do software em ambientes operacionais diferentes. Baseado nos três tipos de manutenção, as principais atividades referentes à manutenção são: remoção de defeitos, adição, alteração e evolução de funcionalidades, ajustes, atualização e melhoria de desempenho (ENGHOLM JR, 2010).

A manutenção de software pode ser responsável por mais de 60% de todo o esforço despedido por uma organização de desenvolvimento, e a porcentagem continua a crescer à medida que mais software é produzido (PRESSMAN, 2006).

> De acordo com o Information Technology Research Institute, ETIS-project, o custo relacionado à manutenção de software e ao gerenciamento de sua evolução representa atualmente cerca de 90% do custo total (ENGHOLM JR, 2010, p.31).

Grande parte dos softwares existentes passam por atualizações e evoluções com o passar do tempo. ENGHOLM JR (apud MARTIN, 1983; NOSEK & PALVIA 1990; VAN VLIET, 2000) afirma: “[...] cerca de 75% dos custos de manutenção são relativos a fornecer evolução ao software, na forma de manutenção adaptativa e evolutiva [...]”.

Estudos também apontam que cerca de 50% do tempo é gasto no entendimento do código a ser mantido. ENGHOLM JR (apud FJELDSTAD & HAMLEN, 1983; STANDISH, 1984).

Existem séries de itens que é de extrema influência nos custos de manutenções do software, sendo: documentação, qualidade do software, paradigma de programação utilizado e o tamanho do software relacionado à quantidade de linhas de código (ENGHOLM JR, 2010).

No contexto de manutenção de softwares que utilizam JavaScript nativo ou JQUERY*,* comparando-os, conclui-se que:

- No quesito documentação, tanto o JavaScript e o JQUERY possuem extensa documentação dos proprietários, inúmeros livros e documentação digital e online.
- Para o quesito qualidade do software, tanto o JavaScript como JQUERY possuem recursos que agregam qualidade ao software e à forma de codificação. Tais recursos apenas estarão disponíveis nos softwares onde os mesmo forem aplicados, ou seja, fica cargo dos envolvidos agregarem qualidade ao software.

- No quesito de paradigma de programação utilizado existem diferenças. No JavaScript existe a possibilidade de escolher determinadas telas que serão desenvolvidas com orientação a objetos ou programação estruturada. Ainda existe possibilidade de utilizar as duas formas juntas em um software. No JQUERY já não existe a possibilidade de programação estruturada. O JQUERY foi escrito utilizando o paradigma de orientação a objetos, ou seja, toda chamada de métodos e/ou atributos deverão ser feitas pelo objeto JQUERY. Caso sejam feitas chamadas dos recursos ou parte deles em programação estrutura, esta deverá ser feita pelo objeto JQUERY.
- No quesito de tamanho do software relacionado à quantidade de linhas de código, na maior parte onde está realizando manutenção, com o JQUERY é notável que quantidade linhas produzidas são menores comparada ao JavaScript, isto até porque um framework existe para prover recursos de automatização de atividades rotineiras e repetitivas.

4.1.11 Comparativo de Recursos

Todos os recursos contidos no JQUERY foram codificados utilizando o JavaScript nativo, com isso assume-se que todos os recursos implementados pelo JQUERY são possível serem codificados com JavaScript nativo.

No comparativo de recursos entre o JavaScript nativo e o JQUERY é possível perceber que o JQUERY sai na frente pela facilidade, produtividade e conformidade com os navegadores e padrões web. Os fatores que fazem o JQUERY largar na frente no comparativo é pelo fato de já ser um framework documentado, testado, possuir abstração e compatibilidade com os principais navegadores.

Um ponto negativo na opção por utilização dos recursos de um framework é o quesito desempenho. São raras as circunstancia em que um desenvolvedor faz utilização de todos os recursos de um framework. Para não precisar carregar grandes arquivos contendo codificação de framework muitos desenvolvedores preferem desenvolver suas próprias funções JavaScript que atendam a necessidade e seja compatível com os navegadores (SILVA, 2010).

Os principais e mais utilizados recursos do JQUERY são: seletores, manipulação de DOM, eventos e tecnologia Ajax. Todos estes recursos encontram-se disponíveis no JavaScript nativo. O que os difere são as maneiras de codificação, quantidade de linhas codificadas e repetitivos testes para atender determinados requisitos.

Para compreender melhor a maneira como os recursos estão disponíveis tanto no JavaScript e no framework, e poder visualizar as diferenças de implementação, será apresentado uma página em HTML com retângulo. Junto deste retângulo haverá dois botões um deles com recurso de mostrar e outro com recurso de ocultar. Ao ser acionado qualquer um dos dois botões, será disparado um evento para mostrar ou ocultar o retângulo, dependerá de qual botão for acionado.

Figura 23 - Página HTML para comparativo de recursos

Fonte: Própria

Figura 24 - Codificação com JavaScript para o comparativo de recursos

```
<!DOCTYPE html>
<html>
   <head>
      <title>Comparativo de Recursos</title>
      <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
      <script type="text/javascript">
         window.onload = function() {
            document.getElementById('mostrar').onclick = function() {
               document.getElementById('bloco').style.display = '';
            };
            document.getElementById('ocultar').onclick = function() {
               document.getElementById('bloco').style.display = 'none';
            };
         };
      </script>
      <style type="text/css">
         #bloco {border:1px solid #000;width: 600px;height: 100px;background-color: #FF8000}
         #area-botao {text-align: center;width: 600px;padding: 2px;}
      </style>
   </head>
   <body>
      <div id="bloco" style=""></div>
      <div id="area-botao">
         <button type="button" id="mostrar">Mostrar</button>
          <button type="button" id="ocultar">Ocultar</button>
      </div>
   </body>
</html>
```

Fonte: Própria

Figura 25 - Codificação com JQUERY para o comparativo de recursos

```
<!DOCTYPE html>
<html>
   <head>
      <title>Comparativo de Recursos</title>
      <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
      <script type="text/javascript" src="js/jquery-1.10.2.js"></script>
      <script type="text/javascript">
         $(function() {
            $("#mostrar, #ocultar").click(function(a) {
               (this.id === 'ocultar') ? $("#bloco").hide() : $("#bloco").show()
            });
         });
      </script>
      <style type="text/css">
         #bloco {border:1px solid #000;width: 600px;height: 100px;background-color: #FF8000}
         #area-botao {text-align: center;width: 600px;padding: 2px;}
      </style>
   </head>
   <body>
      <div id="bloco" style=""></div>
      <div id="area-botao">
         <button type="button" id="mostrar">Mostrar</button>
          <button type="button" id="ocultar">Ocultar</button>
      </div>
   </body>
</html>
```

Fonte: Própria

O resultado apresentado na figura 23 pode ser codificado conforme demostra as figuras 24 e 25, os recursos utilizados para a codificação foram seletores, eventos e manipulação de DOM.

Sobre a utilização dos recursos das tecnologias conclui-se que o JQUERY, por ser um framework, tem a responsabilidade de abstrair a parte operacional do desenvolvimento. A parte operacional que o framework propõe a resolver ou facilitar, são preocupações como: funcionamento nos principais navegadores, documentação, aprendizado e simplicidade. Por outro lado, utilizando o JavaScript nativo torna-se necessário à criação de recursos como portabilidade, aproveitamento ou reutilização de codificação, facilidade de implantação e documentação. Todos estes itens consomem muito tempo para sua idealização, o que acaba sendo fator decisivo para a utilização de um framework.

4.2 RESULTADO DA PESQUISA

Esta seção apresenta os resultados obtidos da aplicação de uma pesquisa de campo na região metropolitana da Grande Vitória, estado do Espírito Santo, Brasil. A

pesquisa foi elaborada com intuito de capturar a referência de mercado relativo à utilização das tecnologias JavaScript e o framework JQUERY.

A pesquisa contou com a participação de onze profissionais atuante no mercado da Grande Vitória em distintos ramos de atuação do mercado. Os profissionais entrevistados estão distribuídos em nove empresas distintas. A pesquisa não levou em consideração o nível de experiência ou conhecimento nas tecnologias avaliadas de cada profissional, o requisito foi apenas fazer a utilização das tecnologias avaliadas.

Os participantes responderam um questionário que pode ser visualizado no APENDICE A deste documento. O questionário aplicado foi elaborado para resgatar informações quantitativa e qualitativa dos participantes.

O questionário foi divido em duas partes. Na primeira parte, os participantes responderam seis questões quantitativas, também houve a possibilidade de capturar informações caso o participante tenha interesse em fazer complementações, observações e/ou colaboração para as questões quantitativas. Nas questões quantitativas o objetivos é quantificar indicadores como utilização, performance, produtividade, curva de aprendizado, portabilidade e manutenibilidade. Para as questões quantitativas, os participantes atribuíram notas entre zero a cinco para cada questão. Zero, a menor nota representou que o participante não sabe ou não possui habilidades para avaliá-las. Para as notas entre um e cinco representa que o participante soube avaliar a questão e utilizou notas entre um e cinco, onde cinco representa nota máxima nos critérios de avaliação.

As tecnologias avaliadas neste questionário são o JavaScript nativo e o framework JQUERY. Com o objetivo de não limitar os participantes apenas à utilização de um framework e não aparentar um estudo tendencioso, foi elaborada uma terceira alternativa definida como “outro”, que pode ser entendido como outro framework e o participantes não tem a obrigatoriedade de avaliar e/ou atribuir nota. A criação do terceiro item não entra como fator criterioso para interpretação, contabilização e divulgação de resultados pelo fato deste estudo limitar-se apenas na avaliação do JavaScript nativo e o framework JQUERY.

Para a segunda parte do questionário foi utilizado o método qualitativo onde o participante define pontos fortes e fracos das tecnologias avaliadas.

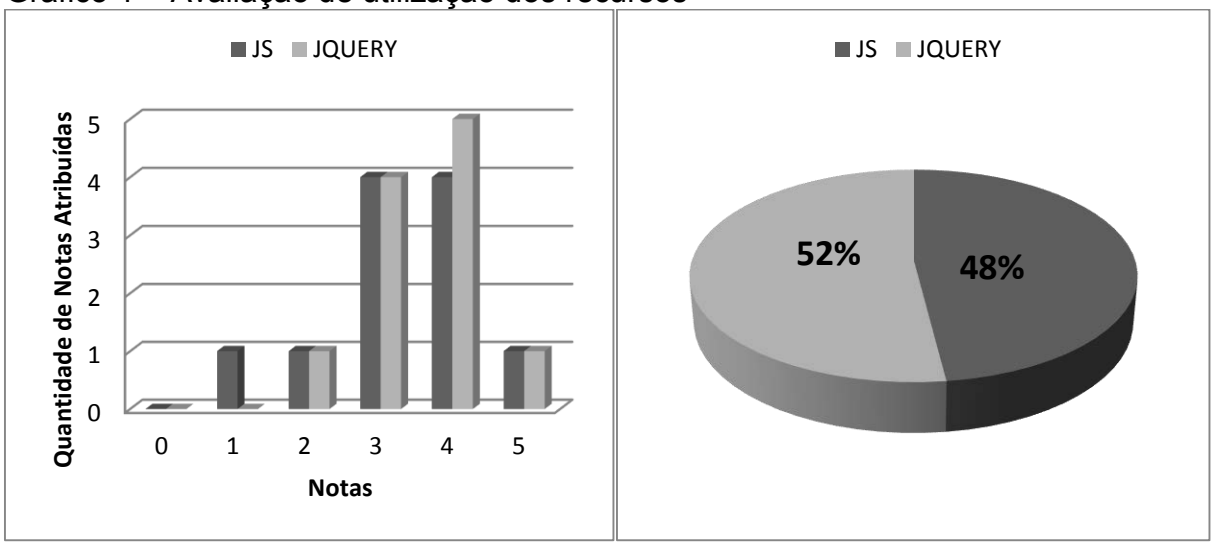
Os resultados obtidos serão apresentados por cada indicador nas subseções desta seção.

O resultado consolidado da pesquisa para os indicadores avaliados é representado pelo gráfico de “pizza”. Consolidação, neste caso, pode ser entendida como a representação de avaliação geral dos entrevistados. A metodologia utilizada para confecção dos gráficos pode ser acompanhada pelo APÊNDICE C.

4.2.1 Resultado da Avaliação de Utilização dos Recursos

Para obter resultado do indicador de utilização dos recursos das tecnologias, os participantes da pesquisa responderam a questão onde o objetivo era capturar a utilização dos recursos disponíveis em cada tecnologia.

Gráfico 1 – Avaliação de utilização dos recursos



Fonte: Questionário de pesquisa

Os resultados para o indicador de utilização de recursos apontam que:

- Nenhum participante atribui nota zero para as tecnologias avaliadas.
- Utilizando nota um, o JQUERY não foi avaliado e para JavaScript apenas 9,09% avaliaram.
- Utilizando nota dois, 9,09% representam às avaliações para cada uma das tecnologias.
- Utilizando nota três, 36,36% representam às avaliações para cada uma das tecnologias.

- Utilizando nota quatro, 36,36% representam às avaliações para o JavaScript, 45,45% representam às avaliações para o JQUERY.
- Utilizando nota cinco, 9,09% representam às avaliações para cada uma das tecnologias.

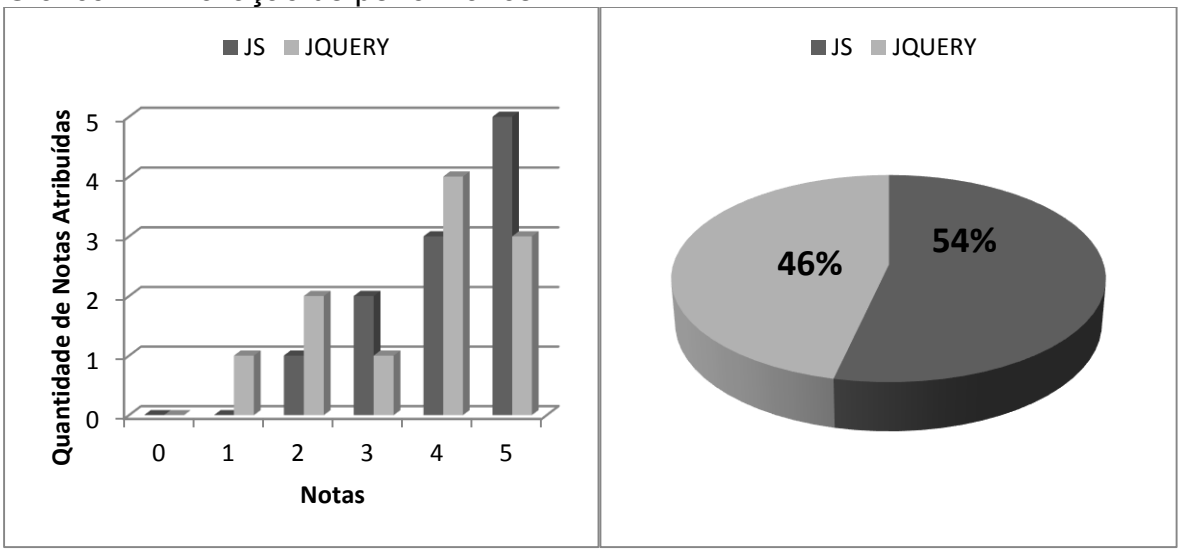
Consolidando o resultado da pesquisa do indicador de utilização de recursos, os resultados apontam o JQUERY com 52,00% dos votos foi avaliado tendo seus recursos mais utilizados, 48,00% representa a utilização dos recursos do JavaScript.

Os participantes da pesquisa puderam também expressar alguma observação ou colaborar com algum comentário para a questão avaliada. Para a pergunta que avalia a utilização dos recursos as tecnologias houve um dos participantes que afirmou que o JavaScript e o JQUERY possuem os mesmos recursos, a diferença está presente na facilidade providenciada pelo JQUERY, que visa evitar o retrabalho e reduzir a quantidade de linhas de código.

4.2.2 Resultado da Avaliação de Performance

Para obter resultado do indicador de performance, os participantes da pesquisa responderam a questão onde o objetivo foi capturar qual a avaliação de performance das tecnologias avaliadas.

Gráfico 2 - Avaliação de performance



Fonte: Questionário de pesquisa

Os resultados para o indicador de performance apontam que:

- Nenhum participante atribuiu nota zero para as tecnologias avaliadas.
- Utilizando nota um, para o JavaScript não foi atribuído nota, para o JQUERY apenas 9,09% dos participante atribuíram suas notas.
- Utilizando nota dois, 9,09% representam os participantes que atribuíram suas notas para o JavaScript, 18,18% representam as notas atribuídas para o JQUERY.
- Utilizando nota três, 18,18% representam os participantes que atribuíram notas para o JavaScript, 9,09% representam as notas atribuídas para o JQUERY.
- Utilizando nota quatro, 27,27% representam os participantes que atribuíram notas para o JavaScript, 36,36% representam as notas atribuídas para o JQUERY.
- Utilizando nota cinco, 45,45% representam os participantes que atribuíram notas para o JavaScript, 27,27% representam as notas atribuídas para o JQUERY.

Consolidando os resultados da pesquisa para o indicador de performance, a referência de mercado aponta o JavaScript com o melhor índice de performance comparado com o JQUERY. O JavaScript obteve 54,00% das notas e o JQUERY com 46,00%.

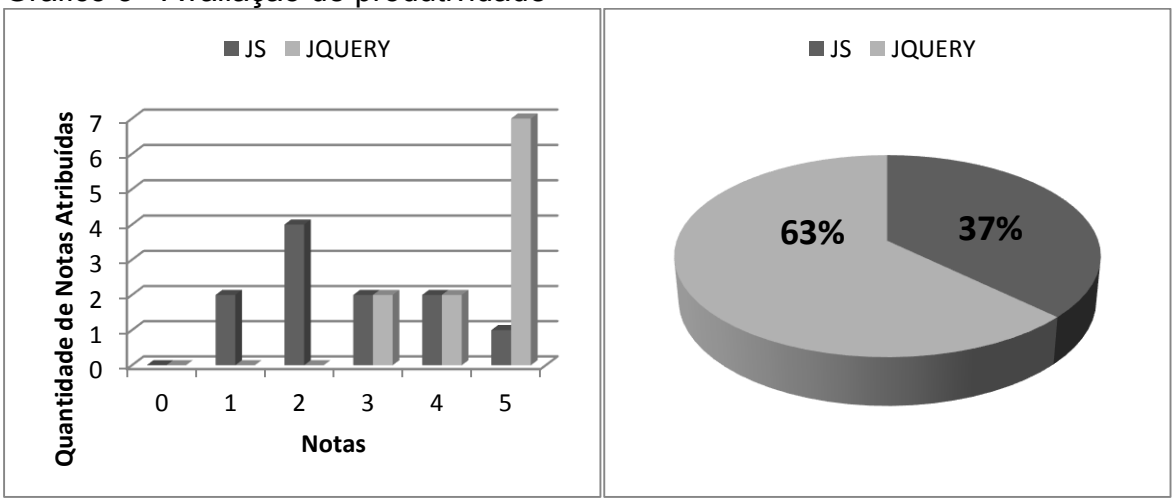
Os participantes avaliando o indicador de performance das tecnologias também puderam complementar e/ou colaborar expressando seu parecer para questão. Os participantes expressaram seu parecer comentando:

- JQUERY possui performance satisfatória e está em constante aperfeiçoamento podendo aproximar-se cada vez mais do JavaScript.
- JavaScript e JQUERY tem boa performance, pelo fato de ser carregado no computador do internauta. Mas é aconselhável que o desenvolvedor utilize algumas técnicas importantes para aperfeiçoar seus scripts, uma boa prática, por exemplo, é deixar o JavaScript sempre carregar no rodapé das páginas e ter cuidado com a quantidade de scripts adicionadas na página.
- A função do JQUERY é facilitar as operações, mas em contrapartida, pela quantidade de recursos, algumas operações são menos performáticas. Mas existem métodos que podem melhorar o desempenho do JQUERY.

4.2.3 Resultado da Avaliação de Produtividade

Para obter resultado do indicador de produtividade, os participantes da pesquisa responderam uma questão onde o objetivo foi capturar a avaliação de produtividade das tecnologias avaliadas.

Gráfico 3 - Avaliação de produtividade



Fonte: Questionário de pesquisa

Os resultados para o indicador de produtividade apontam que:

- Nenhum participante atribuiu nota zero para as tecnologias avaliadas.
- Utilizando nota um, 18,18% representam os participantes que atribuíram notas para o JavaScript, para o JQUERY não houve atribuição de notas.
- Utilizando nota dois, 36,36% representam os participantes que atribuíram notas para o JavaScript, para o JQUERY não houve atribuição de notas.
- Utilizando nota três, 18,18% representam às avaliações para cada uma das tecnologias.
- Utilizando nota quatro, 18,18% representam às avaliações para cada uma das tecnologias.
- Utilizando nota cinco, 9,09% representam os participantes que atribuíram notas para o JavaScript, 63,64% representam as notas atribuídas para o JQUERY.

Consolidando os resultados da pesquisa para o indicador de produtividade, a referência de mercado aponta o JQUERY com o melhor índice de produtividade comparado com o JavaScript. O JavaScript obteve 37,00% das notas e o JQUERY com 63,00%.

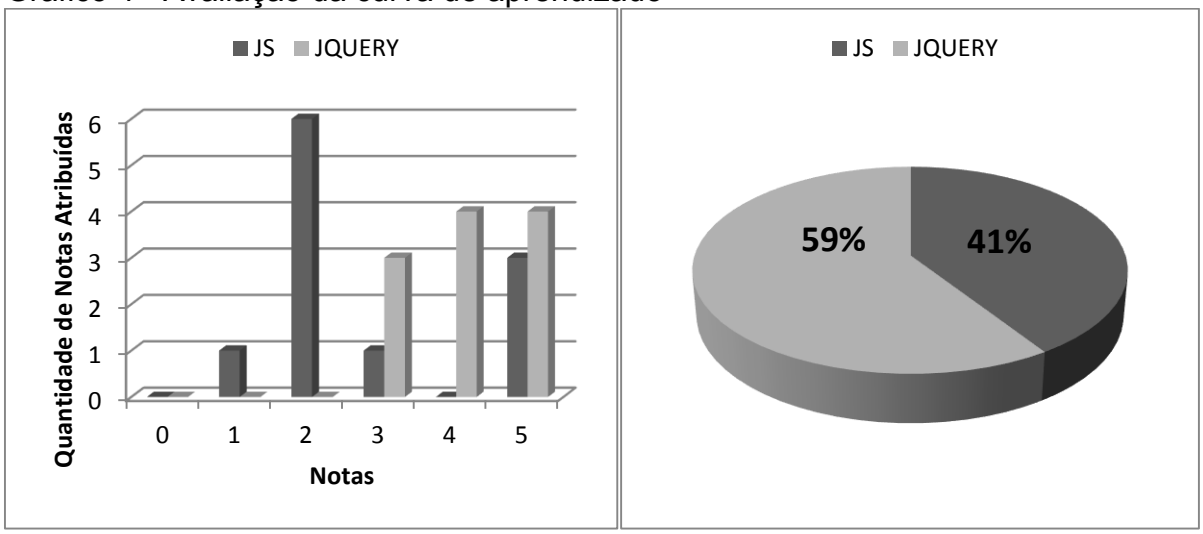
Os participantes avaliando a questão do indicador de produtividade das tecnologias também puderam complementar e/ou colaborar expressando seu parecer para questão. Os participantes expressaram seu parecer comentando:

- Muitas vezes precisamos escrever muito para fazer pouco com JavaScript, este é o meu principal motivo para a adoção do JQUERY. Claro que isso varia de acordo com o conhecimento do desenvolvedor tanto na linguagem como no projeto.
- A produtividade com JavaScript é pouco menor, pois o desenvolvedor tem que criar tudo na mão do ponto zero, já o JQUERY tem muitos plug-ins prontos e funções que com uma linha de código o desenvolvedor faz o que gostaria de fazer.
- Por ser uma biblioteca baseada em JavaScript, o JQUERY possui diversas funções para agilizar os procedimentos que serão efetuados no lado cliente. Por isso, o mesmo possui uma produtividade maior, pois diminui o retrabalho.

4.2.4 Resultado Avaliação de Curva de Aprendizado

Para obter resultado do indicador de curva de aprendizado, os participantes da pesquisa responderam a questão onde o objetivo foi capturar a avaliação da curva de aprendizado necessário para desenvolver com as tecnologias avaliadas.

Gráfico 4 - Avaliação da curva de aprendizado



Fonte: Questionário de pesquisa

Os resultados para o indicador de curva de aprendizado apontam que:

- Nenhum participante atribuiu nota zero para as tecnologias avaliadas.
- Utilizando nota um, 9,09% representam os participantes que atribuíram notas para o JavaScript, para o JQUERY não houve atribuição de notas.
- Utilizando nota dois, 54,55% representam os participantes que atribuíram notas para o JavaScript, para o JQUERY não houve atribuição de notas.
- Utilizando nota três, 9,09% representam os participantes que atribuíram notas para o JavaScript, 27,27% representam as notas atribuídas para o JQUERY.
- Utilizando nota quatro, não houve participante que atribuí nota para o JavaScript, 36,36% representam as notas atribuídas para o JQUERY.
- Utilizando nota cinco, 27,27% representam os participantes que atribuíram notas para o JavaScript, 36,36% representam as notas atribuídas para o JQUERY.

Consolidando os resultados da pesquisa para indicador de avaliação da curva de aprendizado, a referência de mercado aponta o JQUERY com 59,00% possuí a maior curva de aprendizado no menor tempo possível, 41% representa a avaliação para o JavaScript.

Os participantes avaliando a questão do indicador de curva de aprendizado das tecnologias também puderam complementar e/ou colaborar expressando seu parecer para questão. Os participantes expressaram seu parecer comentando:

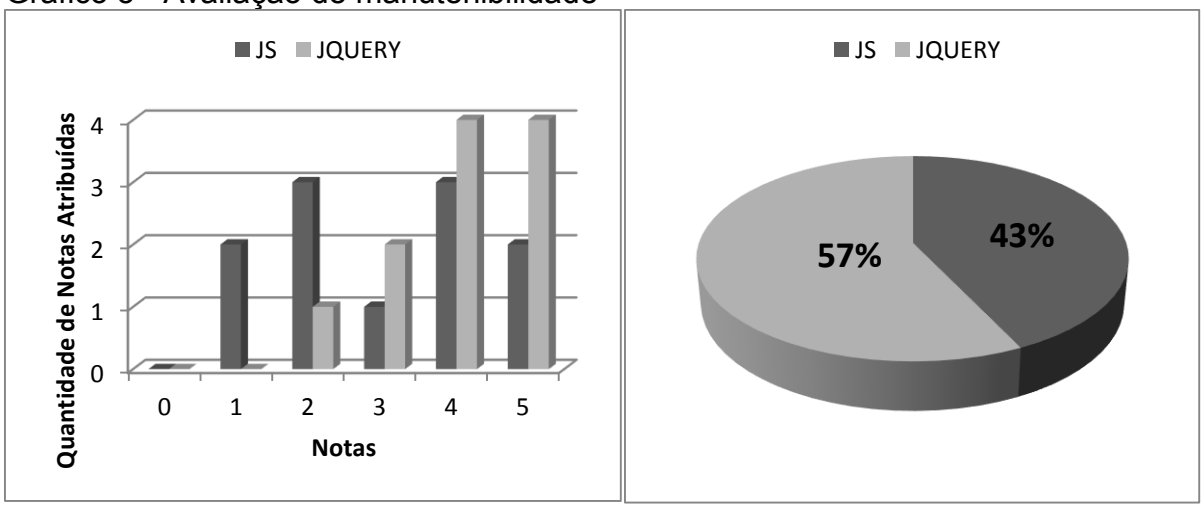
- A curva de aprendizagem do JQUERY é maior devido à abstração de seu código. O que torna mais legível e familiar ao programador. Já o JavaScript tem uma curva de aprendizado diferente, pois é lenta no começo, mas quanto mais se sabe da linguagem, mais essa curva de torna íngreme até o ponto que fica normal, como em outras linguagens. Isso acontece, pois a base do script é difícil de aprender, mas depois de aprendida, fica fácil aprender outras coisas da linguagem.
- O JavaScript em si é uma linguagem fácil. Como o JQUERY se baseia em um JavaScript é necessário ter um conhecimento do mesmo para entender melhor o JQUERY. A função básica do JQUERY e facilitar as operações em JavaScript com alguma melhorias. Portanto, a curva de aprendizados dos dois são equivalentes.

- No meu caso é difícil fazer essa avaliação de uma forma não tendenciosa, pois aprendi JavaScript antes do JQUERY, o que tornou muito mais fácil o aprendizado do JQUERY

4.2.5 Resultado da Avaliação de Manutenibilidade

Para obter resultado do indicador de manutenibilidade, os participantes da pesquisa responderam uma questão onde o objetivo foi capturar a avaliação da manutenibilidade das tecnologias avaliadas.

Gráfico 5 - Avaliação de manutenibilidade



Fonte: Questionário de pesquisa

Os resultados para o indicador de manutenibilidade apontam que:

- Nenhum participante atribuiu nota zero para as tecnologias avaliadas.
- Utilizando nota um, 18,18% representam os participantes que atribuíram notas para o JavaScript, para o JQUERY não houve atribuição de notas.
- Utilizando nota dois, 27,27% representam os participantes que atribuíram notas para o JavaScript, 9,09% representam as notas atribuídas para o JQUERY.
- Utilizando nota três, 9,09% representam os participantes que atribuíram notas para o JavaScript, 18,18% representam as notas atribuídas para o JQUERY.
- Utilizando nota quatro, 27,27% representam os participantes que atribuíram notas para o JavaScript, 36,36% representam as notas atribuídas para o JQUERY.

- Utilizando nota cinco, 18,18% representam os participantes que atribuíram notas para o JavaScript, 36,36% representam as notas atribuídas para o JQUERY.

Consolidando os resultados da pesquisa para indicador manutenibilidade, a referência de mercado aponta o JQUERY com 57,00% possuí a maior facilidade para ser executadas as devidas manutenções, 41,00% representa percentual de manutenibilidade para o JavaScript.

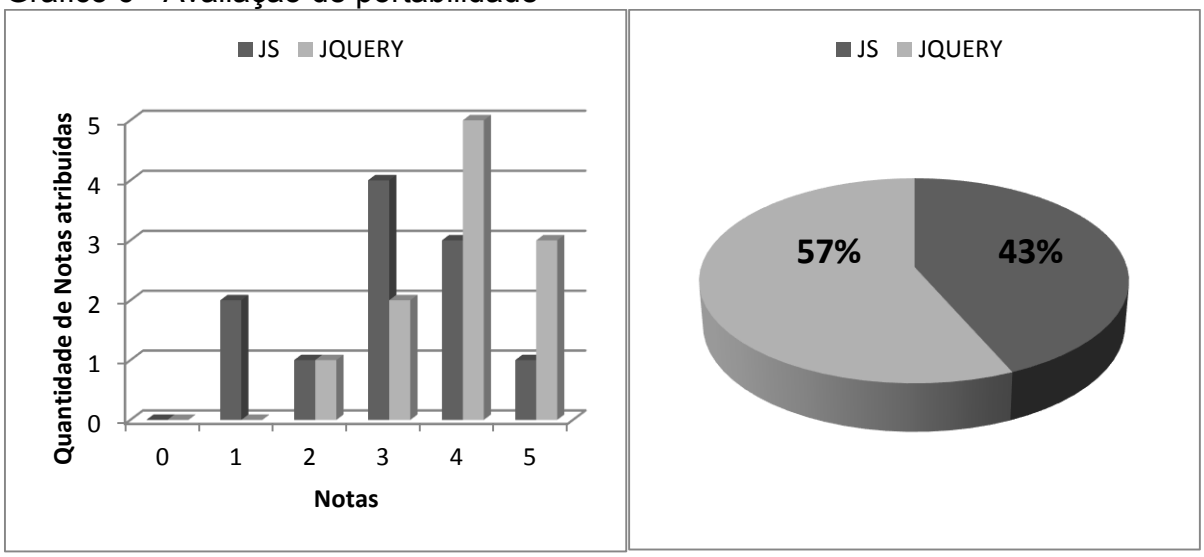
Os participantes avaliando a questão do indicador de manutenibilidade das tecnologias também puderam complementar e/ou colaborar expressando seu parecer para questão. Os participantes expressaram seu parecer comentando:

- A rastreabilidade sempre foi pobre no JavaScript. Desenvolvedores novos em um determinado projeto podem sofrer bastante para localizar um determinado recurso. Apesar das ferramentas para desenvolvedores que existem nos navegadores ajudarem muito, ainda acho que poderia ser melhor.
- Tudo vai depender de como o desenvolvedor vai organizar seus arquivos e funções, como em quaisquer outras linguagens.
- Os dois possuem o mesmo nível de manutenibilidade, pois cada uma possui suas particularidades.

4.2.6 Resultado da Avaliação de Portabilidade

Para obter resultado do indicador de portabilidade, os participantes da pesquisa responderam uma questão onde o objetivo foi capturar a avaliação da portabilidade das tecnologias avaliadas.

Gráfico 6 - Avaliação de portabilidade



Fonte: Questionário de pesquisa

Os resultados para o indicador de portabilidade apontam que:

- Nenhum participante atribuiu nota zero para as tecnologias avaliadas.
- Utilizando nota um, 18,18% representam os participantes que atribuíram notas para o JavaScript, para o JQUERY não houve atribuição de notas.
- Utilizando nota dois, 9,09% representam os participantes que atribuíram notas para o JavaScript e JQUERY.
- Utilizando nota três, 36,36% representam os participantes que atribuíram notas para o JavaScript, 18,18% representam as notas atribuídas para o JQUERY.
- Utilizando nota quatro, 27,27% representam os participantes que atribuíram notas para o JavaScript, 45,46% representam as notas atribuídas para o JQUERY.
- Utilizando nota cinco, 9,09% representam os participantes que atribuíram notas para o JavaScript, 27,27% representam as notas atribuídas para o JQUERY.

Consolidando os resultados da pesquisa para o indicador de avaliação da portabilidade, a referência de mercado aponta que com 57,00%, o JQUERY é considerado pelos participantes como o mais portável comparando com o JavaScript que obtém 41%,00 na avaliação de portabilidade.

Os participantes avaliando a questão do indicador de portabilidade das tecnologias também puderam complementar e/ou colaborar expressando seu parecer para questão. Os participantes expressaram seu parecer comentando:

- Muitas das funções JavaScript não funcionam em dispositivos móveis, mas desenvolvendo com conhecimento prévio delas, podemos antecipar esse problema e deixar o código portátil. Resumindo, a portabilidade utilizando esse tipo de linguagem é proporcional ao conhecimento do desenvolvedor. Quanto menos conhecimento sobre a linguagem, menos portátil será o código desenvolvido.
- Por se tratar de um framework, o JQUERY possui funções para prever a portabilidade, enquanto o JavaScript se mantém fortemente em cima do browser, acarretando em alguns problemas de compatibilidade ao mudar de plataforma. (Exemplo: Carregar uma página web em um dispositivo móvel).

4.2.7 Resultado da Avaliação dos Pontos Fortes

Na segunda parte do questionário foram elaboradas questões qualitativas. Os participantes foram submetidos a contribuir com a pesquisa avaliando os pontos fortes do JavaScript e JQUERY.

Os questionários foram consolidados e os resultados referentes aos pontos fortes das tecnologias avaliadas podem ser vistas quadro 3.

Quadro 3 – Resultado da pesquisa de avaliação dos pontos fortes do JavaScript e do JQUERY.

JavaScript	JQUERY
Base de pesquisa	Abstração de codificação
Bom desempenho	Codificação legível
Fácil aprendizado	Cross browser
Facilidade em desenvolver pequenos algoritmos	Documentação
Interação com o usuário	Efeitos
Ligação direta com recursos do browser	Fácil aprendizado
Linhas de código reduzidas	Fácil utilização
JavaScript	Facilidade de entendimento
Muito conhecido e disseminado	Facilidade em localizar componentes
Não dependência de servidor	Grande comunidade
Processamento na máquina cliente	Manipulação de eventos
Utilização de eventos no navegador	Manutenção
Utilização de recursos do cliente	Orientação a aspecto
Validações	Plug-ins
	Portabilidade
	Processamento na máquina cliente
	Produtividade
	Quantidade de recursos prontos

Fonte: Questionário de pesquisa

4.2.8 Resultado da Avaliação dos Pontos Fracos

Na segunda parte do questionário foram elaboradas questões qualitativas. Os participantes foram submetidos a contribuir com a pesquisa avaliando os pontos fracos do JavaScript e JQUERY.

Os questionários foram consolidados e os resultados referentes aos pontos fracos das tecnologias avaliadas podem ser vistas no quadro abaixo.

Quadro 4 - Resultado da pesquisa de avaliação dos pontos fracos do JavaScript e do JQUERY.

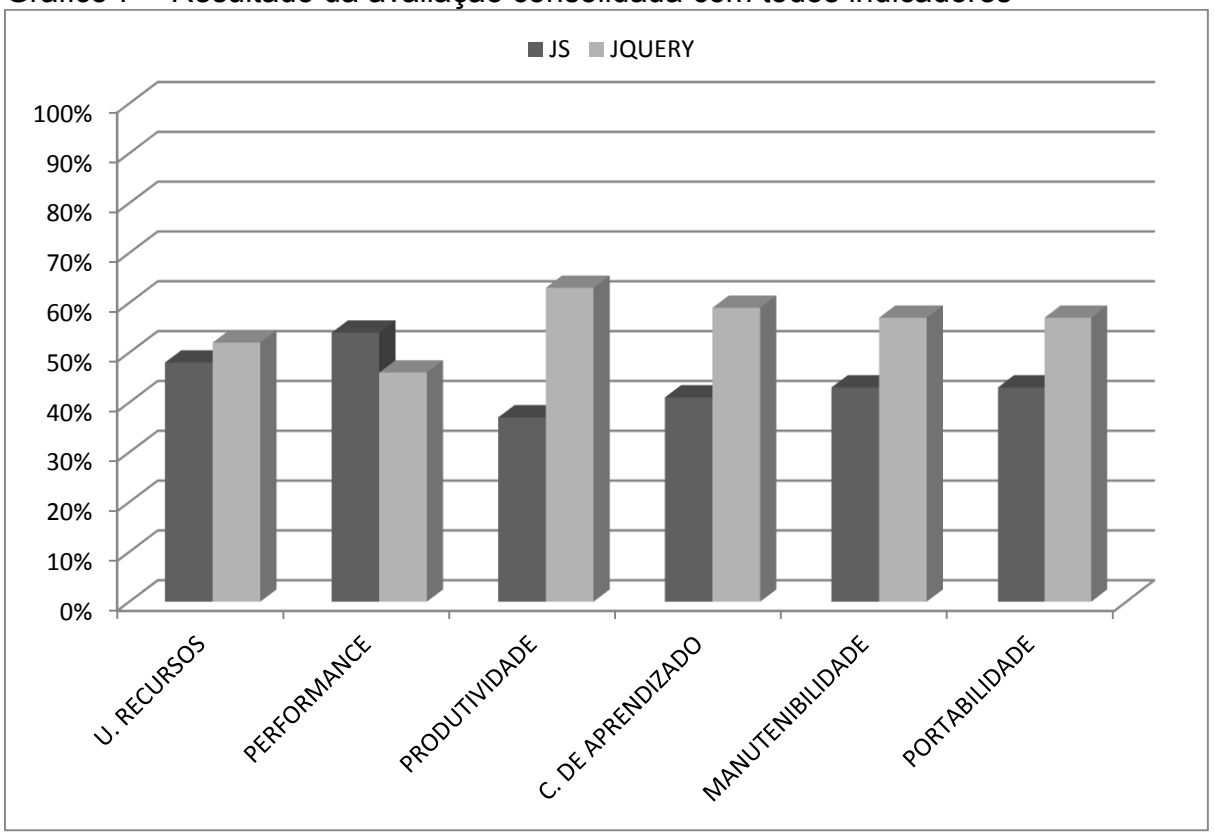
JavaScript	JQUERY
Codificação exclusiva para determinado navegador	Atualizações de versão com remoção de recursos e descontinuidade
Cross browser	Baixa performance
Gera conflito com CSS	Baixo desempenho
Grande de quantidade de código produzida para simples recursos	Conflitos na utilização de plug-ins
Interoperabilidade	Legibilidade
Legibilidade	Manutenção
Manutenção	Portabilidade
Operação limitada pelo usuário	Segurança
Portabilidade	Sintaxe
Produtividade	Validação
Rastreabilidade de erros	
Segurança	
Sintaxe	
Tempo gasto com codificação	

Fonte: Questionário de pesquisa

4.2.9 Consolidação dos Indicadores da Pesquisa

A consolidação de todos indicadores avaliados proporciona uma visão mais ampla do estudo desenvolvido. Esta subseção apresenta resultados de todos os indicadores de forma consolidada. Os resultados aqui apresentados foram baseados nos resultados apresentados pelas subseções anteriores que por sua vez apresentam os resultados individualmente dos indicadores.

Gráfico 7 – Resultado da avaliação consolidada com todos indicadores



Fonte: Própria

Os resultados obtidos demostram que:

- Dos indicadores avaliados o JQUERY apenas perde no indicador de performance.
- O JQUERY é bem superior no indicador de produtividade.
- O JavaScript é superior ao JQUERY apenas no indicador de performance.
- Para o indicador de curva de aprendizado, o JQUERY possui uma boa superioridade de 18,00%.
- Mesmo vencendo no indicador de utilização de recursos o JQUERY vence com superioridade abaixo de 5,00%.

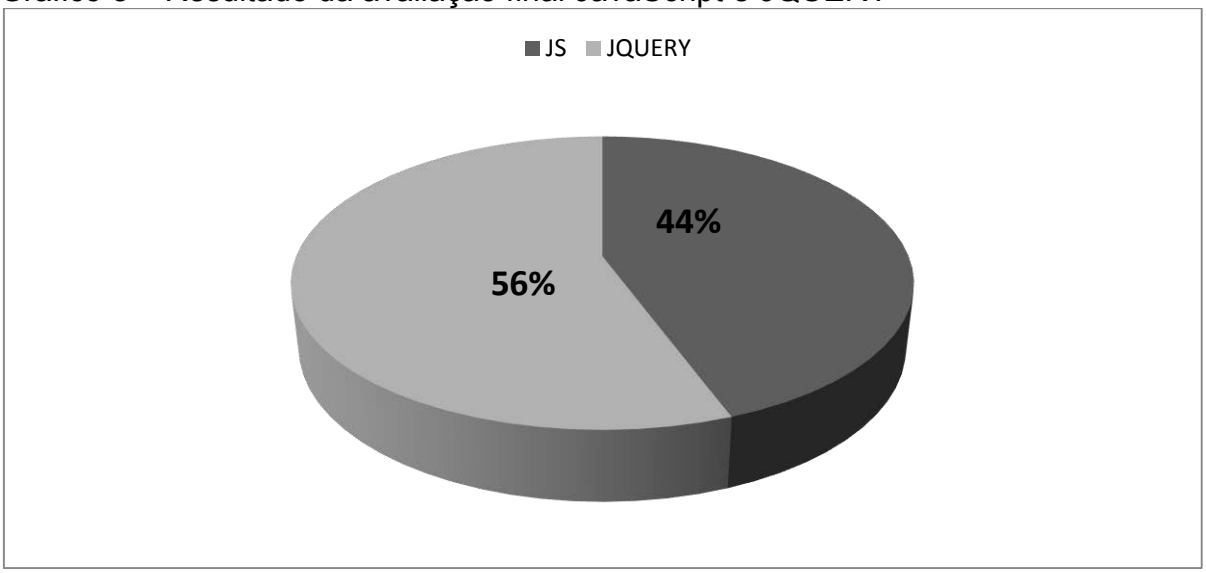
De uma maneira geral, o resultado da pesquisa permite dizer que o JQUERY vence o JavaScript em quase todos os indicadores, o que garante sua superioridade em adoção do framework para codificação em JavaScript.

O resultado geral coloca o JQUERY como sendo a melhor opção para o desenvolvimento de páginas web utilizando a tecnologia JavaScript. JQUERY obteve 56,00% e o JavaScript 44,00% das avaliações dos indicadores medidos

nesta pesquisa, o JQUERY vence o JavaScript com diferença de 12,00% na avaliação final do comparativo.

A metodologia utilizada para confecção do gráfico 8, foi a realização do somatório de todas as porcentagens resultantes dos gráficos de “pizza” contidos na subseção 4.2. O somatório foi realizado separados por tecnologias, ou seja, foi realizado o somatório das avaliações do JavaScript e JQUERY separadamente. Após realizado o somatório, foi aplicado a porcentagem para confecção do gráfico 8.

Gráfico 8 – Resultado da avaliação final JavaScript e JQUERY



Fonte: Própria

5 CONSIDERAÇÕES FINAIS

As referências literárias e as documentações oficiais das tecnologias avaliadas neste trabalho, principalmente as referências para o framework JQUERY, assumem a codificação em JavaScript utilizando o framework JQUERY como a maneira mais simples e produtiva para o desenvolvimento de páginas web.

Baseando-se em referências literárias e comparando na prática os recursos contidos no framework, o JQUERY apresentou características que o torna mais produtivo e mais simples se comparado ao JavaScript nativo.

Avaliando os resultados obtidos da pesquisa conduzida neste trabalho, o JQUERY venceu cinco dos seis indicadores avaliados. Os indicadores avaliados na pesquisa foram: utilização de recursos, performance, produtividade, curva de aprendizado, manutenibilidade e portabilidade.

Na maioria dos indicadores quantificados, o JQUERY, em média não venceu com uma diferenças extremamente superiores. Os principais indicadores que o JQUERY consagrou-se superior na pesquisa foram em produtividade com 63,00% e curva de aprendizado com 59,00%.

O JavaScript nativo venceu apenas um dos seis indicadores avaliados, o indicador que tornou o JavaScript campeão no comparativo foi no comparativo de performance, JavaScript obteve 54,00% contra 46,00% para o JQUERY.

De maneira geral, comparando o que dizem as referências literárias com a referência de mercado, que foram obtidas dos questionários, é possível concluir que JQUERY é superior ao JavaScript. Isso quer dizer que tudo que feito com JQUERY é possível fazer com JavaScript, simplesmente pelo fato do JQUERY ser um framework escrito em JavaScript. Os pontos que torna o JQUERY superior é a produtividade, compatibilidade e simplicidade.

Mesmo vencendo o comparativo que avalia o indicador de portabilidade, o JQUERY ainda possui algumas deficiências como falta de suporte a determinados navegadores e descontinuidade de recursos nas versões subsequentes. O JavaScript por sua vez, possui recursos para que seja solucionado, porém requer que o programador e/ou desenvolvedor crie recursos exclusivos para determinados

navegadores e soluções que resolva problemas de padrões proprietários entre navegador do mesmo produtor, como exemplo o navegador Internet Explorer.

Para a avaliação do indicador de performance, o JavaScript venceu pelo fato de não precisar de código fontes de terceiros, já o JQUERY utiliza um arquivo de código fonte que contém a codificação necessária para seu funcionamento. Ao carregar este arquivo também são carregados todos os recursos contidos no framework que na maioria das vezes não são utilizados. Mesmo perdendo no quesito performance, ainda assim existem recursos que possibilitam aumentar a performance do framework JQUERY.

Na avaliação para o indicador de produtividade o JQUERY consagra-se superior pelo recurso de abstração da codificação escrita com JavaScript. Isto significa dizer que JQUERY possui recursos que permite escrever pouco e obter um resultado em alto nível.

Avaliando o indicador de curva de aprendizado o JQUERY é o vencedor por implementar boas práticas, padrões de projeto e ser simples para o público utilizador. A pesquisa apontou que houve participante que afirma que aprendizado entre as tecnologias são equivalente e se complementam, ou seja, primeiro aprenderam JavaScript e com isso conseguiram aprender JQUERY facilmente pelo fato de já terem habilidades com JavaScript.

Na avaliação do indicador de recursos das tecnologias as pesquisas apontam que mesmo o JQUERY sendo o vencedor seus recursos não são tão superiores ao do JavaScript.

Avaliando o indicador de manutenibilidade o JQUERY é o vencedor, conforme demostra a pesquisa, no entanto houve participante (s) da pesquisa que consideram que as tecnologias comparadas possuem o mesmo nível de manutenibilidade pois cada uma possuem suas particularidades. Outra situação apontada pela pesquisa é que a manutenibilidade não está ligada apenas a linguagem e sim como está organizada a codificação e a documentação.

Pode-se concluir que das tecnologias avaliadas neste trabalho, segundo o resultado das pesquisas o JQUERY é melhor opção comparada ao JavaScript. O JQUERY só não será melhor opção caso critério principal seja a utilização de uma tecnologia

com o maior índice de performance, neste caso a recomendação sempre será o JavaScript.

Este estudo limitou-se em avaliar uma tecnologia nativa comparando com um framework, que é uma evolução dos recursos nativos. A pesquisa apresenta resultados que colaboram no processo decisório para utilização das tecnologias avalidas. Como sugestão para desenvolvimento de futuras pesquisas seria ampliar a pesquisa e comparar apenas frameworks para JavaScript mais utilizados pelo mercado. Um exemplo seria comparar os frameworks JQUERY e DOJO, incluindo mais indicadores comuns aos frameworks avaliados e filtrar a experiência dos entrevistados.

REFERÊNCIAS

ALENCAR, M.S. **Engenharia de redes de computadores**. São Paulo: Editora Érica, 2012.

ALLEN, R; LO, N; BROWN, S. **Zend Framework em Ação**. Rio de Janeiro: Alta Books, 2009.

BASHAM, B; SIERRA, K; BATES, B. **Use a cabeça! Servlets & JSP.** 2ª ed. Rio de Janeiro: Alta Books, 2008.

BIBEAULT, B; KATZ, Y. **JQUERY in Action.** Greenwich: Manning Publications, 2008.

BOCK, A.M.B; FURTADO, O; TEIXEIRA, M.L.T. **Psicologia**: Uma Introdução ao Estudo de Psicologia. 13ª ed. São Paulo: Saraiva, 2002.

BRANSFORD, J.D; BROWN, A.L; COCKING, R.R. **Como as pessoas aprendem:** Cérebro, Mente, Experiência e Escola. São Paulo: Editoria Senac São Paulo, 2007.

BRASIL. **Lei de Software nº 9.609, de 19 de fevereiro de 1998.** Disponível em: <http://www.planalto.gov.br/ccivil_03/leis/l9609.htm%2011/10/12>. Acesso em: 27 mai. de 2012.

CARDOSO, F. C; RAMIREZ, Alejandro Rafael Garcia. **Servidores de internet embarcada**. Rio de Janeiro: Ciência Moderna, 2009.

CONVERSE, T; PARK, J. **PHP**: A Bíblia. 2ª ed. Rio de Janeiro: Elsevier, 2003.

DALL'OGLIO, P. **PHP**: Programando com orientação a objetos. São Paulo: Novatec Editora, 2007.

DEITEL, H.M; DEITEL P.J; NIETO, T.R. **Internet & Word Wide Web:** Como Programar. 2ª ed. Porto Alegre: Bookman, 2003.

DEITEL, H.M; DEITEL P.J; J. Listfield et al. **C#**: Como Programar. São Paulo: Pearson Education, 2003.

DEITEL, P.J; DEITEL, H.M. **Ajax, Rich Internet Applications e desenvolvimento Web para Programadores.** São Paulo: Pearson Prentice Hall, 2008.

DICIONÁRIO PRIBERAM DE LÍNGUA PORTUGUESA. **Portabilidade.** [s.d]. Disponível em: <http://www.priberam.pt/DLPO/portabilidade>. Acesso em: 16 de out. 2013.

DUCKETT, J. **Beginning HTML, XHTML, CSS, and JavaScript.** Indiana: Wiley Publishing, Inc., 2010.

ENGHOLM JR, Hélio. **Engenharia de Software na Prática**. São Paulo: Novatec Editoria, 2010.

FERREIRA, A. B. H. **Mini Aurélio:** O minidicionário da língua portuguesa. Rio de Janeiro: Nova Fronteira, 2000.

FREEMAN, Elisabeth; FREEMAN, Eric. **Use a Cabeça:** HTML com CSS E XHTML. 2 ed, Rio de Janeiro: Alta Books, 2008.

GESAC. **O que é Software Livre**. [s.d]. Disponível em: <http://www.gesac.gov.br/o-que-e-software-livre>. Acesso em: 08 abr. 2013.

GOOGLE DEVELOPERS. **Evaluating network performance.** [s.d]. Disponível em: <https://developers.google.com/chrome-developer-tools/docs/network?hl=pt-BR>. Acesso em: 29 ago. 2013.

GOODMAN, Danny. **JavaScript**: a bíblia. Rio de Janeiro: Elsevier, 2001.

GRIMES, Galen A. **10 minutos para aprender internet**. 2 ed. São Paulo: Berkeley, 2000.

GUIZZO, Érico Marui. **Internet:** O que é, o que oferece e como conectar-se. São Paulo: Ática, 2000.

ISO/IEC 9126-1**. Engenharia de software - Qualidade de produto**: Parte 1: Modelo de qualidade, 2003.

JONES, C. Applied Software Measurement: Assuring Productivity and Quality. 2. ed. McGraw-Hill: 1996

JQUERY. **JQUERY API**. [s.d]. Disponível em: <http://JQUERY.com>; Acesso em: 04 jun. 2013.

KUROSE, James; F.ROSS, Keith W. **Redes de computadores e a internet**: uma abordagem top-down. 5 ed. São Paulo: Addison Wesley, 2011.

LAUDON, K. C.; LAUDON, J.P. **Sistemas de Informação Gerenciais**: Administrando a empresa digital. 5 ed. São Paulo: Prentice Hall, 2004.

LEMAY, Laura. **Aprenda a criar páginas web com HTML e XHTML em 21 dias**. São Paulo: Pearson Education do Brasil, 2002.

LISBOA, F. G. S. **Zend Framework**: componentes poderosos para PHP. São Paulo: Novatec Editora, 2009.

MACEDO, M. S. **Construindo sites adotando padrões web**. Rio de Janeiro: Editora Ciência Moderna, 2004.

MARCONI, M. de A. e LAKATOS, E. M. **Fundamentos de metodologia científica**. 5a ed. São Paulo: Atlas, 2003.

MCFARLAND, D. S. **JavaScript & JQUERY**: The Missing Manual. 2 ed. Sebastopol: O'Reilly: 2011.

MCLAUGHLIN, B. **Use a cabeça Ajax.** Rio de Janeiro: Alta Books, 2006.

MELO, A. A.; NASCIMENTO, M. G. F. **PHP Profissional**: Aprenda a desenvolver sistemas profissionais orientado ao objetos com padrões de projeto. São Paulo: Novatec Editora, 2007.

MEMÓRIA, F. **Design para internet:** projetando a experiência perfeita. Rio de Janeiro: Elsevier, 2006.

MORESI, Eduardo. **Metodologia de Pesquisa**, Universidade Católica de Brasília, 2003. Disponível em <http://www.inf.ufes.br/~falbo/files/MetodologiaPesquisa-Moresi2003.pdf>. Acesso em 16 Jun. 2013.

MORRISON, M. **Use a Cabeça**: JavaScript. Rio de Janeiro: Alta Books, 2008.

MOZILLA DEVELOPER NETWORK. **Introdução ao HTML**. [s.d]. Disponível em <https://developer.mozilla.org/pt-BR/docs/HTML/Introduction>. Acesso em: 17 abr. 2013.

_____. **Aprenda CSS**. [s.d]. Disponível em: <https://developer.mozilla.org/pt-BR/learn/css>. Acesso em: 17 abr. 2013.

_____. **CSS 2013**. [s.d]. Disponível em: <https://developer.mozilla.org/pt-BR/docs/CSS>. Acesso em: 17 abr. 2013.

NIELSEN, Jakob; LORANGE, Hoa. **Usabilidade na Web**: Projetando Websites com qualidade. Rio de Janeiro: Elsevier, 2007.

PEREIRA, M.G; REHDER, W. S. **Guia Prático Webdesigner**. São Paulo: Editora Viena, 2003.

PFAFFENBERGER, B; SCHAFER M.S; WHITE, C et al. HTML, XHTML and CSS BIBLE. 3.ed. Indianapolis: Wiley Publishing Inc, 2004.

POWELL, T.A. **HTML & CSS**: The complete reference 5.ed. The McGraw-Hill, 2010.

PRESSMAN, R.S. **Engenharia de Software.** McGraw-Hill, 2006.

SILVA, C.C; VARGAS, E.S. **HTML**: Construindo a Internet. São Paulo: Editora Viena, 2007.

SILVA, M. S. **JQUERY**: A Biblioteca do Programador JavaScript. São Paulo: Novatec, 2008.

_____. **JavaScript**: Guia do Programador. São Paulo: Novatec, 2010.

TAROUCO, L.M.R. **Protocolos Cliente/Servidor**. [s.d]. Disponível em <http://penta2.ufrgs.br/redes296/cliente_ser/tutorial.htm>. Acesso em: 30 jun 2013.

YIN, Roberto K. **Estudo de Casos:** Planejamento e métodos. 2.ed. São Paulo: Bookman, 2001.

W3C. **HTML & CSS**. [s.d]. Disponível em: <http://www.w3.org/standards/webdesign/htmlcss>. Acesso em: 16 abr. 2013.

APÊNDICE A – Questionário de Pesquisa

Questionário para avaliação da utilização das tecnologias JavaScript (nativo) e JQUERY (framework) no mercado de trabalho

Este questionário tem como objetivo a coleta de informações para a pesquisa utilizada no Trabalho de Conclusão de Curso (TCC) do aluno Silas Stoffel a ser apresentado na Faculdade Católica Salesiana como requisito obrigatório para obtenção do título de Bacharel em Sistemas de Informação.

IDENTIFICAÇÃO DO RESPONDENTE

Nome: _____

Cargo: _____

Empresa: _____

INSTRUÇÕES

Avalie as questões a seguir utilizando notas de zero (0) a cinco (5). Onde **ZERO** (0) é exclusivo para questões onde o entrevistado **NÃO POSSUA CONHECIMENTO** para responder e/ou avaliar e **CINCO** (5) representa a **NOTA MÁXIMA** para a questão.

QUESTINÁRIO FECHADO

1 – No quesito de quantidade de utilização dos recursos das tecnologias *JavaScript, JQUERY* ou outras, como você as avalia na **UTILIZAÇÃO** dos **RECURSOS DISPONÍVEIS** em cada tecnologia no ambiente coorporativo.

UTILIZAÇÃO DOS RECURSOS DISPONÍVEIS nesta questão pode ser entendida como quanto dos recursos disponíveis que a tecnologia possui e que você faz utilização.

JavaScript	JQUERY	Outro
( ) 0	( ) 0	( ) 0
( ) 1	( ) 1	( ) 1
( ) 2	( ) 2	( ) 2
( ) 3	( ) 3	( ) 3

( ) 4	( ) 4	( ) 4
( ) 5	( ) 5	( ) 5

Comentário (opcional)

_____
_____
_____
_____

2 – No quesito de **OTIMIZAÇÃO** e **PERFORMANCE** das tecnologias *JavaScript, JQUERY* ou outras, como você as avalia.

PERFORMANCE pode ser entendido como processamentos com o tempo de resposta adequado às necessidades do usuário (ENGHOLM JR, 2010).

JavaScript	JQUERY	Outro
( ) 0	( ) 0	( ) 0
( ) 1	( ) 1	( ) 1
( ) 2	( ) 2	( ) 2
( ) 3	( ) 3	( ) 3
( ) 4	( ) 4	( ) 4
( ) 5	( ) 5	( ) 5

Comentário (opcional)

_____
_____
_____
_____

3 – No quesito de **PRODUTIVIDADE** das tecnologias *JavaScript, JQUERY* ou outras como você as avalia no desenvolvimento dos projetos.

PRODUTIVIDADE, de um modo geral, é a relação entre a quantidade de bens ou serviços produzidos e a despesa ou trabalho necessário para produzi-los (Jones, 1996).

JavaScript	JQUERY	Outro

( ) 0	( ) 0	( ) 0
( ) 1	( ) 1	( ) 1
( ) 2	( ) 2	( ) 2
( ) 3	( ) 3	( ) 3
( ) 4	( ) 4	( ) 4
( ) 5	( ) 5	( ) 5

Comentário (opcional)

_____

_____

_____

_____

4 – No quesito de **CURVA DE APRENDIZADO** das tecnologias *JavaScript, JQUERY* ou outras, como você as avalia a trajetória traçada para adquirir o conhecimento para utilização das tecnologias no **MENOR TEMPO POSSÌVEL**.

Para esta questão, **CURVA DE APRENDIZADO** deve ser entendida o esforço aplicado para aprender as tecnologias em função do tempo. O esforço pode ser entendido como horas de estudos, testes, a quantidade de tentativas para se adquirir a capacidade de usá-las adequadamente.

Avalie esta questão utilizando notas de zero a cinco, onde cinco representa o máximo de conhecimento adquirido no menor tempo que podem ser aplicados.

JavaScript	JQUERY	Outro
( ) 0	( ) 0	( ) 0
( ) 1	( ) 1	( ) 1
( ) 2	( ) 2	( ) 2
( ) 3	( ) 3	( ) 3
( ) 4	( ) 4	( ) 4
( ) 5	( ) 5	( ) 5

Comentário (opcional)

_____
_____
_____
_____

5 – No quesito relacionado **MANUTENABILIDADE** das tecnologias *JavaScript, JQUERY* ou outras, como você avalia a facilidade ou a não facilidade das tecnologias em aplicar manutenção nos projeto.

Entende-se como **MANUTENABILIDADE** a facilidade, precisão, segurança e economia na execução de ações de manutenção (BLANCHARD, 1992). Na engenharia de software **MANUTENABILIDADE** é refere-se à facilidade de um software ser modificado a fim de corrigir os defeitos, adequar aos novos requisitos ou ao novo ambiente.

JavaScript	JQUERY	Outro
( ) 0	( ) 0	( ) 0
( ) 1	( ) 1	( ) 1
( ) 2	( ) 2	( ) 2
( ) 3	( ) 3	( ) 3
( ) 4	( ) 4	( ) 4
( ) 5	( ) 5	( ) 5

Comentário (opcional)

_____
_____
_____
_____

6 – No quesito relacionado **PORTABILIDADE** das tecnologias *JavaScript, JQUERY* ou outras, como você as avalia.

PORTABILIDADE, em um ambiente computacional pode ser entendida como a qualidade de um componente ou de um programa informático que o torna capaz de ser usado em diferentes sistemas e computadores. (DICIONÁRIO PRIBERAM DA LÍNGUA PORTUGUESA) em definição para informática.

JavaScript	JQUERY	Outro
( ) 0	( ) 0	( ) 0
( ) 1	( ) 1	( ) 1
( ) 2	( ) 2	( ) 2
( ) 3	( ) 3	( ) 3
( ) 4	( ) 4	( ) 4
( ) 5	( ) 5	( ) 5

Comentário (opcional)

_____

_____

_____

_____

QUESTINÁRIO ABERTO

Mediante a experiência na utilização das tecnologias *JavaScript* e *JQUERY* o que você pode mensurar de pontos fortes e fracos.

Pontos fortes

- JavaScript

I) _____

||)_____

III)_____

- JQUERY

I) _____

||)_____

III)_____

Pontos fracos

- JavaScript

I) _____

||)_____

III)_____

- JQUERY

I) _____

||)_____

III)_____

Autorização

Eu _____ autorizo o aluno Silas Stoffel reproduzir e fazer uso dos dados existentes neste documento para fins acadêmicos, inclusive citando o meu nome e o da empresa para qual trabalho quando assim lhe convier.

_____

Respondente

APÊNDICE B – Estrutura HTML da página para comparativo de produtividade

```
<!DOCTYPE html>
<html>
    <head>
        <title>Produtividade</title>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <style type="text/css">
            body {margin: 0px;padding: 0px;background-color: white;font-size: 12px;color: #333;font-family: arial,verdana sans-serif;}
            .todo-corpo {margin: 0px auto; width: 900px;}.linha {width:799px;margin: 0px auto;padding: 3px;}
            label {font-weight: bold;font-size: 14px;}
        </style>
    </head>
    <body>
        <div class="todo-corpo">
            <p><a href="javascript:;" id="marcarTodos">Marcar Todas</a></p>
            <div class="linha"><p>Disciplinas</p></div>
            <div class="linha">
                <input name="opcoes" type="checkbox"  value="Engenharia de Software">Engenharia de Software<br/>
                <input name="opcoes" type="checkbox"  value="Cálculo">Cálculo<br/>
                <input name="opcoes" type="checkbox"  value="Algebra Linear">Algebra Linar<br/>
                <input name="opcoes" type="checkbox"  value="Estatística">Estatística<br/>
                <input name="opcoes" type="checkbox"  value="Programação I">Programação I<br/>
                <input name="opcoes" type="checkbox"  value="Programação II">Programação II<br/>
                <input name="opcoes" type="checkbox"  value="Programação III">Programação III<br/>
                <input name="opcoes" type="checkbox"  value="Sistemas Distribuidos">Sistemas Distribuidos<br/>
            </div>
        </div>
    </body>
</html>
```

APÊNDICE C – Metodologia de confecção dos gráficos

Para quantificar a pesquisa são utilizados os seguintes procedimentos:

1. Cálculo do total de notas atribuídas para a tecnologia JavaScript (NJS)

NJS = (1 * quantidade de nota 1) + (2 * quantidade de nota 2) + (3 * quantidade de nota 3) + (4 * quantidade de nota 4) + (5 * quantidade de nota 5).

2. Cálculo do total de notas atribuídas para a tecnologia JQUERY (NJQ)

NJQ = (1 * quantidade de nota 1) + (2 * quantidade de nota 2) + (3 * quantidade de nota 3) + (4 * quantidade de nota 4) + (5 * quantidade de nota 5).

3. Cálculo do total de notas (TN)

TN = NJS + NJQ

Baseado na tabela abaixo, é possível gerar dados para confecção do gráfico de barras e criação da consolidação da pesquisa que é representado pelo gráfico de pizza.

NOTA	0	1	2	3	4	5	TOTAL
JAVA SCRIPT	0	0	1	2	3	5	45
JQUERY	0	1	2	1	4	3	39

O gráfico em barras que representa a avaliação individual por notas e as tecnologias comparadas. O gráfico em pizza representa a consolidação do gráfico de barras utilizando a seguinte metodologia:

NJS = (1 * 0) + (2 * 1) + (3 * 2) + (4 * 3) + (5 * 5)
NJS = 0 + 2 + 6 + 12 + 25
NJS = 45

NJQ = (1 * 1) + (2 * 2) + (3 * 1) + (4 * 4) + (5 * 3)
NJQ = 1 + 4 + 3 + 16 + 15
NJQ = 39

TN = NJS + NJQ
TN = 45 + 39
TN = 84

O gráfico de pizza é representado em porcentagem para isto a seguinte metodologia é utilizada:

Porcentagem JavaScript (PJS).

PJS = (NJS / TN) * 100

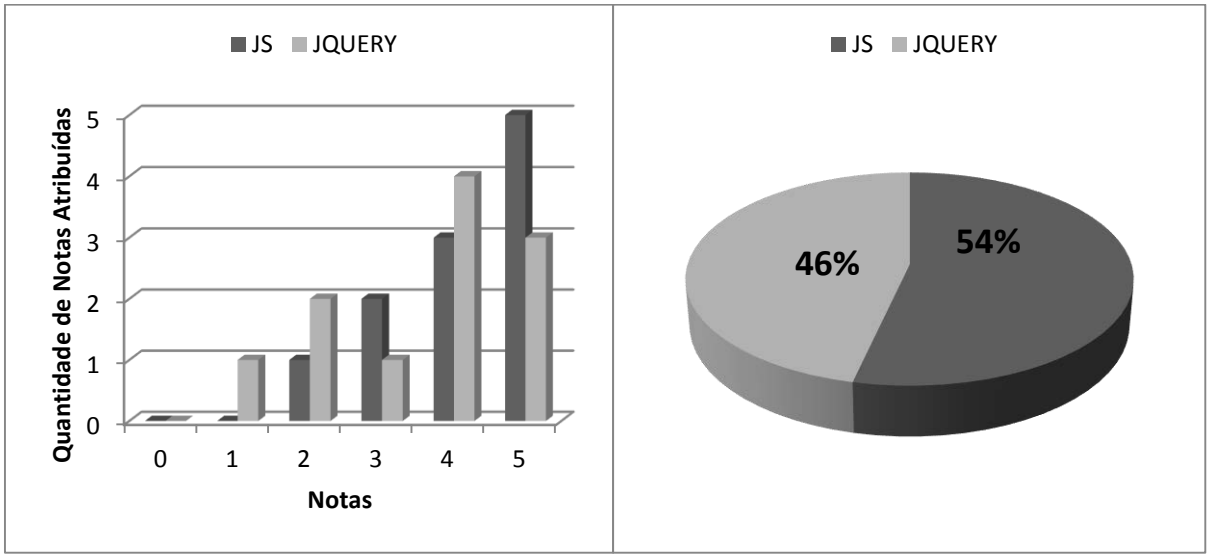
PJS = (45 / 84) * 100

PJS = 54

Porcentagem JQUERY (PJQ)

PJS = (NJQ / TN) * 100

PJS = (39 / 84) * 100

PJS = 46

Modelo de representação gráfica do trabalho: