

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ  
DEPARTAMENTO ACADÊMICO DE INFORMÁTICA  
ESPECIALIZAÇÃO EM TECNOLOGIA JAVA E DESENVOLVIMENTO PARA  
DISPOSITIVOS MÓVEIS**

**MARCELA ELOISA PROSTT**

**INTERFACE WEB UTILIZANDO DESIGN RESPONSIVO: UM ESTUDO  
DE CASO APLICADO A SMARTPHONES, TABLETS,  
COMPUTADORES E TELEVISORES**

**MONOGRAFIA DE ESPECIALIZAÇÃO**

**CURITIBA**

**2013**

**MARCELA ELOISA PROSTT**

**INTERFACE WEB UTILIZANDO DESIGN RESPONSIVO: UM ESTUDO  
DE CASO APLICADO A SMARTPHONES, TABLETS,  
COMPUTADORES E TELEVISORES**

Monografia de Especialização apresentada ao Departamento Acadêmico de Informática, da Universidade Tecnológica Federal do Paraná como requisito parcial para obtenção do título de Especialista em Tecnologia Java e Desenvolvimento para Dispositivos Móveis.

Orientadora: Prof.<sup>a</sup> Dr.<sup>a</sup> Marília A. Amaral

**CURITIBA**

**2013**

## **AGRADECIMENTOS**

Certamente estes parágrafos não irão atender a todas as pessoas que fizeram parte dessa importante fase de minha vida. Portanto, desde já peço desculpas àquelas que não estão presentes entre essas palavras, mas elas podem estar certas que fazem parte do meu pensamento e de minha gratidão.

Reverencio a Professa Dra. Marília A. Amaral pela sua dedicação e pela orientação deste trabalho e, por meio dele, eu me reporto a toda a comunidade da Universidade Tecnológica Federal do Paraná (UTFPR) pelo apoio incondicional.

Agradeço a Prefeitura Municipal de São José dos Pinhais pela oportunidade de trabalhar o tema dessa monografia no estudo de caso.

A todos os ex-colegas de trabalho da Prefeitura gostaria de externar minha satisfação de poder conviver com eles durante boa parte da realização deste estudo.

Gostaria de deixar registrado também, o meu reconhecimento à minha família, principalmente ao meu esposo Antonio Rodrigues Barros, pelo carinho, amor, compreensão, sem o apoio deles seria muito difícil vencer esse desafio.

## RESUMO

PROSTT, Marcela E. **Interface Web utilizando Design Responsivo**: um estudo de caso aplicado a *smartphones*, *tablets* e televisores. 2013. Monografia (Especialização em Tecnologias Java e Desenvolvimento para Dispositivos Móveis) - Universidade Tecnológica Federal do Paraná. Curitiba, 2013.

A crescente evolução da Internet móvel aquece cada vez mais o mercado dos dispositivos móveis, porém a carência no desenvolvimento de aplicações *mobile* para plataformas operacionais específicas e para aplicações web, também cresceu junto com este avanço tecnológico.

Uma experiência, não tão agradável, nesta era pós-PC é acessar uma aplicação web em um dispositivo com tela pequena e perceber que todo o seu conteúdo também é reduzido, ilegível e de difícil navegação.

Desenvolver uma aplicação web que seja adaptativa para dispositivos como *smartphone*, *tablets* e até mesmo televisores com acesso à Internet sem perder a navegabilidade, legibilidade do conteúdo e que seja de fácil manutenção, ou seja, que permita criar uma única aplicação para a enorme variedade de dispositivos existentes é algo ainda desconhecido para muitos desenvolvedores e gestores de TI e, este foi o foco deste trabalho, descobrir tecnologias de desenvolvimento web que auxiliem neste processo de adaptação de conteúdo e *layout* web convencionais para navegadores de dispositivos móveis de diversos tamanhos, focando em linguagens comumente utilizadas por programadores da área, buscando um método facilitador de manutenções sem o uso de grandes soluções extraordinárias ou frameworks complexos.

Um estudo de caso foi desenvolvido para aplicar os conceitos obtidos, onde teve resultados bastante satisfatórios em diversos dispositivos, inclusive em televisores, e fez-se uso de métodos e tecnologias simples e bem conhecidas por desenvolvedores web.

**Palavras-chave:** Dispositivos móveis. Design responsivo. Mobilidade. Interface Web.

## ABSTRACT

PROSTT, Marcela E. **Web Interface using Responsive Design**: a case study applied to smartphones, tablets, and televisions. 2013. Monografia (Especialização em Tecnologias Java e Desenvolvimento para Dispositivos Móveis) - Universidade Tecnológica Federal do Paraná. Curitiba, 2013.

The growing trend of mobile Internet increasingly heated market of mobile devices , though the deficiency in the development of mobile applications for specific operating platforms and web applications has also grown along with this technological advancement.

An experience, not so nice, in this post-PC is accessing a web application on a device with small screen and realize that all your content is also reduced, unreadable and difficult to navigate.

Develop a web application that is adaptive devices like smartphone, tablets and even televisions with Internet access without losing navigability, readability of the content and it is easy to maintain, in order to create a single application for the huge variety of existing devices is something still unknown to many developers and managers, and this was the focus of this work, discovering web development technologies that assist in the process of adapting content and layout for conventional web browsers of mobile devices of various sizes, focusing on languages commonly used by programmers of the area, seeking a method of facilitating maintenance without the use of extraordinary solutions large or complex frameworks.

A case study was developed to apply the concepts obtained, which had satisfactory results in various devices, including televisions, and made use of methods and technologies simple and well known to web developers.

**Keywords:** Mobile devices. Responsive design. Mobility. Web Interface.

## LISTA DE ILUSTRAÇÕES

Figura 1 - Comportamento de um <i>site</i> responsivo .....	18
Figura 2 - Resoluções de telas .....	19
Figura 3 - A Casa do Código - Página web acessada através de um computador convencional.....	20
Figura 4 - A Casa do Código - Página web acessada através de um dispositivo móvel .....	21
Figura 5 - Fork-CMS - Acesso à página web por diferentes dispositivos .....	21
Figura 6 - <i>Mr. Simon Collison</i> - Página web simples e responsiva .....	22
Figura 7 - Exemplo de atributos e valores em CSS .....	24
Figura 8 - Exemplo de estilo CSS para fontes, cores e tamanhos.....	25
Figura 9 - Medidas flexíveis no CSS .....	27
Figura 10 - Layout flexível - Proporção dos elementos .....	28
Figura 11 - Definindo um box-sizing .....	29
Figura 12 - Limitando um layout fluido com max-width .....	30
Figura 13 - Exemplo do funcionamento do viewport .....	31
Figura 14 - Comparativo entre Layout viewport e Visual viewport .....	32
Figura 15 - Testando a largura da página web com a mesma largura que tamanho real do dispositivo móvel .....	33
Figura 16 - Aplicação da metatag viewport para 320 pixels de largura.....	34
Figura 17 - Resultado do uso da metatag viewport com 320px de largura .....	34
Figura 18 - Aplicação da metatag viewport sem especificação da largura.....	35
Figura 19 - Aplicação da metatag viewport com valor de escala .....	35
Figura 20 - Parâmetros do viewport .....	35
Figura 21 - Viewport definido no CSS .....	36
Figura 22 - Exemplo de inserção de um imagem com largura fixa .....	36
Figura 23 - Implementação da classe figure no CSS .....	37

Figura 24 - Exemplo de uma imagem de grande resolução com largura fixa	.37
Figura 25 - Propriedade CSS Max-width	37
Figura 26 - Adequando a imagem ao elemento que a contem	38
Figura 27 - Exemplo de implementação do atributo srcset do HTML5	38
Figura 28 - Exemplo de implementação da função image-set() no CSS	39
Figura 29 - Layout fluido sem <i>Media Queries</i>	40
Figura 30 - Implementando o exemplo com media queries	40
Figura 31 - Resultado do uso de <i>Media Queries</i>	41
Figura 32 - Exemplo de breakpoint definido para largura máxima de 600px	42
Figura 33 - Implementação de breakpoint para medidas intermediárias	42
Figura 34 - Esboço de interface para smartphone na orientação retrato com menu retraído	47
Figura 35 - Esboço de interface para smartphone na orientação retrato com menu expandido	48
Figura 36 - Esboço de interface para tablet na orientação paisagem com menu retraído	48
Figura 37 - Esboço de interface para grandes resoluções	49
Figura 38 - Mockup para smartphone na orientação retrato com menu retraído	50
Figura 39 - Mockup para smartphone na orientação retrato com menu expandido	50
Figura 40 - Mockup para tablet na orientação paisagem com menu retraído	51
Figura 41 - Mockup para grandes resoluções	51
Figura 42 - Comportamento do projeto para cada tipo de dispositivo	52
Figura 43 - Implementação do CSS para identificar as resoluções	53
Figura 44 - Estrutura geral do projeto	54
Figura 45 - Comparativo do layout sem design responsivo para notebooks	55
Figura 46 - Comparativo do layout com design responsivo para notebooks	56
Figura 47 - Comparativo do layout sem design responsivo para tablets	57

Figura 48 - Comparativo do layout com design responsivo para tablets.....	58
Figura 49 - Comparativo do layout com design responsivo para tablets – menu expandido .....	59
Figura 50 - Comparativo do layout sem design responsivo para smartphones... .....	60
Figura 51 - Comparativo do layout com design responsivo para smartphones... .....	60
Figura 52 - Comparativo do layout com design responsivo para smartphones – menu expandido .....	61
Figura 53 - Comparativo do layout sem design responsivo para smartphones – orientação retrato .....	62
Figura 54 - Comparativo do layout com design responsivo para smartphones – orientação retrato .....	63
Figura 55 - Comparativo do layout com design responsivo para smartphones – orientação retrato, menu expandido .....	64
Figura 56 - Resultados obtidos a partir do navegador proprietário de uma smart TV .....	68
Figura 57 - Comparativo do layout sem viewport (esquerda) e com viewport (direita) .....	69
Figura 58 - Comparativo do menu sem viewport (esquerda) e com viewport (direita) .....	70
Figura 59 - Resultados obtidos em diversos dispositivos.....	71



## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>8</b>
1.1.	PROBLEMA	8
1.2.	OBJETIVO	8
1.2.1.	Objetivo Geral	8
1.2.2.	Objetivos Específicos	9
1.3.	JUSTIFICATIVA	9
1.4.	METODOLOGIA EMPREGADA	10
1.5.	ORGANIZAÇÃO DO TRABALHO	10
<b>2</b>	<b>GOVERNO ELETRÔNICO BRASILEIRO</b>	<b>11</b>
2.1.	CONHECENDO O e-GOV	11
2.1.1.	e-MAG	12
2.1.2.	e-PWB	13
<b>3</b>	<b>MOBILIDADE</b>	<b>15</b>
<b>4</b>	<b>WEB DESIGN RESPONSIVO</b>	<b>18</b>
4.1.	CSS e DESIGN RESPONSIVO	24
4.1.1.	Tipos de medidas CSS	25
4.2.	OS PILARES DO DESIGN RESPONSIVO	26
4.2.1.	Layout fluido	26
4.2.1.1.	100% Layout fluido	29
4.2.1.2.	O viewport	30
4.2.2.	Imagens e Recursos flexíveis	36
4.2.3.	Media Queries	39
4.3.	MOBILE FIRST	43
<b>5</b>	<b>ESTUDO DE CASO</b>	<b>45</b>
5.1.	METODOLOGIA	45
5.2.	CONTEXTO	45
5.3.	PROJETO DE INTERFACE	46
5.3.1.	Visão interativa de baixa fidelidade	46
5.3.2.	Projeto mockup	49
5.4.	IMPLEMENTAÇÃO	51
5.5.	DEMONSTRAÇÃO	55

<b>6</b>	<b>RESULTADOS OBTIDOS.....</b>	<b>65</b>
<b>7</b>	<b>CONSIDERAÇÕES FINAIS .....</b>	<b>72</b>

# 1 INTRODUÇÃO

## 1.1. PROBLEMA

Antes do crescimento do mercado de dispositivos móveis, grande parte da Internet era acessada utilizando computadores com resoluções muito semelhantes.

Hoje, com a gama de outros dispositivos que também podem acessar a Internet como *smartphones*, *tablets*, *netbooks*, televisores, etc., estas resoluções perderam suas similaridades criando uma enorme variedade de tamanhos e gêneros de telas.

A experiência de acessar uma página web não adaptativa nestes dispositivos, que podem ser tão pequenos em comparação aos computadores convencionais, tornou-se algo desagradável e até mesmo frustrante para seus usuários, pois a falta de um *layout* adequado para estes aparelhos implica na má visualização, navegação e até mesmo no desempenho da aplicação.

Na maioria dos casos os sites apresentam o mesmo comportamento em navegadores dos computadores de mesa quanto em navegadores dos dispositivos móveis ao invés de terem características específicas para cada tipo de dispositivo. Projetos web devem ser idealizados desde a sua concepção para serem flexíveis e exibidos de forma adequada, o problema não é apenas uma questão somente técnica, mas também conceitual.

Neste contexto, a questão seguinte é levantada: Como adaptar *layouts* e conteúdos web para estes dispositivos que tem gêneros e tamanhos diferentes usando metodologias e tecnologias disponíveis no mercado?

## 1.2. OBJETIVO

### 1.2.1. Objetivo Geral

Pesquisar e aplicar métodos de desenvolver aplicações web que sejam adaptáveis a diversos tamanhos de dispositivos móveis proporcionando boa apresentação e navegação do conteúdo ao usuário. Contudo, o desenvolvimento deve apresentar uma única versão para todos estes dispositivos, a fim de garantir a

manutibilidade da aplicação e que também tenha comportamento adequado em computadores com telas de tamanhos convencionais.

### 1.2.2. Objetivos Específicos

1. Aplicar tecnologias e metodologias web para promover desenvolvimento de *layouts* web que sejam flexíveis e bem apresentados em diversos artefatos tecnológicos a fim de proporcionar melhor legibilidade, navegabilidade e manutibilidade do conteúdo.
2. Realizar pesquisa exploratória para identificar os principais recursos tecnológicos e metodológicos empregados no desenvolvimento de adaptações de interfaces web para dispositivos móveis;
3. Focar em linguagens comumente utilizadas por profissionais da área de desenvolvimento, buscando um método facilitador de manutenções.
4. Selecionar o resultado de maior relevância obtido na pesquisa exploratória e aplicá-lo na prática por meio de um estudo de caso;

### 1.3. JUSTIFICATIVA

A demanda de telefonia móvel pela sociedade tem crescido nos últimos anos. Segundo dados estatísticos levantados pela Agência Nacional de Telecomunicações (Anatel) em fevereiro de 2013, o ano de 2012 registrou um crescimento de 8,08% de Serviços Móveis Pessoal (SMP) em relação a 2011 (ANATEL, 2013, p. 4). A tecnologia que teve maior crescimento percentual em 2012 foi a *Wideband Code Division Multiple Access* (W-CDMA) com 57,8% e, esse fato deve-se à massificação do uso da banda larga terceira geração (3G) pela população (ANATEL, 2013, p. 5).

Pode-se notar que é cada vez mais comum o uso da Internet nos celulares, *tablets* e até nas *smart* TVs. Em qualquer caso desses, a situação ideal é que as páginas web desenvolvidas, sejam bem apresentadas conforme o dispositivo que a esta acessando no momento. Adaptar páginas web para que haja um funcionamento correto, legível e navegável para a variedade de dispositivos móveis existentes, tornou-se imprescindível nos dias de hoje.

Ainda é grande a carência de sites que se adaptem a estes dispositivos, até

mesmo grandes portais não oferecem o comportamento adequado de seu conteúdo na web *mobile* e, talvez a dificuldade esteja apenas na falta de informação ou até mesmo por questões culturais.

#### 1.4. METODOLOGIA EMPREGADA

O presente trabalho compõe-se basicamente de estudos, testes e aplicação das metodologias e técnicas de *design* responsivo, *viewport* e *mobile first*, agregando-as em um protótipo destinado a dispositivos móveis e televisores.

Para identificar essas tecnologias e conceitos, foi realizada uma pesquisa bibliográfica relacionada à adaptabilidade de páginas web, mobilidade, dispositivos móveis e resoluções de telas.

Um protótipo foi desenvolvido aplicando o conjunto destas tecnologias e ao final foi elaborado um comparativo de como seria o protótipo sem o uso destas técnicas.

#### 1.5. ORGANIZAÇÃO DO TRABALHO

Nesta Seção mostrou um introdutório deste trabalho.

Na Seção 2, são abordados aspectos de como o Governo Eletrônico desenvolve *sites* e portais governamentais.

A Seção 3 é destinada a comunicação móvel, como a mobilidade tecnológica afetou a rotina das pessoas.

No entanto, a Seção 4 é o assunto principal do trabalho, *design* responsivo, técnicas e conceitos utilizados para planejar e desenvolver *sites* web para várias resoluções de telas sem a necessidade de criar múltiplas versões para cada dispositivo.

Na Seção 5 é apontado como as técnicas foram implementadas, como a interface foi desenvolvida e como seria o protótipo sem o *design* responsivo.

Já na Seção 6, são apontados os resultados obtidos.

E por fim, a Seção 7 traz algumas considerações finais sobre o trabalho como um todo.

## 2 GOVERNO ELETRÔNICO BRASILEIRO

Nesta Seção, o tema será abordado com o intuito de compartilhar os padrões adotados pelo Governo Eletrônico Brasileiro (e-GOV) no desenvolvimento de *sítes* e portais governamentais, onde envolvem principalmente conceitos como usabilidade, redação, codificação e administração, itens envolvidos indiretamente com o trabalho apresentado.

### 2.1. CONHECENDO O E-GOV

O e-Gov no Brasil consiste do uso da tecnologia de informação para prestação de serviços públicos que ganhou espaço na administração pública federal em março de 2000 (e-Gov 2000), ano que foi criado o grupo de trabalho interministerial com a finalidade de examinar e propor políticas, diretrizes e normas relacionadas às novas formas eletrônicas de interação (ROVER, 2010), através do Decreto Presidencial de 3 de abril de 2000<sup>1</sup> (e-Gov, 2000).

A finalidade do programa é democratizar o acesso à informação, ampliar discussões, dinamizar a prestação de serviços públicos, transformar as relações do Governo com os cidadãos, empresas e também entre os órgãos do próprio governo (e-Gov, 2000).

Desde então, o e-Gov vem desenvolvendo programas com a utilização das modernas Tecnologias de Informação e Comunicação (TICs) para democratizar o acesso à informação, ampliar discussões e dinamizar a prestação de serviços públicos com foco na eficiência e efetividade das funções governamentais (e-Gov, 2000).

O Programa segue um conjunto de diretrizes e legislações como: Decretos, Instruções Normativas, Leis, Medida Provisória, Portarias, Regimento Interno e Resoluções. O programa também conta com um Comitê Executivo e oito Comitês Técnicos responsáveis pelo desenvolvimento das políticas e ações definidas nos princípios e diretrizes estabelecidas para toda a Administração Pública Federal (e-

---

<sup>1</sup> Decreto Presidencial de 3 de abril de 2000. Disponível em: <<http://www.governoeletronico.gov.br/o-gov.br/historico>> Acesso em: 19 mai. 2013

<sup>2</sup> Portal Govbr pode ser acessado em: <http://www.governoeletronico.gov.br/o-gov.br>

Gov, 2000).

Para ampliar a transparência e o controle social sobre as ações do Governo Federal, foi criada a Secretaria de Logística e Tecnologia da Informação (SLTI), onde sua função é planejar, coordenar, supervisionar e orientar, normativamente, as atividades do sistema de administração dos recursos de tecnologia da informação (e-Gov, 2000).

O Governo Eletrônico Brasileiro possui *sites* e portais com informações relacionadas ao Programa:

Gov.br: portal oficial do programa de Governo Eletrônico Brasileiro. Temas como acessibilidade, banda larga, inclusão digital, compras eletrônicas, etc., ocupam local de destaque no *site*.

Dados.gov.br: sistema de informação que agrega e apresenta dados criados ou mantidos pela administração pública. O ambiente promove a interação entre cidadãos e governo para pensar a melhor utilização dos dados abertos em prol de uma sociedade melhor.

ITI: Instituto Nacional de Tecnologia da Informação, seu objetivo é manter a Infraestrutura de Chaves Públicas Brasileiras (ICP-Brasil), sendo a primeira Autoridade Certificadora Raiz Brasileira (AC-Raiz). O certificado digital da ICP-Brasil, além de personificar o cidadão na rede mundial de computadores, garante, por força da legislação atual, validade jurídica aos atos praticados com seu uso (e-Gov, 2000).

Outros portais como: Inclusão Digital, Compras públicas, Licitações e Convênios, Governo para Governo (G2G), Software Livre e Portais de informação restrita, também podem ser conferidos no Portal Oficial do Governo Eletrônico (Govbr)<sup>2</sup>.

Para uso da identidade visual do programa, o Govbr, disponibiliza em seu portal, o manual com as informações sobre o modo de utilização, proporções, reduções mínimas, cores, tipografia, assinatura conjunta horizontal e vertical dos logotipos Govbr.

### 2.1.1.e-MAG

Os projetos desenvolvidos pelo Governo Federal, nas mais diversas áreas, contribuem para a melhoria da sociedade, criando opções de acesso e inclusão

---

<sup>2</sup> Portal Govbr pode ser acessado em: <http://www.governoeletronico.gov.br/o-gov.br>

digital de baixo custo, além de um canal de relacionamento no qual o cidadão pode participar do processo de discussão e implementação das políticas públicas (e-Gov, 2000).

Estes projetos são desenvolvidos em conformidade com os padrões internacionais de acessibilidade, onde consistem em um conjunto de recomendações a ser considerado para que o processo de acessibilidade dos sites e portais do Governo Brasileiro seja conduzido de forma padronizada e de fácil implementação.

O Modelo de Acessibilidade de Governo Eletrônico (e-MAG) foi formulado para orientar profissionais que tenham contato com publicação de informações ou serviços na Internet a desenvolver, alterar e/ou adequar páginas, sites e portais, tornando-os acessíveis ao maior número de pessoas possível (e-Gov, 2000).

A última versão (3.0) do modelo foi elaborada por meio da parceria entre o Departamento de Governo Eletrônico, da Secretaria de Logística e Tecnologia da Informação (SLTI) do Ministério do Planejamento, e o projeto de acessibilidade virtual da Rede de Pesquisa e Inovação em Tecnologias Digitais (RENAPI) seguindo as Recomendações de Acessibilidade para Conteúdo Web (WCAG) 2.0, da World Wide Web Consortium (W3C) que tem como objetivo prover um único padrão compartilhado para acessibilidade de conteúdo web que atenda às necessidades de indivíduos, organizações e governos a nível internacional. Os documentos WCAG explicam como tornar o conteúdo web mais acessível às pessoas com deficiência (WCAG, 2008 ).

#### 2.1.2. e-PWB

Os Padrões Web em Governo Eletrônico (e-PWG) são recomendações de boas práticas agrupadas em formato de cartilhas com o objetivo de estabelecer padrões de qualidade de uso, desenho, arquitetura de informação e navegação; Estabelecer um fluxo de criação, desenvolvimento e manutenção na gestão dos sites governamentais; Consolidar a acessibilidade; Criar artefatos de acordo com os padrões estabelecidos pelo W3C e aprimorar a comunicação e o fornecimento de informações e serviços prestados por meios eletrônicos pelos órgãos do Governo Federal (e-Gov, 2000).



A administração pública é responsável por oferecer ao cidadão a melhor experiência possível de acesso ao governo eletrônico, respeitando inclusive, as particularidades da população atingida, sua preocupação é desenvolver e manter a usabilidade dos *sites* e portais do governo (e-Gov, 2000).

O uso da padronização garante o nível de qualidade dos *sites*, possibilita a mensuração de resultados, fornece requisitos para a correta contratação da equipe/empresa que desenvolva/mantenha o projeto e facilita o processo de adaptação e migração para outras tecnologias (e-Gov, 2000).

As cartilhas estão disponíveis no portal oficial do e-Gov em formatos HTML e PDF:

### **Guia de Administração**

O objetivo do Guia de Administração é oferecer subsídios para a concepção, desenvolvimento, manutenção e administração de sites de governo eletrônico na esfera federal.

### **Cartilha de Codificação**

Neste guia são recomendadas boas práticas em codificação, com o propósito de torná-los identificáveis, portáteis, relevantes, acessíveis e efetivos à população.

### **Cartilha de Usabilidade**

Há certa complexidade na inserção da usabilidade no desenvolvimento e redesenho de sites e esta cartilha propõe ser um guia de forma clara e descomplicada. Neste são apresentadas recomendações de usabilidade descritas de forma prática e aplicável, assim como orientações sobre como realizar testes de usabilidade.

### **Cartilha de Redação Web (*webwriting*)**

Esta cartilha auxilia nas boas práticas de elaboração de informação clara, estruturada e eficaz para o meio digital.

Além das cartilhas o portal também oferece materiais de apoio para o desenvolvimento de sites e portais, incluindo principalmente, uma referência para "Boas práticas em Web Móvel" da W3C.

A adoção de plataformas web para prestar serviços públicos e interagir com os cidadãos é um excelente canal de comunicação entre Governo e Cidadão e, seguir boas práticas e padrões de desenvolvimento é fundamental para que portais Governamentais sejam ágeis e práticos para atender a expectativa do cidadão.

### 3 MOBILIDADE

Esta Seção apresentará informações sobre mobilidade móvel, redes sem fio, difusão da tecnologia móvel no Brasil e como a mobilidade esta afetando o cotidiano das pessoas.

Loureiro e Mateus (1998) já indicavam em seu trabalho o crescimento extraordinário na área da comunicação móvel, redes locais sem fio e serviços via satélite (LOUREIRO; MATEUS, 1998, p.1). Na época, já se falava em mobilidade como capacidade de acessar informações e serviços a qualquer momento e em qualquer lugar.

A pesquisa de doutorado de Ana Lúcia Damasceno Moura Fé, (2008) - Tecnologias móveis e vida pessoal - aborda a mobilidade como um benefício consequente das tecnologias móveis, dando o poder livre de comunicação o qual conFigura um dos principais avanços da comunicação sem fio (FÉ, 2008, p. 63). Fé (2008) também relata a origem da palavra mobilidade que vem do latim *mobilis* que abrange significados como: fácil de mover, livre, não fixo, não firme, flexível, ágil e rápido.

Dentro do contexto tecnológico, a mobilidade é o inverso da computação tradicional, formada por computadores estáticos, também conhecidos como *Desktops* (LOUREIRO; MATEUS, 1998, p.2), por exemplo, para acessar a Internet precisávamos ir até um computador, enquanto que a mobilidade nos oferece dispositivos móveis, como *laptops*, *smartphones* e *tablets* que estão presentes conosco na maior parte do tempo (DARIVA, 2011, p. 3).

O celular tem se consolidado como dispositivo móvel preferido para comunicação (FÉ, 2008, p. 45). No Brasil o total de *smartphones* é de pouco mais de 20 milhões de aparelhos (ALMEIDA; TERRA, 2011, p. 29).

Em 2012 a empresa Google lançou uma pesquisa onde mostra a difusão dos *smartphones*, 14% de adoção no Brasil (Google, 2012). Segundo a Anatel, o crescimento do serviço móvel ainda é contínuo:

No final de 2012, o Brasil contava com 261,81 milhões de acessos do Serviço Móvel Pessoal (SMP), tendo registrado crescimento de 8,08% em relação ao ano anterior, inferior ao observado em 2011. O mercado brasileiro tem crescido nos últimos

anos, o que implica no atendimento gradual da demanda de telefonia móvel pela sociedade[...] (ANATEL, 2013, p.3).

A Secretaria de Desenvolvimento Social e Direitos Humanos e da Secretaria de Ciência e Tecnologia de Pernambuco (SEDSDH, 2013), também estão fazendo sua parte para que o crescimento móvel continue, o Edital de N.º 001/2013, torna pública a concessão de aparelhos de telefonia móvel e/ou de uso de software tradutor de voz, que serão utilizados, em fase de experimental, no Projeto Nambiquara "Audição Digital", executado pela SECTEC em convênio com o Centro de Excelência em Tecnologia de Software do Recife (SOFTEXRECIFE), o projeto é destinado às pessoas com deficiência auditiva. Futuramente a ideia é comercializar o aparelho e/ou o software com preços acessíveis, mais um impulso para o crescimento móvel.

Diante de todo esse progresso da comunicação móvel, os hábitos e conceitos da sociedade tem sofrido um reflexo direto (PELLANDA, 2009).

Arrieta (2008) entrevistou em meados de 2008 na Colômbia, o filósofo italiano Maurizio Ferraris, professor da Universidade de Turim, autor do livro "Onde está você? Ontologia do celular" (2005). Em sua entrevista, Arrieta, aborda justamente o tema de como a comunicação móvel está introduzida em nossas vidas, onde Ferraris responde uma das questões da seguinte maneira:

[...] podemos imaginar alguém que, há uns dez anos, saísse de sua casa às oito da manhã. Este homem, ao sair de sua casa, deixava também seu telefone, porque ele era fixo. Fazia uma viagem de uma hora para chegar ao seu escritório ou de muitas horas por razões de trabalho; durante todo esse período, ele estava completamente desconectado de tudo no mundo, e isso parecia muito natural. Hoje, se esse mesmo homem sair de sua casa sem celular, vai ter a sensação de estar isolado, de ser um homem mudo, sem uma das coisas que representa para ele a segurança ontológica, talvez a mais importante do mundo atual: o celular. (ARRIETA, 2008, p. 154)

A evolução da comunicação sem fio e da tecnologia da informação busca atender muitas das necessidades do mercado: serviços móveis, redes sem fio, sistema de navegação, base de dados geográfica, etc. (LOUREIRO; MATEUS, 1998, p.2). A comunicação sem fio é um suporte para a computação móvel e graças à evolução,

usuários podem manter-se conectados sem a necessidade de uma infraestrutura fixa (LOUREIRO; MATEUS, 1998, p.1).

O fato dos dispositivos móveis estarem sempre à mão, nos permite fazer ações simples como ler as principais notícias enquanto toma-se um café, conferir e redigir e-mails, delegar tarefas (DARIVA, 2011, p. 3).

A mobilidade pode trazer redução de custos e ganho de produtividades para as empresas (DARIVA, 2011, p. 3). Uma aplicação móvel, por exemplo, pode trazer ganhos quando uma compra de passagens aéreas é aprovada, como as passagens áreas têm mudanças de preços com frequência, a demora de um diretor aprovar a compra poderia fazer com que a empresa comprasse bilhetes pelo dobro do preço. Outro exemplo são os sistemas de *Business Intelligence* (DARIVA, 2011, p. 4), onde os enormes relatórios gerados pelo sistema podem ser acessados por *tablets* para que os gestores os visualizem sem a necessidade de gastar com impressões. Podemos citar vários exemplos nos quais a mobilidade trás como benefício: Ganho de tempo ao acionar uma emergência; Na área rural podem-se acompanhar as ações do mercado; Acionar um mecânico para consertar uma colheitadeira quebrada, algo que antigamente poderia custar dias de trabalho parado (TELESP CELULAR, 2002, p. 48); ou simplesmente acompanhar a previsão do tempo. Também tem uso fundamental na logística de transplantes de órgãos e tecidos (TELESP CELULAR, 2002, p. 78), ou até mesmo em rotinas como localizar um endereço ou estabelecimento; Encontrar um amigo que não o vê há tempos, fotografar uma cena inusitada, conferir o saldo bancário, etc.

E cada vez mais o consumo desses equipamentos é potencializado com o lançamento de novas interfaces e formatos dando margem para competitividade entre as grandes empresas que os desenvolvem (PELLANDA, 2009, p. 90).

É possível notar, o quanto a tecnologia móvel, somada com a disponibilidade de acesso à Internet, está ficando cada vez mais presente e próxima de nós. Os dispositivos móveis estão deixando de serem artigos de luxo e diante de tal cenário, é importante realizar estudos que destaquem a forma como os conteúdos e aplicações web estão sendo desenvolvidos.

## 4 WEB DESIGN RESPONSIVO

Usuários de dispositivos móveis que fazem uso da web se deparam em diversas situações com problemas de visualização de *sites* não adaptativos aos diversos dispositivos existentes, como por exemplo: fontes pequenas, imagens cortadas, *links* ilegíveis e conteúdos distorcidos. Cerca de 94% das empresas ainda não possuem um *site* móvel (ALMEIDA; TERRA, 2011, p.29-30) e ter um site adaptado, flexível de conteúdo legível e navegável à internet móvel é imprescindível.

A presente Seção apresenta uma abordagem que pode ser útil para minimizar estas situações.

A tecnologia móvel tem avançado e há muitos *sites* que ainda não consideram este meio de acesso às suas informações. A web de hoje é multi-dispositivos e é necessário considerar que suporte além do *desktop*, telas pequenas, *touch screen*, *smart TV*, redes móveis e muitas outras tecnologias (FISHER; SHARKIE, 2013, p. 1).

Desta forma surgem questões a serem respondidas. Como exibir o mesmo conteúdo web para os mais variados tipos de dispositivos móveis? É possível desenvolver uma única página que seja adaptável para toda essa gama de eletrônicos móveis?

O *design* responsivo aborda justamente estas questões, permitindo que as páginas web respondam a qualquer dispositivo, sem que haja perda de informações por parte do usuário, conforme é apresentado na Figura 1.

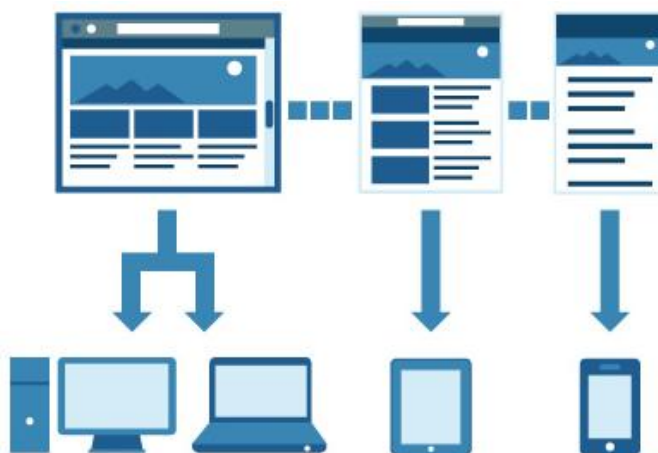
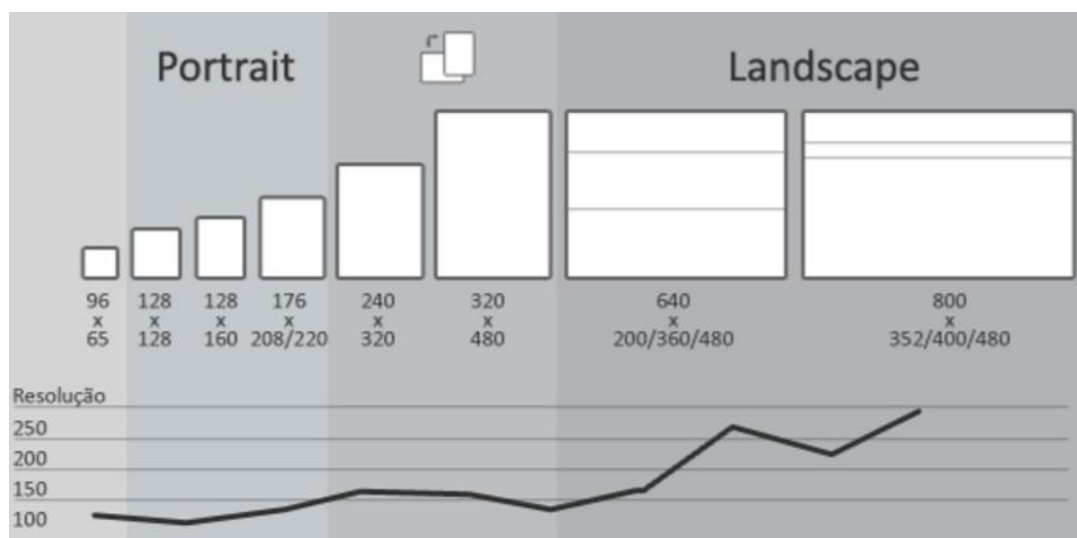


Figura 1 - Comportamento de um *site* responsivo

Fonte: Master Studio Web (2012)

O conteúdo de um site responsivo se adapta à área que cabe a ele, adequando a visualização e a navegabilidade sem perder as informações, independente do dispositivo, independente da resolução, tamanho de tela, interface com *touch* ou *mouse*, seja ele móvel ou não.

Neste contexto é importante conhecer algumas resoluções de telas, levando em consideração sua orientação retrato e paisagem. A Figura 2 apresenta tais resoluções.



**Figura 2 - Resoluções de telas**

**Fonte: (ZEMEL, 2013, p. 21)**

Observa-se na apresentação da Figura 2, que não necessariamente um dispositivo deve ter uma tela maior para se obter uma resolução grande, por exemplo, o iPhone4S mede 58,6mm x 115,2mm e tem resolução de 960 x 640 pixels, já o iPhone5, medindo quase o mesmo tamanho físico que o iPhone4S, tem 640 x 1136 pixels, com diferença de apenas 8,6 milímetros de altura 58,6mm x 123,8mm (Apple, 2013).

Não é o tamanho físico da tela ou do dispositivo que importa no *design* responsivo, mas sua resolução.

Web *design* responsivo envolve uma série de técnicas e tecnologias que são combinadas para fazer uma única aplicação funcionar em uma variedade de dispositivos de modo mais prático possível. E não são apenas os profissionais da web que viram esta necessidade, pequenas e grandes empresas estão procurando maneiras de fazer o seu projeto web, independentemente de onde o usuário pode acessá-lo (FISHER; SHARKIE, 2013, p. 2).

Ethan Marcotte (2011), creditado como o pai da Web *Design* Responsivo (RWD),

publicou em maio de 2010 um artigo no "A List Apart" chamado "Responsive Web Design". Nele, Ethan concentrou-se em *Layout* fluido ou *Grids* flexíveis, Imagens e recursos flexíveis e *Media Queries*. Itens que serão abordados individualmente neste trabalho.

Na opinião de ZEMEL (2013), não existe uma "web *mobile*", a web é única, logo o web *design* responsivo é a resposta única para uma web única. (ZEMEL, 2013, p. 10). Há vários exemplos citados por ZEMEL (2013) de sites responsivos:

Casa do código<sup>3</sup>: A página Web da Casa do Código, acessada pelo navegador de um computador convencional, apresenta a disposição do seu conteúdo de maneira comum comparada as demais páginas web, como mostra a Figura 3.



**Figura 3 - A Casa do Código - Página web acessada através de um computador convencional**

Fonte: (ZEMEL, 2013, p. 3)

Já a Figura 4, mostra a mesma página, mas acessada por um dispositivo móvel, o conteúdo tem um comportamento diferente, observado pela disposição vertical das imagens e textos e da reformulação do menu.

<sup>3</sup> Disponível em: <http://www.casadocodigo.com.br>



**Figura 4 - A Casa do Código - Página web acessada através de um dispositivo móvel**

Fonte: (ZEMEL, 2013, p. 4)

Assim como a Casa do Código, a página web Fork-CMS<sup>4</sup>, também é adaptada conforme o dispositivo que o acesse, nota-se, na Figura 5, a diferença de comportamento entre os elementos e a prioridade na exibição dos principais conteúdos.



**Figura 5 - Fork-CMS - Acesso à página web por diferentes dispositivos**

Fonte: (ZEMEL, 2013, p. 5)

<sup>4</sup> Disponível em <http://www.fork-cms.com>



Outro exemplo citado por ZEMEL (2013), apresentado na Figura 6, é a página web *Mr. Simon Collison*<sup>5</sup>.



Figura 6 - *Mr. Simon Collison* - Página web simples e responsiva

Fonte: (ZEMEL, 2013, p. 6)

Na Figura 6, apresenta uma página web basicamente contida por imagens, é evidente a diferença apresentada quando acessado por outros dispositivos. Apenas o posicionamento das imagens sofreu alteração, não houve um comportamento diferenciado para um menu ou prioridade de exibição de conteúdo, nem por ser simples, deixa ser responsivo.

Uma página web com conteúdo responsivo pode ser acessada nos computadores convencionais, nos *laptops*, nos *smartphones*, *tablets*, TV e qualquer outro dispositivo, que tenha acesso à Internet, de forma bem apresentada.

A técnica preza por não utilizar linguagens próprias para *mobile*, nem desenvolver versões específicas para celulares ou subdomínios e diretórios específicos, a ideia é fazer uma única web.

Em qualquer que seja o dispositivo, o ideal é ter *sites* desenvolvidos que sejam bem apresentados e se possível otimizados conforme a tela que o exiba.

O maior problema do desenvolvimento web convencional, é que estes não foram idealizados desde sua concepção como flexíveis e, além disso, o desafio não é

<sup>5</sup> Disponível em: <http://www.colly.com>

técnico, mas sim conceitual (ZEMEL, 2013, p. 13).

Acreditar que o *site* já está bom para android, iphone e outros, pode levar a interpretações equivocadas, um *site* focado para cada dispositivo, tem suas desvantagens, inclusive quando o fabricante lança um novo aparelho com uma tela totalmente diferente, obrigando-o desenvolver uma nova versão do *site* que se adapte a esse novo aparelho, e com o devido cuidado para que as alterações não sejam incompatíveis com as versões dos aparelhos anteriores (ZEMEL, 2013, p. 14).

Algumas estratégias são adotadas para contornar o problema da adaptabilidade, como o "**m ponto**", ao abrir uma página web pelo celular, o navegador redirecionará para um endereço web parecido com <http://m.minhapaginaweb.com.br>, e se acessado em um computador convencional serão mal formatados e sem uma boa navegabilidade, pois foi definido especificamente para um celular (ZEMEL, 2013, p. 14).

Outra abordagem, também comentada por ZEMEL (2013), é o **One Web**, onde há um único endereço web e um único sistema com o mesmo conteúdo para todos. Ainda há outra técnica chamada **RESS** (*Responsive Design + Server Side Components*), onde se serve ao mesmo endereço web e a mesma página, mas o conteúdo é ajustado parcialmente no *server-side*, sabendo-se qual é o navegador (ZEMEL, 2013, p. 14).

FISHER e SHARKIE (2013) dizem que a melhor forma de se fazer, será diferente para cada desenvolvedor, aplicação e *site* desenvolvido, se a estrutura do projeto é sólida, aplicando as práticas de *Layout* fluido, pode muito bem ser possível, se o conteúdo é sólido e seu CMS, do inglês *Content Management System*, é flexível o suficiente, um novo tema pode ser tudo que se precisa para adicionar *Medias Queries* ou se o código fonte do projeto for robusto, inserir imagens e recursos flexíveis pode ser uma implementação bastante simples e fácil. O projeto pode até não exigir a aplicação de todos os conceitos do *design* responsivo, pode-se apenas criar um *layout* novo com várias *grids* ou melhorar o desempenho do CSS (Folha de Estilo em Cascata, do original *Cascading Style Sheets*) do projeto aplicando as *medias queries*, ou ainda combinar as abordagens (FISHER; SHARKIE, 2013, p. 9-10).

#### 4.1. CSS E DESIGN RESPONSIVO

O item principal para o desenvolvimento deste trabalho, o design responsivo, é baseado na técnica chamada CSS. Esta Seção abordará uma breve introdução do que é esta técnica.

O CSS é responsável pelo aspecto da página, do *layout* e da leitura da página web pelo navegador. Não é uma linguagem de programação ou de marcação, é uma sequência de propriedades com valores definidos para manipular o comportamento e aparência dos elementos da página, podendo estes ser herdados por outros elementos (GOMES, 2010, p. 11).

Normalmente os desenvolvedores de páginas web, criam um arquivo separado do conteúdo HTML (Linguagem de Marcação de Hipertexto, do original *HyperText Markup Language*) para suas folhas de estilo. Quando se faz necessário uma alteração na aparência da página, basta modificar o arquivo CSS (JOBSTRAIBIZER, 2009, p. 6).

A técnica é composta por atributos e valores que podem ser visualizados na Figura 7.

```
1  .nome {  
2      atributo: valor;  
3  }  
4  
5  #nome {  
6      atributo: valor;  
7  }  
8  
9  a:visited {  
10     atributos;  
11 }
```

**Figura 7 - Exemplo de atributos e valores em CSS**

**Fonte: (JOBSTRAIBIZER, 2009, p. 5)**

Ainda na Figura 7, o bloco identificado por ponto (.), na primeira linha, são atributos de classes, podem ser utilizados quantas vezes forem necessários.

O bloco identificado pela cerquilha (#), na linha cinco, são atributos identificadores de elementos HTML, conhecidos como *id*, é utilizado apenas em locais específicos.

E os blocos que não possuem ponto ou cerquilha, linha nove da Figura 7, são atributos de tags HTML como <a> <h1> <div> etc.

O exemplo apresentado na Figura 8 define para a página Web um estilo de fonte, cor e tamanho:

```
1 body{
2   width:100%;
3   font-family: sans-serif;
4   font-size:1.5em;
5   background-color: blue;
6 }
```

Figura 8 - Exemplo de estilo CSS para fontes, cores e tamanhos

Fonte: (JOBSTRAIBIZER, 2009, p. 9)

O exemplo apresentado na Figura 8 foi aplicado para um atributo de tag HTML, o `<body>`, logo todo o "corpo" da página ocupará 100% de largura, os textos ficarão com fonte `sans-serif` de 1.5em de tamanho e terá a cor de fundo azul.

O CSS pode ser aplicado a qualquer elemento HTML, mas este é um assunto bastante amplo e, não é o foco deste trabalho discutir técnicas e teorias do CSS, há vários materiais disponíveis na Internet que abordam o assunto com mais detalhes, inclusive ZEMEL (2103) indica a literatura de (MAZZA, 2012).

#### 4.1.1. Tipos de medidas CSS

Existem ao menos quatro tipos de medidas no CSS: pixel, ponto, porcentagem e em. De acordo com ZEMEL (2013), estas medidas são definidas como segue:

- Pixel (px): é a unidade de medida fixa mais utilizada no CSS. É o menor elemento num dispositivo de exibição. Profissionais de *web design* preferem usar este tipo de medida para fazer uma estrutura HTML/CSS em pixel *perfect*, não medindo esforços para estruturar seus documentos para que fiquem idênticos à imagem do *web design*.

- Ponto (point): são tradicionalmente utilizados para CSS de impressão. Um ponto é igual a 1/72 polegadas. Assim como o pixel, ponto é uma unidade de tamanho fixo.

- Ems (em): é uma unidade escalável. Quando se trata do tamanho da fonte, 1em é igual ao tamanho atual da fonte do elemento pai. Por exemplo, se o tamanho da fonte do elemento é 12pt, 1em é igual a 12pt. Ems são escaláveis por natureza. 2em seria igual a 24pt, 0.5 seria 6pt, etc.

- Porcentagem (%): é muito parecida com a unidade em, mas possui algumas diferenças fundamentais. O atual tamanho da fonte é igual a 100% ou seja, 12pt

equivalem a 100%. Durante o uso da unidade porcentagem, o texto permanece totalmente escalável para dispositivos móveis.

Seguindo a técnica para *design* responsivo, fica claro que deve ser dada preferência para as medidas relativas e fixas, claro que pode ser necessário em algum momento específico, definir medidas fixas. Mas qual a diferença entre as medidas flexíveis **ems** e **porcentagem**?

Conceitualmente parecem ser iguais, mas existem algumas pequenas diferenças. Em termos de *layout* é importante saber que marcar unidades com porcentagem fornece uma exibição do conteúdo mais consistente e acessível para os visitantes. As medidas marcadas com % se alteram de maneira mais razoável, permitindo que a legibilidade seja preservada (ZEMEL, 2013, p, 25).

Apesar de ser possível usar qualquer um dos tipos de medidas relativas, nota-se um uso maior da unidade em porcentagem pelos desenvolvedores, provavelmente chegou-se a este consenso pelos muitos testes e experiências com responsividade, usar porcentagem para o tamanho do *layout* (larguras, margens, espaçamentos, etc.) e usar em para as fontes. A medida em "em" pode até ser usado fora de textos, mas vai ser sempre uma medida relativa ao `font-size`; já a porcentagem é relativa ao `font-size` quando usada em `font-size`, mas, quando usada como outras medidas, é relativa à largura do elemento pai. (ZEMEL, 2013, p, 25).

## 4.2. OS PILARES DO DESIGN RESPONSIVO

A presente Seção abordará os três principais conceitos da técnica *design* responsivo: *Layout* fluido, Flexibilidade de imagens e recursos e, *Media Queries*.

### 4.2.1. Layout fluido

Também chamado de *Grid* flexível por Ethan Marcotte (2011), é o primeiro passo a ser dado no desenvolvimento web *design* responsivo, seu principal conceito é **não utilizar medidas absolutas no CSS** (ZEMEL, 2013, p. 23) como *pixels*, ou pontos, centímetros e milímetros. *Layout* fluido significa usar medidas flexíveis e é algo tão antigo quanto o HTML (LOPES, 2013, p. 49).

Esta técnica é baseada em usar valores percentuais ao invés de absolutos, as

porcentagens são usadas para especificar medidas de tamanho com relação ao tamanho do elemento pai (LOPES, 2013, p. 49).

Ao invés de declarar a largura da página com valores absolutos em pixels (px), são usados valores proporcionais em percentual (%), desta maneira a página e suas subpartes ocuparão uma área relativa ao tamanho da janela do navegador. O trecho de código da Figura 9 expõe um código usando medidas flexíveis:

```

1 body {
2     /* a página ocupa a largura da tela toda */
3     width: 100%;
4 }
5
6 article {
7     /* o article ocupa 3/4 da página */
8     width: 75%;
9     /* e tem uma margem interna de 10% do tamanho do pai */
10    padding: 10%;
11 }

```

**Figura 9 - Medidas flexíveis no CSS**  
(LOPES, 2013, p. 49-50)

Quando, um projeto tem por objetivo contar com *design* responsivo, é importante que os desenvolvedores envolvidos pensem de forma flexível, ou seja, em implementar o *layout* se valendo de medidas relativas no CSS, atribuindo medidas em **porcentagem** (%) ao *layout* e o tamanho de fontes em **ems** (em) ao invés de **pixels** (px). Se o projeto está passando por uma reformulação, as medidas absolutas existentes no CSS devem ser convertidas para medidas relativas.

Existe uma fórmula simples e rápida de realizar a conversão de tamanhos absolutos para tamanhos relativos em layouts web (ZEMEL, 2013, p.26; MARCOTTE, 2011, p.20):

$$\text{Alvo/Contexto} = \text{Resultado}$$

Onde, Alvo é o elemento-alvo com a medida atual, Contexto é onde o elemento-alvo está (baseado no elemento pai) e Resultado representa o valor relativo que se está procurando.

Esta fórmula é tanto para o cálculo de tamanho de fontes, apesar de existir certa convenção de que o tamanho padrão de fontes em navegadores *desktop* é de 16px, quanto para medidas de *layout* (ZEMEL, 2013, p27).

Aplicando este cálculo, por exemplo, em um título com tamanho de fonte de 24px, ficaria assim:

$$24 / 16 = 1,5$$

Onde, Alvo é o título de 24px; Contexto é o elemento pai *body*, com tamanho de fonte de 100%, ou no caso 16px e resultado da divisão é 1,5em.

Agora considerando um *link* dentro do título com tamanho de fonte de 11px, neste caso o contexto muda, agora o elemento alvo não é mais o contexto geral, *body*, logo:  $11/24=0,458333333333333$  (ZEMEL, 2013, p. 27).

Desta vez o alvo é de 11px e que está dentro do contexto do título de 24px, logo o resultado é 0.458333333333333em.

Marcotte (2011) sugere para resultados extensos após a vírgula, deixar até quatro casas, já Zemel (2013) diz para realizar nossos próprios testes e optar pelo qual trazer melhores resultados.

Quando se está acostumado com medidas fixas, usar medidas flexíveis de maneira eficaz pode ser bastante difícil. O segredo de um *layout* com medidas flexíveis é pensar na proporção dos elementos (LOPES, 2013, p 53). A Figura 10 apresenta o exemplo de um *layout* onde há quatro elementos na parte inferior da página, que somados ocupam 100% da largura do navegador.



Figura 10 - Layout flexível - Proporção dos elementos

Fonte: (LOPES, 2013, p.53)

Ainda na Figura 10, a proporção dos elementos é estabelecida na porcentagem e o tamanho final é recalculado automaticamente pelo navegador do dispositivo que o acessa. No caso das fontes, a estratégia é a mesma, se o título principal é 50% maior que o título secundário, estabeleça essa mesma relação no desenvolvimento da página, onde neste exemplo poderia ser uma fonte de 1em e outra 1.5em, se a fonte base da página for alterada, todas as medidas em em mudam junto proporcionalmente (LOPES, 2013, p. 54).

Outra questão interessante apresentada por (LOPES 2013), refere-se à

realização do cálculo necessário para se obter, por exemplo, uma página com dois elementos ocupando 50% da tela cada com uma margem de 5% entre os elementos. Observando o contexto, nota-se que os elementos não podem ter 50% de largura cada, pois há uma margem a ser considerada, logo o cálculo das medidas devem ser redefinidos, para que atinjam corretamente a largura total disponível de 100%.

No entanto, a propriedade CSS chamada `box-sizing`, pode ser utilizada para que não haja a necessidade de se fazer estes cálculos precisos. A propriedade faz com que os tamanhos dos elementos se relacionem, assim determinam o tamanho final desejado (LOPES, 2013, p. 55).

Para tanto, basta atribuir o trecho de código da Figura 11 no arquivo CSS do projeto:

```
1 * {  
2   -webkit-box-sizing: border-box;  
3   -moz-box-sizing: border-box;  
4   box-sizing: border-box;  
5 }
```

**Figura 11 - Definindo um `box-sizing`**

**Fonte: (LOPES, 2013, p. 56)**

No exemplo da Figura 11, a propriedade `box-sizing` é aplicada para todos os elemento da página e prevê a compatibilidade entre os navegadores existentes.

Na opinião de (LOPES, 2013), seria mais fácil analisar o tamanho do elemento considerando seu tamanho total ao invés de subtrair os valores dos outros elementos, deixando os valores mais óbvios a primeira vista (LOPES, 2013, p. 55).

#### 4.2.1.1. 100% Layout fluido

Para abranger os inúmeros tamanhos de telas dos diferentes dispositivos existentes, desenvolve-se um *layout web* totalmente fluido para o projeto. Mas uma questão é levantada: deixar a página ocupar 100% da tela sempre atingirá os resultados esperados?

Por exemplo, um *layout* dividido em quatro colunas com espaçamentos entre as colunas, acessado por um monitor com 4000px de resolução, terá sua apresentação funcionando sem maiores problemas, porém com uma aparência não tão aprazível, pois a resolução é grande, o *layout* foi definido para ser 100% fluido,



ocupará 100% da tela, logo o resultado deste exemplo trará os espaçamentos entre as colunas de maneira exagerada, dando a impressão de espaços mal aproveitados no *layout* (LOPES, 2013, p. 57).

Deixar um *layout* 100% fluido, nem sempre é tão interessante, principalmente nos extremos, ou muito pequeno ou muito grande, no caso dos televisores (LOPES, 2013, p. 56). O ideal é restringir o *layout* fluido, limitar a flexibilidade em casos específicos.

Para tanto, o CSS provê de um atributo, bastante utilizado inclusive, para limitar a largura nestas situações, este atributo é o `max-width`, que pode ser para um elemento em específico ou então para a própria página, no caso o `body` (LOPES, 2013, p. 57). A Figura 12 a seguir, mostra um exemplo de uma restrição de *layout* fluido:

```
1 body {  
2   max-width: 2000px;  
3   margin-left: auto;  
4   margin-right: auto;  
5   width: 100%;  
6 }
```

**Figura 12 - Limitando um layout fluido com max-width**  
(LOPES, 2013, p. 57)

O trecho de código apresentado na Figura 12 resultará em uma página web limitada há 2000 pixels de largura, isso significa que, se acessá-la em um dispositivo com maior resolução, a página não ocupará 100% da largura do navegador e os atributos `margin-left` e `margin-right` definidos como `auto`, deixará a página centralizada. Lopes (2013) afirma que o ideal seria pensarmos em flexibilidade total, não ter restrição e deixar tudo fluir, mas o ideal também seria manter o *design* adaptável e bonito em grandes proporções.

#### 4.2.1.2. O viewport

*Viewport* é o espaço disponível para a página ser renderizada no navegador, é todo o espaço branco de um navegador onde a página é exibida e pode ser medido pelo *javascript* através de `window.innerWidth` e `window.innerHeight` (LOPES, 2013, p. 68).

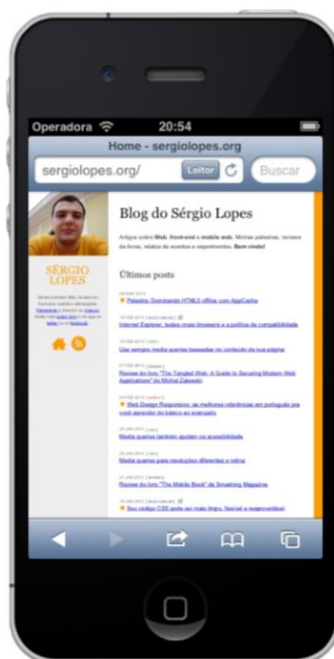
Todo navegador moderno permite que o usuário faça um *zoom* no conteúdo da

página, é um recurso interessante para páginas que não foram desenvolvidas para dispositivos com telas de tamanhos diferentes. Quando o usuário dá *zoom*, de 200% por exemplo em uma página e, nela existe uma imagem de `width: 300px`, esta ficará com 600px de largura, isso é o que chamamos de CSS pixel, neste caso, o CSS pixel faz os 300px da imagem serem renderizados para 600px físicos. A regra é (LOPES, 2013, p. 69):

$$\text{nº de pixel físico} = \text{nº de CSS pixel} \times \text{zoom.}$$

Ao dar zoom os elementos da página ficam maiores e a capacidade de exibí-los na tela é menor, pois o *viewport* diminuiu. Isso ocorre porque o tamanho da tela é medido em *pixels* físicos e o tamanho do *viewport* é medido em CSS *pixels*, que mudam conforme o zoom aumenta. (LOPES, 2013, p. 70).

Em um dispositivo móvel, as páginas feitas para *desktops* são redimensionadas e o conteúdo é diminuído para caber no celular. Lopes (2013) mostra na Figura 13 o exemplo do seu *blog*<sup>6</sup> pessoal, desenvolvido para *desktop*, mas acessado em um *smartphone*.



**Figura 13 - Exemplo do funcionamento do viewport**

**Fonte: (LOPES, 2013, p.71).**

Segundo Lopes (2013), sua foto foi definida no CSS da sua página com largura de 280px, nota-se na Figura 13, que a imagem não ocupou toda a tela do

---

<sup>6</sup> Disponível em <http://sergiolopes.org/>

*smartphone* - 320px - isso é o que o *viewport* faz:

"Um iPhone tem 320 pixels físicos de largura mas assume um viewport de 980 CSS pixels." (LOPES, 2013, p. 71).

A foto que consta na Figura 13 ocupa aproximadamente 28% da tela, mesmo sendo definida em 280px e o dispositivo tendo 320px de resolução. O navegador do dispositivo faz algo semelhante ao *zoom* utilizado no *desktop*, o navegador usa uma escala diferente para o CSS *pixel* em relação ao *pixel* físico, a imagem é criada com 280px, mas a página é enxergada com 980px, todo o trabalho de redimensionamento é feito pelo navegador (LOPES, 2013, p 71).

O *zoom* usado nos celulares são chamados de *page scale*, diferente do *desktop*, chamado *page zoom*, o *page scale* simplesmente dimensiona a parte visível da página, ele não altera o *design* da página, o *site* continua renderizado, o *zoom* só faz a visualização de um pedaço específico da página, ou seja, o *page zoom* altera o tamanho do *viewport* e o *page scale* altera o quanto vemos o *viewport*, mas ele continua igual (LOPES, 2013, p 71).

É neste contexto que Lopes (2013) se refere ao termo dois *viewport*, um representa a área disponível para a página, chamado de *layout viewport*, onde no iPhone mede 980px físicos e o *visual viewport* que é o tamanho que vemos atualmente na tela (LOPES, 2013, p. 72). Esta comparação pode ser vista na Figura 14.

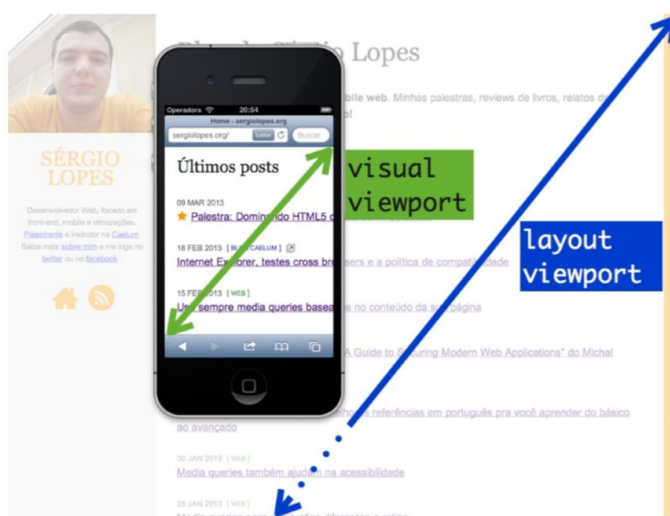


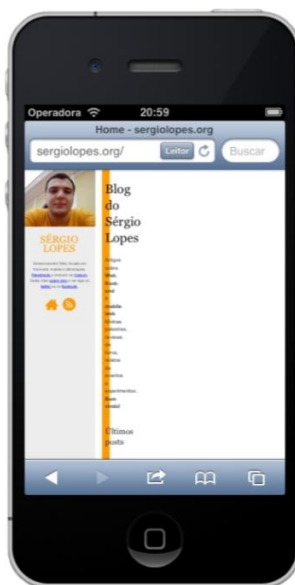
Figura 14 - Comparativo entre Layout viewport e Visual viewport

Fonte: (LOPES, 2013, p.73)

Abrir uma página web projetada para *desktop* em um celular é uma experiência pouco agradável, o correto seria desenvolver uma página otimizada para *mobile* que não demande tanto *zoom* e que mostre o conteúdo em tamanho e formato interessante para uma tela pequena.

Deixar a página com *layout* fixo de 960px, por exemplo, não seria a melhor solução. Criar a página usando porcentagens como `width:100%` no elemento principal também não irá resolver, porque o *layout viewport* é grande - 980px no caso do iPhone citado acima. A página será mostrada como se fosse de *desktop*, com *zoom* mínimo e conteúdo praticamente ilegível (LOPES, 2013, p 74).

Continuando com o exemplo do *blog*, Lopes (2013) sugere colocar o `width` de 320px, já que este valor é a largura real do dispositivo, mas o resultado não foi satisfatório, como mostra a Figura 15:



**Figura 15 - Testando a largura da página web com a mesma largura que tamanho real do dispositivo móvel**

**Fonte: (LOPES, 2013, p.75).**

Observa-se na Figura 15 que o *layout viewport* continua com 980px, mas o conteúdo ficou restrito em 320px, o conteúdo ficou quebrado e com espaços em branco. É necessário redimensionar o *layout viewport* para que ele seja mais adequado à tela (LOPES, 2013, p 75).

A *Apple*<sup>7</sup> introduziu uma *metatag viewport* no iPhone que depois foi adotada em

---

<sup>7</sup> Apple Inc: empresa multinacional norte-americana de produtos eletrônicos. Disponível em: <http://www.apple.com/>

praticamente todas as plataformas móveis, com finalidade de indicar ao navegador que o *layout viewport* deve se comportar em relação ao tamanho especificado. A Figura 16 aplica um exemplo de viewport.

```
1 <html>
2 <head>
3   <meta name="viewport" content="width=320">
4 </head>
5 </html>
```

**Figura 16 - Aplicação da metatag viewport para 320 pixels de largura  
(LOPES, 2013, p 75)**

Na Figura 16, o *layout viewport* terá 320px, logo definir `width:100%` de largura, significa usar exatos 320 pixels, uma visão mais confortável da página como mostra a Figura 17.



**Figura 17 - Resultado do uso da metatag viewport com 320px de largura  
Fonte: (LOPES, 2013, p. 76)**

Mas seria correto deixarmos fixo o valor da *metatag viewport* em 320px, uma vez que o mercado dispõe de aparelhos com tamanhos diferentes como: um Galaxy S3 de 360px , um iPad de 768px, Nexus 4 de 384px, um Galaxy Note de 400px, um Kindle Fire de 600px, etc (LOPES, 2013, p. 77)?

A ideia é deixar flexível e, isso é possível também com a *metatag viewport*, apresentada no exemplo da Figura 18.

```

1 <html>
2 <head>
3   <meta name="viewport" content="width=device-width">
4 </head>
5 </html>

```

**Figura 18 - Aplicação da metatag viewport sem especificação da largura**

**Fonte: (LOPES, 2013, p. 77)**

A Figura 18 mostra um trecho de código onde `width=device-width` indicará o valor específico de cada aparelho, ou seja, o valor mais adequado para seu tamanho, conforme determinado pelo fabricante (LOPES, 2013, p. 77).

A mesma *metatag* pode receber outros parâmetros, como na Figura 19.

```

1 <meta name="viewport" content="width=device-width, initial-scale=1">

```

**Figura 19 - Aplicação da metatag viewport com valor de escala**

**Fonte: (LOPES, 2013, p. 77)**

Onde, a Figura 19, mostra que o parâmetro `initial-scale=1` indicará ao navegador que a página deve abrir no tamanho especificado. Este parâmetro é importante para os sistemas operacionais fabricados pela empresa Apple, pois sem a especificação deste, as orientações retrato e paisagem, nos dispositivos móveis da Apple, seriam as mesmas (LOPES, 2013, p. 77).

A Figura 20 apresenta outros parâmetros que podem ser utilizados na *metatag*.

```

1 <meta name="viewport" content="width=device-width, user-scalable=no">
2 <meta name="viewport" content="width=device-width, minimum-scale=1, maximum-scale=1">

```

**Figura 20 - Parâmetros do viewport**

**(LOPES, 2013, p. 87)**

Descrevendo a Figura 20, o parâmetro `user-scalable=no` desabilita a opção de *zoom* no conteúdo web e os parâmetros `minimum-scale=1` e `maximum-scale=1` limitam o *zoom* em uma escala. Sergio Lopes (2013, p. 87-88) não recomenda tal técnica, pois telas pequenas, mesmo que tendo o conteúdo adaptável, pode trazer desconforto para alguns usuários, nem todos tem uma boa visão o suficiente para nunca usar o *zoom* em uma página web (LOPES, 2013, p. 89).

Lopes (2013) também aborda em seu livro, a especificação CSS *Device*

*Adaptation* da W3C<sup>8</sup>. A *metatag viewport* foi inventada pela *Apple* e copiada pelos demais, acabou virando um padrão de mercado que não é a maneira oficial de se fazer. O que diferencia é o local onde este é especificado, no caso, no CSS ao invés do HTML, é uma nova regra que pode ser usada conforme a Figura 21.

```

1 @viewport {
2     width: device-width;
3 }
```

**Figura 21 - Viewport definido no CSS**  
(LOPES, 2013, p. 77)

A especificação mostrada na Figura 21 pode ser combinada com *media queries*, Seção que será abordada em seguida, mas Lopes (2013) faz um lembrete, é uma especificação nova, ainda são poucos os navegadores que dão suporte a esta especificação.

#### 4.2.2. Imagens e Recursos flexíveis

Uma parte bem complicada de um bom *design* responsivo é resolver como lidar com as imagens, normalmente elas são criadas de um número fixo de pixels e são grandes vilãs de um design fluido. Colocar porcentagens fará a imagem aumentar ou diminuir de acordo como tamanho da tela ou do elemento pai. Uma imagem de 300px, por exemplo, sendo esticada a 100% de uma tela de 1280px, ficaria com uma aparência bem ruim (LOPES, 2013, p. 131).

Ethan (2011) questiona em seu livro - *Responsive Web Design: O que acontece quando colocamos imagens de largura fixa em um projeto flexível?*

Para encontrar uma resposta, a Figura 22 mostra um exemplo criado:

```

1 <div class="figure">
2     <p>
3         
4         <b class="figcaption">Lo, the robot walks</b>
5     </p>
6 </div>
```

**Figura 22 - Exemplo de inserção de um imagem com largura fixa**  
Fonte: (MARCOTTE, 2011, p. 43)

No código da Figura 22, há um elemento div com uma classe chamada *figure*,

---

<sup>8</sup> Disponível no endereço eletrônico: <http://dev.w3.org/csswg/css-device-adapt/>

dentro deste há uma imagem e um subtítulo de classe `figcaption`.

A Figura 23, a seguir mostra como o CSS da classe `figure` foi implementada.

```

1  .figure {
2      float: right;
3      margin-bottom: 0.5em;
4      margin-left: 2.53164557%;
5      width: 48.7341772%;
6  }

```

Figura 23 - Implementação da classe `figure` no CSS

Fonte: (MARCOTTE, 2011, p. 43)

O código CSS da Figura 23 cria um estilo para o elemento `div` onde será posicionado à direita, com um pouco de margem à esquerda e ocupará quase metade do *layout*. Ethan (2011) escolheu uma imagem de grande resolução e obteve o seguinte resultado proposto na Figura 24:



Figura 24 - Exemplo de uma imagem de grande resolução com largura fixa

Fonte: (MARCOTTE, 2011, p. 45)

Nota-se na Figura 24 que a imagem implementada, excedeu na largura do recipiente, no caso a `div`, para solucionar este tipo de situação, tanto (MARCOTTE, 2011) quanto (ZEMEL, 2013), aplicam o atributo CSS para o elemento `img`, chamado `max-width`, como na Figura 25.

```

1  img {
2      max-width: 100%;
3  }

```

Figura 25 - Propriedade CSS `Max-width`

Fonte: (MARCOTTE, 2011, p. 45)



Da maneira como mostra a Figura 25, o elemento `img` recebe uma restrição de tamanho máximo de largura, no caso 100% do elemento que o contém, fazendo com que o caiba de forma correta no recipiente. O resultado desta implementação é observado na Figura 26.



Figura 26 - Adequando a imagem ao elemento que a contém

Fonte: (MARCOTTE, 2011, p.46)

Esta mesma solução, `max-width`, pode ser aplicada para outros elementos (MARCOTTE, 2011, p. 48) como `img`, `embed`, `object`, `video`.

Outra maneira de trabalhar com imagens flexíveis é criar tamanhos diferentes de imagens para contextos diferentes (LOPES, 2013, p.137), e junto com a técnica *media queries* podem-se criar regras que definem quando a imagem de tamanho mais adequado será exibida para o contexto que a acessa.

Existem soluções para imagens responsivas na especificação do HTML5. A mais simples é um novo atributo `srcset` que permite listar mais de um arquivo ao mesmo tempo e deixar o navegador escolher o mais apropriado para a situação, como no trecho de código da Figura 27.

```
1 <img srcset="mobile.jpg 400w, medio.jpg 700px, grande.jpg">
```

Figura 27 - Exemplo de implementação do atributo `srcset` do HTML5

Fonte: (LOPES, 2013, p.138)

No código fonte da Figura 27, `400w` indica que a imagem `mobile.jpg` deve ser carregada no caso do *viewport* ser no máximo de 400 pixels e assim para os demais casos. Há também uma versão equivalente do `srcset` no CSS através da função

`image-set()` que pode ser usada no atributo `background`, mas o suporte para os navegadores existentes ainda é baixo. (LOPES, 2013, p.138). A Figura 28 mostra como ficaria a função `image-set`.

```
1 .banner {  
2   background: image-set(url(mobile.jpg) 400w, url(grande.jpg));  
3 }
```

Figura 28 - Exemplo de implementação da função `image-set()` no CSS

Fonte: (LOPES, 2013, p.138)

Imagens responsivas ainda é um tema bastante discutido, ao mesmo tempo em que é simples colocar porcentagens e torná-las fluídas, tem-se o problema de servir imagens diferentes em contextos diferentes. A maneira menos prejudicial seria usando CSS e *media queries*, mas para isso, você precisa usar `background-image` e nem sempre esta é a maneira que o desenvolvedor gostaria de aplicá-las. É preciso testar e tomar decisões conscientes para que o esforço dê bons resultados. (LOPES, 2013, p. 143-144).

#### 4.2.3. Media Queries

Ethan Marcotte (2011) apresentou as três técnicas do *design* responsivo: *layout* fluido, mídias flexíveis e *media queries*, onde uma página desenvolvida com *layout* fluido, já pode ser considerada responsiva, mas somente com esta técnica, não se resolve todos os problemas (LOPES, 2013, p. 59).

É a partir deste contexto que surgem as *medias queries*, uma técnica que permite que elementos não sejam somente flexíveis, mas que também possam obter comportamentos diferentes, onde suas posições podem ser alteradas, ocultadas ou redimensionadas conforme a necessidade e tamanho de tela disponível. (ZEMEL, 2013, p. 82)

Analisando a Figura 29, observa-se que o *layout* fluiu conforme o tamanho da tela, mas os elementos que contém os textos das notícias estão sobrepostos.



**Figura 29 - Layout fluido sem *Media Queries***

**Fonte: (LOPES, 2013, p. 60)**

Uma sugestão mais apropriada, no caso da Figura 29, seria trabalhar de tal forma, que cada notícia, incluindo imagem e texto, ocupasse 100% da largura da página posicionados uma abaixo da outra, mas para este tipo de comportamento acontecer, deve existir a implementação de *medias queries* para o caso.

A técnica *medias queries* é a principal aliada do *layout* fluido, é uma técnica disponível no CSS que permite trabalhar com design condicional, para que os elementos possam comportar-se de maneiras diferentes à medida que resolução do navegador mude. (LOPES, 2013, p. 60)

A Figura 30 seguinte, exhibe em trechos de código, como ficaria a sugestão implementada para o caso da Figura 29.

```

1 .noticia {
2     float: left;
3     width: 25%;
4 }
5
6 @media (max-width: 400px) {
7     .noticia {
8         float: none;
9         width: 100%;
10    }
11 }

```

**Figura 30 - Implementando o exemplo com *Media Queries***

**Fonte: (LOPES, 2013, p. 60-61)**

Comentando as instruções da Figura 30, nota-se a existência de uma classe CSS chamada `.noticia` e de uma *media query* definindo uma regra para

navegadores com no máximo 400 pixels de largura. Caso a condição, navegador que acessa a página tenha no máximo 400px, ocorra, o comportamento dos elementos da classe `.noticia` devem mudar conforme especificado dentro do bloco `@media (max-width: 400px)`, que seriam, ocupar 100% da largura do recipiente, no caso a página e, sem alinhar à esquerda, como mostra o resultado na Figura 31.



Figura 31 - Resultado do uso de *Media Queries*

Fonte: (LOPES, 2013, p. 61)

Observa-se na Figura 31, que o comportamento dos elementos mudou em relação à Figura 29. Este resultado deu-se conforme a regra criada com *media query*.

Mas criar *medias queries* para cada tamanho de tela existente não é uma boa prática. Lopes (2013) relata a importância do *layout* fluido no desenvolvimento de páginas Web, pois o *design* irá se adaptar naturalmente a várias resoluções e só quando necessário, as regras de media queries serão aplicadas. No entanto a questão surge: quando tomar esta decisão?

Um conceito chamado *breakpoint*, define os pontos onde o *layout* vai ser ajustado por causa de uma resolução diferente (LOPES, 2013, p 91; ZEMEL, 2013, p, 95).

Na Figura 30, há um *breakpoint* onde deixam as notícias em apenas uma coluna. Seguindo este contexto, pode-se definir um *breakpoint* intermediário, de duas colunas para um dispositivo maior, outro *breakpoint* de três colunas para outro

dispositivo maior ainda, ou para resoluções no formato paisagem, como é o exemplo da Figura 32 seguinte e, assim por diante.



**Figura 32 - Exemplo de breakpoint definido para largura máxima de 600px**  
**Fonte: (LOPES, 2013, p.63)**

Para a Figura 32, definiu-se um breakpoint conforme a Figura 33:

```

1 @media (max-width: 600px) {
2   .noticia {
3     width: 50%;
4   }
5 }
  
```

**Figura 33 - Implementação de breakpoint para medidas intermediárias.**  
**Fonte: . (LOPES, 2013, p. 62)**

Na Figura 33, a classe `.noticia` tem 50% da tela ocupada quando a resolução do navegador tiver no máximo 600 pixel de largura.

Lopes (2013) faz uma observação muito importante quanto ao uso de várias *medias queries*, a ordem em que os *breakpoints* forem definidos no código fonte deve ser respeitada, já que a última escrita sobrescreve a anterior.

Existem outros atributos que podem ser utilizados nas *medias queries*, como: *width*, *device-width*, *height*, entre outras, mas muitas delas irão restringir a fluidez do *design* (LOPES, 2013, p. 63).

O descritivo abaixo, explica cada um destes atributos:

*Width*: definir, por exemplo, um valor 320px para *width* ao invés de usar *max* e ou *min-width*, só irá ter efeito em telas com tamanhos exatos de 320px de largura;

*Device-width*: este representa o tamanho da tela do aparelho, diferente ainda do *width* que representa o tamanho da janela do navegador e, na maioria dos casos o interesse é pelo tamanho do navegador e não pelo tamanho tela do aparelho.

*Height*: este atributo pode ser útil para alguns casos, mas normalmente se está interessado na largura do navegador e não na altura, uma vez que a web flui verticalmente (LOPES, 2013, p. 64).

Também é possível usar *media query* para representar capacidade e característica do navegador como a *orientation*, que obtém a orientação do navegador, seja paisagem ou retrato, que pode se utilizado para aperfeiçoar o *design* conforme o dispositivo é segurado (LOPES, 2013, p. 64-65).

Existem outros atributos<sup>9</sup> que podem ser utilizados em *media queries*, mas que na maioria dos casos, o uso é voltado para `max-width` e `min-width`.

### 4.3. MOBILE FIRST

*Mobile first* é uma metodologia criada por Luke Wroblewski<sup>10</sup>, com o objetivo de planejar o projeto web responsivo, primeiramente para dispositivos móveis e gradualmente para dispositivos maiores (ZEMEL, 2013, p. 112).

A abordagem foca em projetar com melhoramento progressivo (*progressive enhancement*), é uma metodologia que exige cuidado, tempo, e disciplina (ZEMEL, 2013, p. 112), ao invés de criar interação totalmente diferentes no projeto responsivo a cada dispositivo novo lançado, basta otimizar a experiência para o novo contexto sem grandes modificações, esta é a ideia de *Mobile first*.

Projetar primeiramente para dispositivos móveis, não apenas prepara o crescimento explosivo da internet móvel, mas também força a se concentrar e permite inovar em maneiras que antes não se podia prever (WROBLEWSKI, 2011, p.5).

Segundo LOPES (2013) é melhor começar seu *design* tendo em mente as restrições do *mobile* e depois evoluir para os computadores de mesa do que ter o retrabalho de limitar após o desenvolvimento já ter sido definido para o *desktop*.

Em termos práticos, a metodologia força os desenvolvedores a focar e priorizar o conteúdo, uma vez que as restrições *mobile* definem um *design* mais simples e funcional e quando a evolução para a versão *desktop* acontece o resultado também é uma interface mais simples, diz LOPES (2013).

---

<sup>9</sup> Este atributos podem ser conferidos no site: <http://www.w3.org/TR/css3-mediaqueries/>

<sup>10</sup> Página pessoal de Wroblewski [http://www.lukew.com/resources/mobile\\_first.asp](http://www.lukew.com/resources/mobile_first.asp)

Desenvolver primeiramente para *desktop* permite acrescentar muito conteúdo a interface e quando este é transformado para *mobile* o *design* pode parecer exagerado e difícil de adaptar. No *mobile first* o foco é definir o que é prioridade, qual é o principal conteúdo, em uma interface simples e atinja o mesmo objetivo da interface para *desktop*, tarefa bastante difícil (LOPES, 2013, p. 34).

## 5 ESTUDO DE CASO

A presente Seção é trás um estudo de caso baseado no projeto piloto "Gráficos estatísticos" desenvolvido por uma Prefeitura do Estado do Paraná, que tem por objetivo analisar o projeto e aplicar as técnicas e metodologias descritas neste trabalho.

### 5.1. METODOLOGIA

Para o desenvolvimento deste caso, todas as técnicas e metodologias abordadas neste trabalho, foram utilizadas.

Primeiramente definiu-se o layout para cada dispositivo. O passo seguinte foi revisar todo o estilo da página (CSS) para que os elementos de layout fossem flexíveis. Durante este passo, usaram-se recursos web que permitiram testar o comportamento do layout da página conforme o navegador era redimensionado, desta maneira simulando vários tamanhos de dispositivos.

Logo após foi implementado o comportamento do menu de opções, onde dependendo do tamanho disponível de tela, este ficaria disposto em locais diferentes e com tamanhos diferentes.

O quarto passo foi aplicar a flexibilidade nas imagens (gráficos) e o como elas eram exibidas, caso visualizadas em televisores, estas eram vistas uma a uma, caso acessadas em dispositivos móveis, estas tinha comportamento sequencial.

E por fim aplicou-se a técnica *Viewport*, podendo perceber de maneira bem clara a diferença nos tamanhos das fontes do projeto, tornando-as mais apresentáveis e proporcionalmente visíveis.

### 5.2. CONTEXTO

O projeto piloto surgiu devido à dificuldade, do Prefeito obter de forma rápida e fácil, dados importantes gerados pelas Secretarias do Município como: índice da folha de pagamento; número de ocorrências atendidas pela Guarda Municipal; arrecadação do Município; número de servidores atualizado; gastos de cada Secretaria; maiores fornecedores do Município; empenhos pagos e a pagar, etc.



Informações estas que eram solicitadas pelo próprio Prefeito via telefone e entregues, nem sempre em tempo hábil, via relatórios impressos. A dificuldade do acesso a estas informações, a maneira como esta era apresentada e o tempo que se levava para a obtenção dos relatórios, foi o motivo que levou o desenvolvimento deste projeto.

O objetivo deste trabalho é transformar os dados relevantes de cada Secretaria, com base nas solicitações do Prefeito, em gráficos que possam ser visualizados em dispositivos como TV ou monitor de grandes resoluções. Futuramente este mesmo modelo será aplicado nos gabinetes de cada Secretário com gráficos relacionados à sua Secretaria e também será disponibilizado um *tablet* para maior mobilidade.

Alguns benefícios deste projeto:

- Fácil acesso;
- Melhor aparência;
- Fácil interpretação dos dados;
- Dados atualizados em tempo real;
- Redução no consumo de papel;
- Interação com os gráficos (detalhamento das informações apresentadas) e,
- Mobilidade quando acessado por *tablets*, *notebooks* e *smartphones*.

O público alvo deste trabalho envolve o Prefeito Municipal de São José dos Pinhais, os Secretários e Diretores de departamentos.

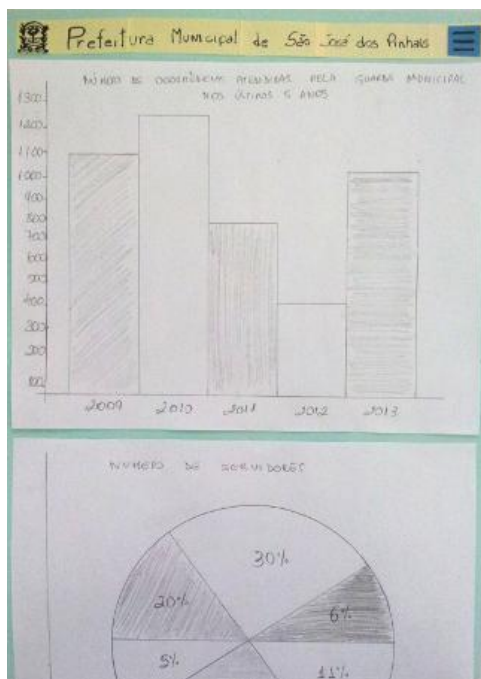
### 5.3. PROJETOS DE INTERFACE

O projeto de interface foi dividido em duas etapas que estão descritas nas subseções 5.2.1, 5.2.2.

#### 5.3.1. Visão interativa de baixa fidelidade

A visão interativa de baixa fidelidade são esboços de uma primeira ideia de interface para o projeto, simulando o acesso por diferentes dispositivos.

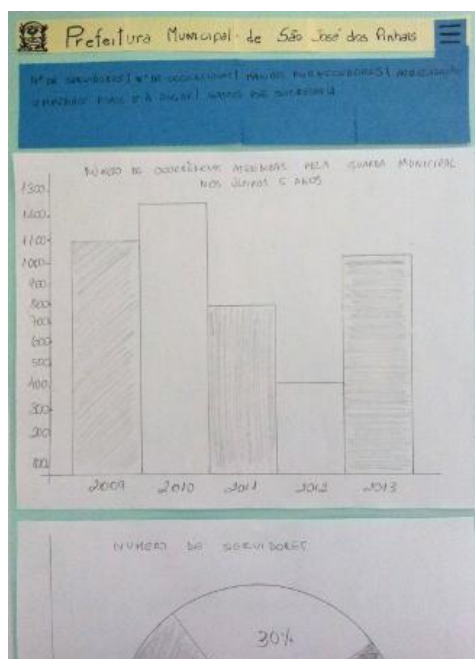
A primeira visão é apresentada na Figura 34.



**Figura 34 - Esboço de interface para smartphone na orientação retrato com menu retraído**  
**Fonte: Autoria própria**

A Figura 34 é uma visão interativa de baixa fidelidade do projeto "Gráficos estatísticos" acessado por *smartphone* na orientação retrato. Esta apresenta na parte superior da tela: o brasão da Prefeitura, o nome da Prefeitura e o menu de opções de gráficos retraído. No restante, encontra-se um gráfico de barras como principal elemento e um gráfico de pizza como secundário com ideia de rolagem vertical.

A próxima apresentação é focada no comportamento do menu de opções do projeto, conforme Figura 35.

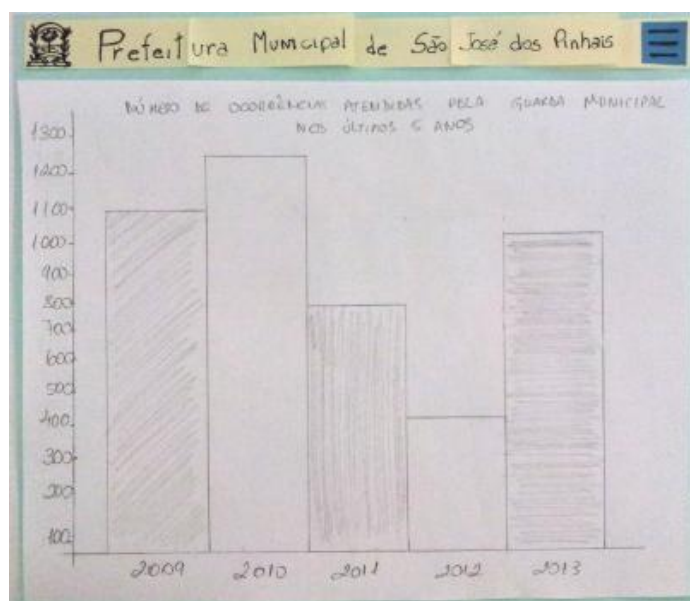


**Figura 35 - Esboço de interface para smartphone na orientação retrato com menu expandido**

**Fonte: Autoria própria**

Observa-se na apresentação da Figura 35, uma visão acessada por um *smartphone* na orientação retrato, onde apresenta na parte superior da tela: o brasão da Prefeitura, o nome da Prefeitura e o menu de opções de gráficos expandido para visualização dos gráficos disponíveis no sistema.

Já na Figura 36, o esboço de tela é de apresentado na simulação de um *tablet*.

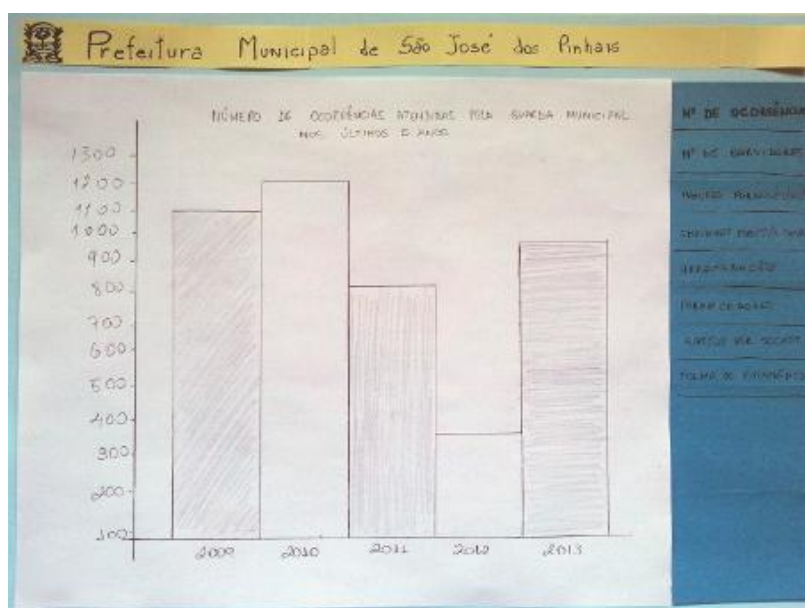


**Figura 36 - Esboço de interface para tablet na orientação paisagem com menu retraído**

**Fonte: Autoria própria.**

A visão interativa da Figura 36 é simulada por um *tablet* na orientação paisagem e apresentada da seguinte maneira: na parte superior da tela há o brasão da Prefeitura, o nome da Prefeitura e o menu de opções de modo retraído.

As Figuras 34, 35 e 36 apresentam o mesmo comportamento e interface para os dispositivos *tablets* e *smartphones* tanto nas orientações retrato quanto paisagem. Porém, a Figura 37, tem um comportamento bastante diferenciado quando acessado por resoluções maiores como *notebooks*, *desktops* e televisores.



**Figura 37 - Esboço de interface para grandes resoluções**

**Fonte: Autoria própria**

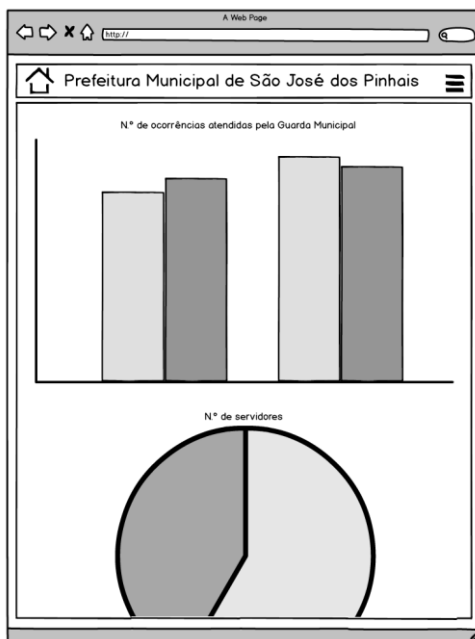
O comportamento dos elementos da Figura 37 é apresentado por uma visão acessada de dispositivos como *notebook*, *desktop* ou televisores. Esta é a versão do projeto piloto apresentada para grandes resoluções, que por sua vez é constituída de um brasão, juntamente com o nome da Prefeitura na parte superior da página e no conteúdo principal encontra-se um gráfico de barras seguido de um menu lateral, o mesmo menu que em resoluções menores era apresentado na parte superior em formato de um ícone quando retraído ou de uma lista horizontal quando expandido.

### 5.3.2. Projeto mockup

O projeto com *mockups* tem por objetivo apresentar visualizações aprimoradas dos protótipos de baixa fidelidade. Nesta Seção, as Figuras apresentadas mostram este tipo de recurso sendo utilizado. As características e

comportamentos das telas prevalecem.

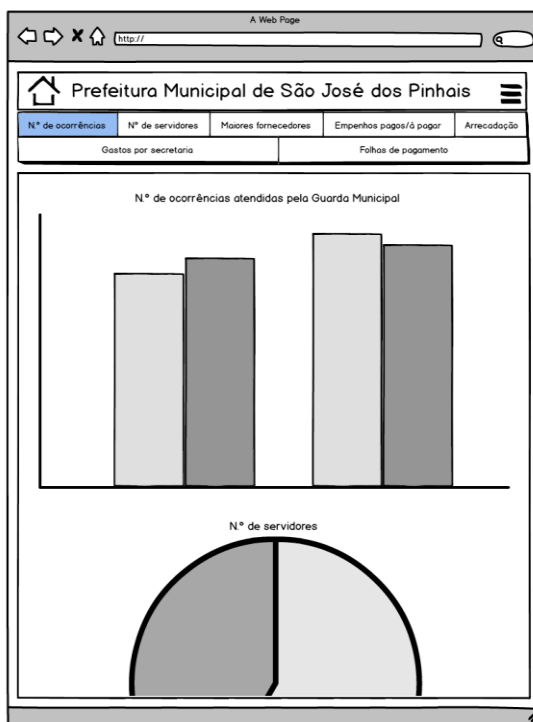
A interface para smartphones na orientação retrato é apresentada na Figura 38.



**Figura 38 - Mockup para smartphone na orientação retrato com menu retraído**

**Fonte: Autoria própria**

A próxima visualização, representada pela Figura 39, também foi elaborada para *smartphones*, porém com a exibição do menu de opções expandido.



**Figura 39 - Mockup para smartphone na orientação retrato com menu expandido**

Fonte: Autoria própria

Já a Figura 40, representa o projeto *mockup* na visão de um *tablet* na orientação paisagem.

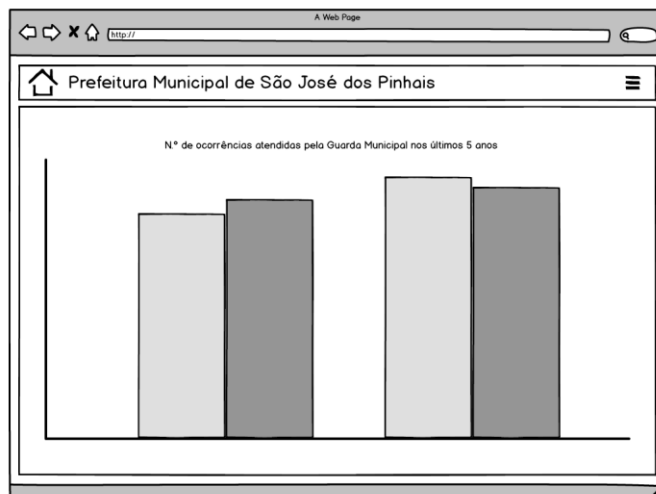


Figura 40 - Mockup para tablet na orientação paisagem com menu retraído

Fonte: Autoria própria

Para resoluções maiores como, por exemplo, *notebooks*, *desktops* e *televisores*, a Figura 41 é apresentada.

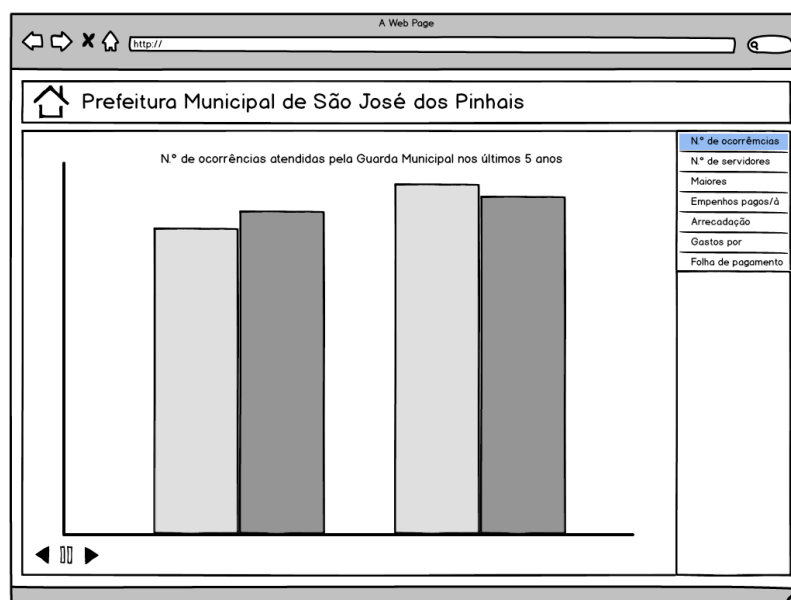


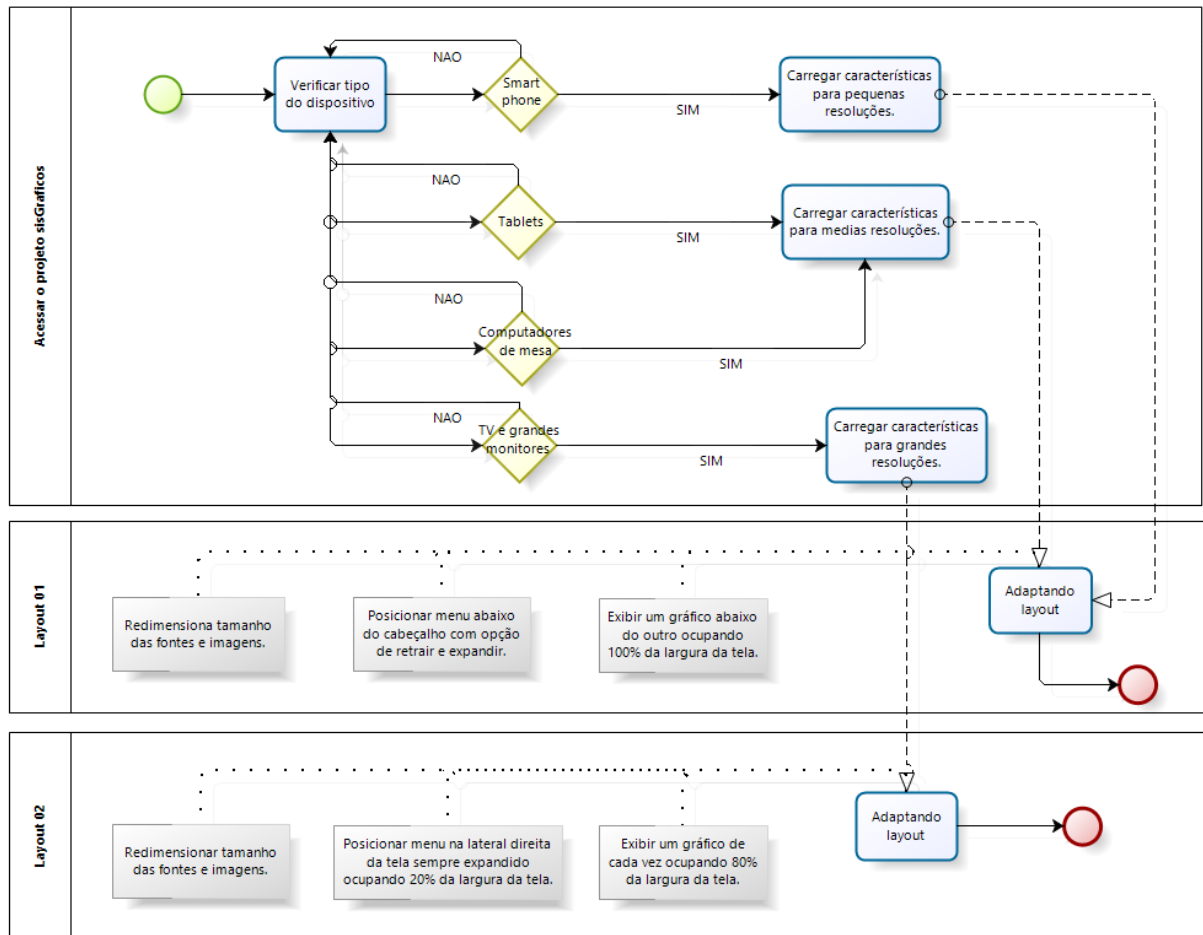
Figura 41 - Mockup para grandes resoluções

Fonte: Autoria própria

## 5.4. IMPLEMENTAÇÃO

Nesta Seção serão abordados os processos do *design* responsivo para desta forma exemplificar como ele pode ser inserido em um projeto. Também serão apresentados principais trechos de códigos para ilustrar a implementação realizada.

O fluxo número 42 representa a visão geral do comportamento do projeto para cada um dos tipos de dispositivos.



**Figura 42 - Comportamento do projeto para cada tipo de dispositivo**

Fonte: Autora própria

A identificação do tipo do dispositivo foi implementada da maneira como pode ser visualizado na Figura 43:

```
grafico.css
1
2 @media (min-width: 300px) {
3     /* mobile first - prepara layout para exibir menu retraído de duas colunas */
4 }
5
6 @media (min-width: 400px) {
7     /* aumentamos a fonte do título */
8 }
9
10 @media (min-width: 500px) {
11     /* aumentamos a fonte do título */
12 }
13
14 @media (min-width: 600px) {
15     /* prepara layout para exibir menu retraído de três colunas */
16     /* aumentamos a fonte do título */
17 }
18
19 @media (min-width: 780px) {
20     /* prepara layout para exibir menu retraído de quatro colunas */
21     /* aumentamos a fonte do título */
22 }
23
24
25 @media (min-width: 1025px) {
26     /* prepara layout para exibir menu expandido na lateral direita */
27     /* aumentamos a fonte do título */
28     /* exibimos um gráfico por vez */
29 }
30
31 @media tv {
32     /* prepara layout para exibir menu expandido na lateral direita */
33     /* este sera utilizado somente quando o projeto for acessado pelo navegador proprietário da TV */
34 }
```

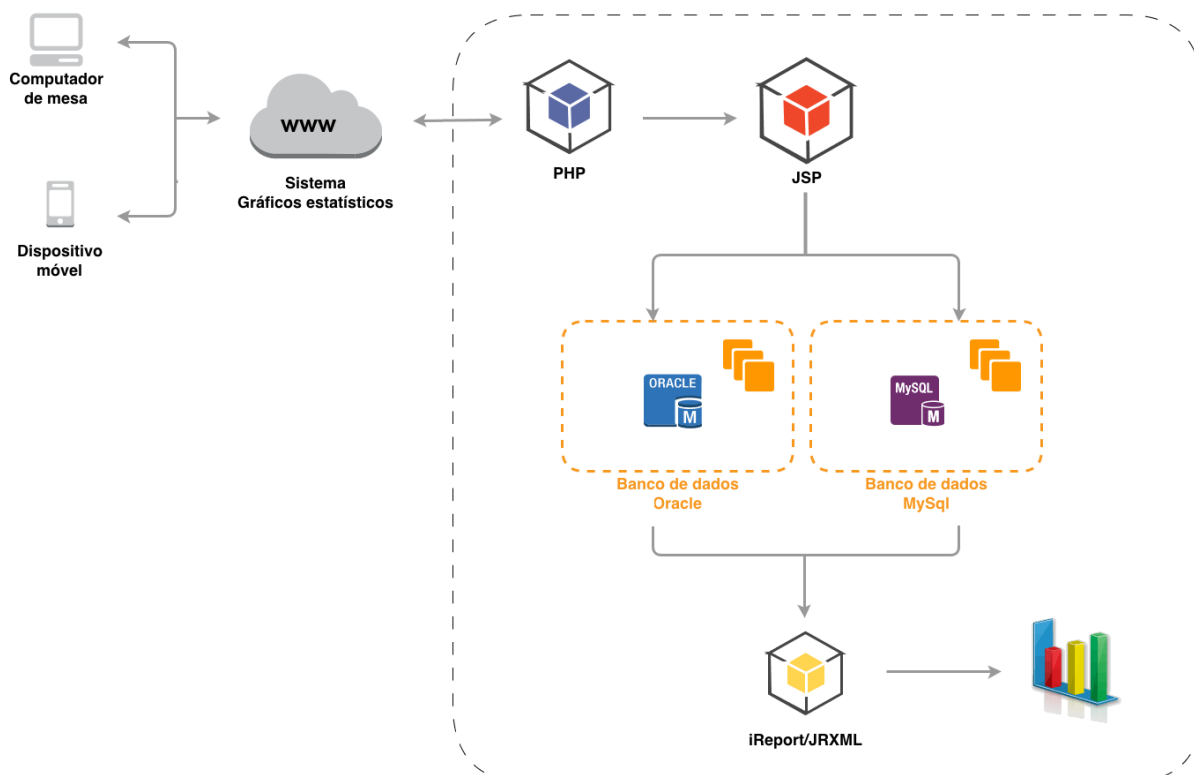
**Figura 43 - Implementação do CSS para identificar as resoluções**

**Fonte: Autoria própria**

Observa-se na Figura 43 que não há nenhuma cláusula verificando qual é o tipo específico do dispositivo, a técnica aplicada é chamada de *medias queries*, em que a propriedade `min-width` do CSS é especificada para o tamanho do navegador e não do dispositivo, deste modo, pode-se especificar a largura que bem necessitar.

Toda a implementação foi baseada em uma única versão, não há uma implementação separada para cada dispositivo, a implementação é única e, a estrutura geral do projeto resume-se no diagrama da Figura 44.





**Figura 44 - Estrutura geral do projeto**

**Autoria própria**

Primeiramente, a Figura 44, é dividida em duas áreas, uma representando a parte externa, onde o usuário faz a interação com o sistema e, outra representa a estrutura do projeto, definindo como os gráficos são gerados e acessados, para isso duas linguagens de programação são utilizadas: o PHP<sup>11</sup> e o JSP<sup>12</sup>.

Através de um dispositivo, seja ele móvel, um computador de mesa ou uma *smart TV*, o usuário acessa a página web do projeto, representada pelo ícone no formato de uma nuvem, que por sua vez faz uma requisição via PHP para o servidor de aplicação.

O PHP carrega todas as suas dependências necessárias, como CSS, JS e HTML para montar o *layout* do projeto.

Quando os gráficos são requisitados, um *script* PHP fica responsável por fazer uma solicitação ao JSP informando qual o gráfico a ser consultado. As consultas são realizadas pelo banco de dados Oracle ou MySQL.

Cada gráfico tem um arquivo de extensão *jrxml*<sup>13</sup> criado pelo programador através da ferramenta iReport.

<sup>11</sup> *Hypertext Preprocessor*, linguagem de programação para desenvolvimento web.

<sup>12</sup> *JavaServer Pages*, linguagem de programação para desenvolvimento web.

<sup>13</sup> Extensão de arquivos de layout do JasperReport baseado em xml.

Após a identificação do gráfico e do banco de dados, o JSP executa o arquivo jrxml do gráfico em questão. Este arquivo exporta os dados e imagens que por sua vez é retornado para o PHP e adicionado ao *layout* final.

## 5.5. DEMONSTRAÇÃO

Nesta subSeção serão apresentadas imagens comparativas do estudo de caso sem as técnicas de *design* responsivo versus o mesmo estudo com a aplicação da técnica. Cada comparativo é realizado sempre no mesmo modelo/marca de dispositivo para manter a lealdade de comparação.

As Figuras 45 e 46 apresentam testes obtidos através do dispositivo Ultrabook Dell Inspiron 14 polegadas. O navegador utilizado no comparativo foi o Google Chrome versão 29.0.1547.66m com resolução de 1366 x 600 pixels.

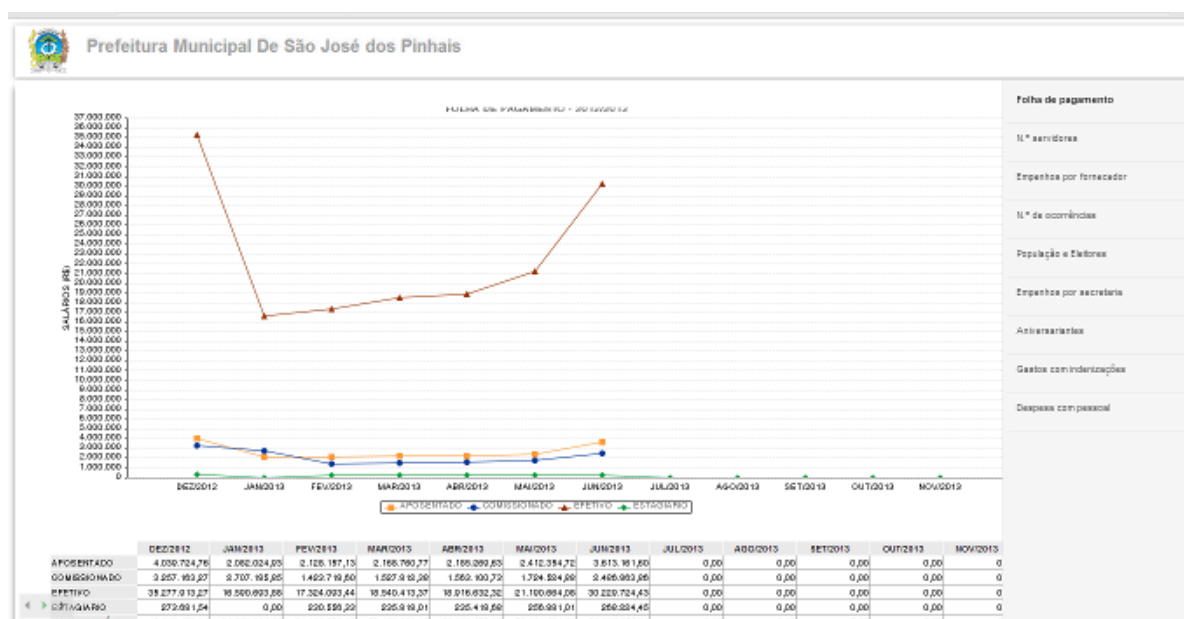


Figura 45 - Comparativo do layout sem design responsivo para notebooks

Fonte: Autoria própria

Na Figura 45 resulta no comparativo sem as técnicas de responsividade. Apresentou-se bom comportamento dos elementos, mas com um breve corte na apresentação dos dados ao fim do gráfico.

Já na Figura 46, as técnicas de *design* responsivo foram aplicadas e, todos os

dados foram apresentados na íntegra. Os valores de y do plano cartesiano, assim como o título do gráfico, apresentou-se de maneira mais legível se comparados à Figura 45.

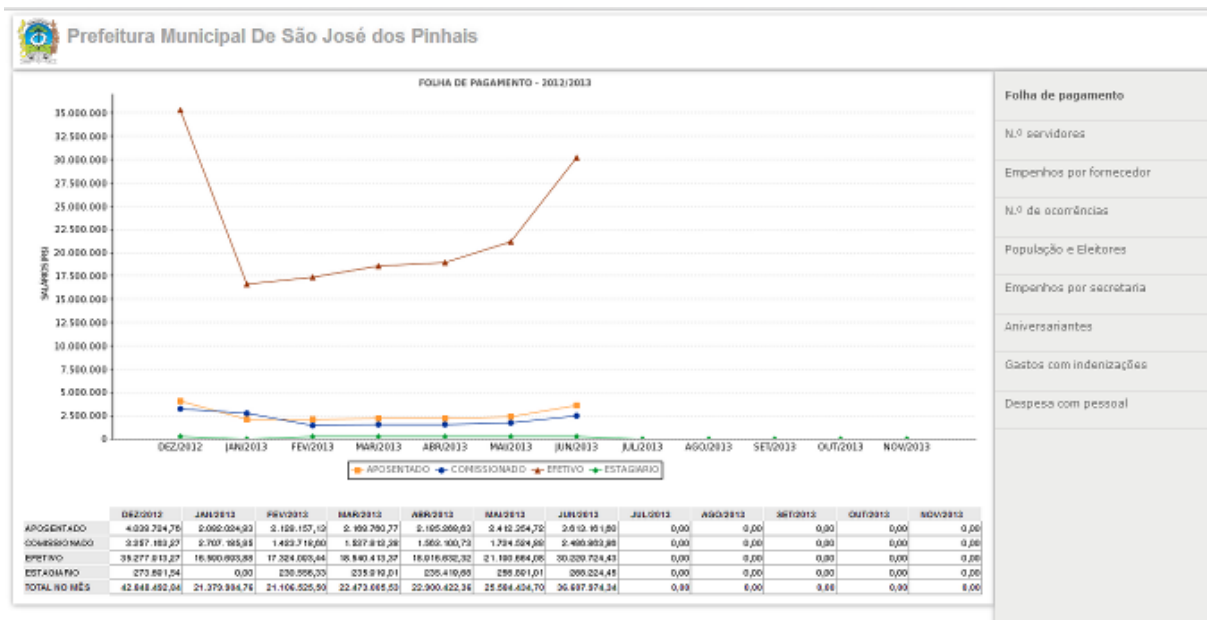


Figura 46 - Comparativo do layout com design responsivo para notebooks

Fonte: Autoria própria

Os testes realizados no simulador<sup>14</sup> web de iPad, são apresentados nas Figuras 47, 48 e 49.

<sup>14</sup> O simulador pode ser acessado através do endereço eletrônico: <http://ipadpeek.com/>

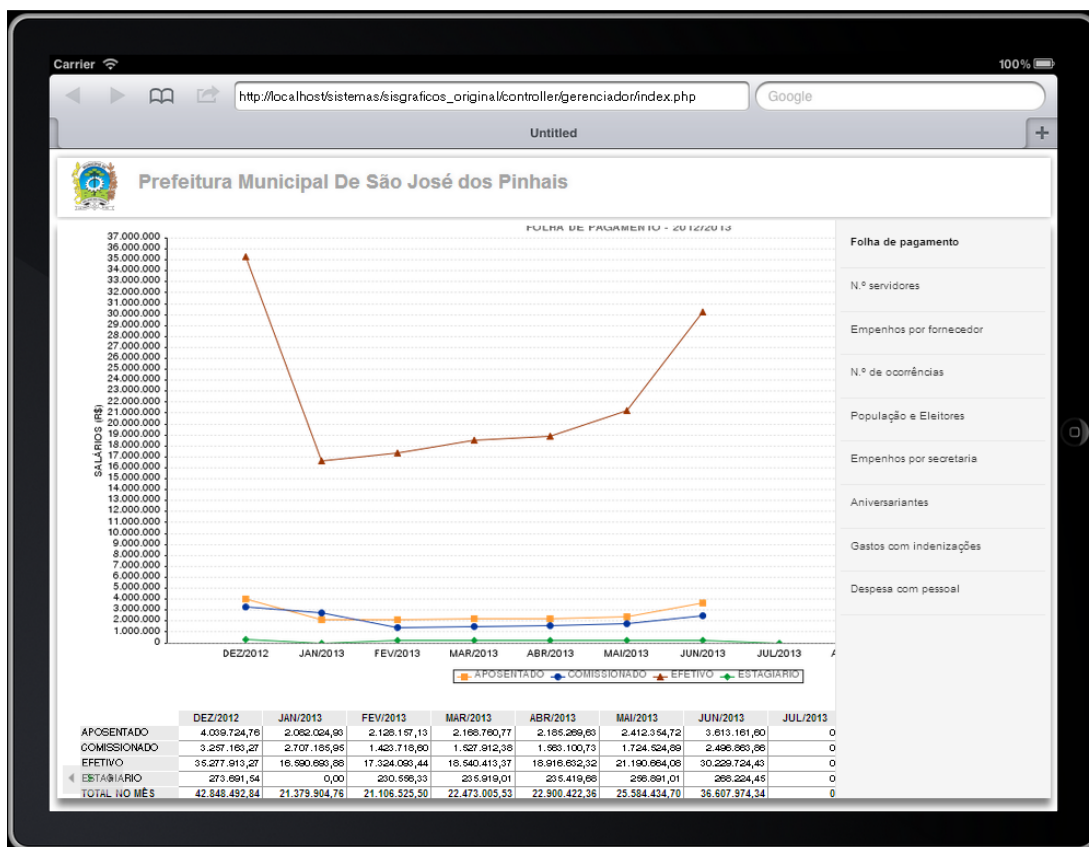
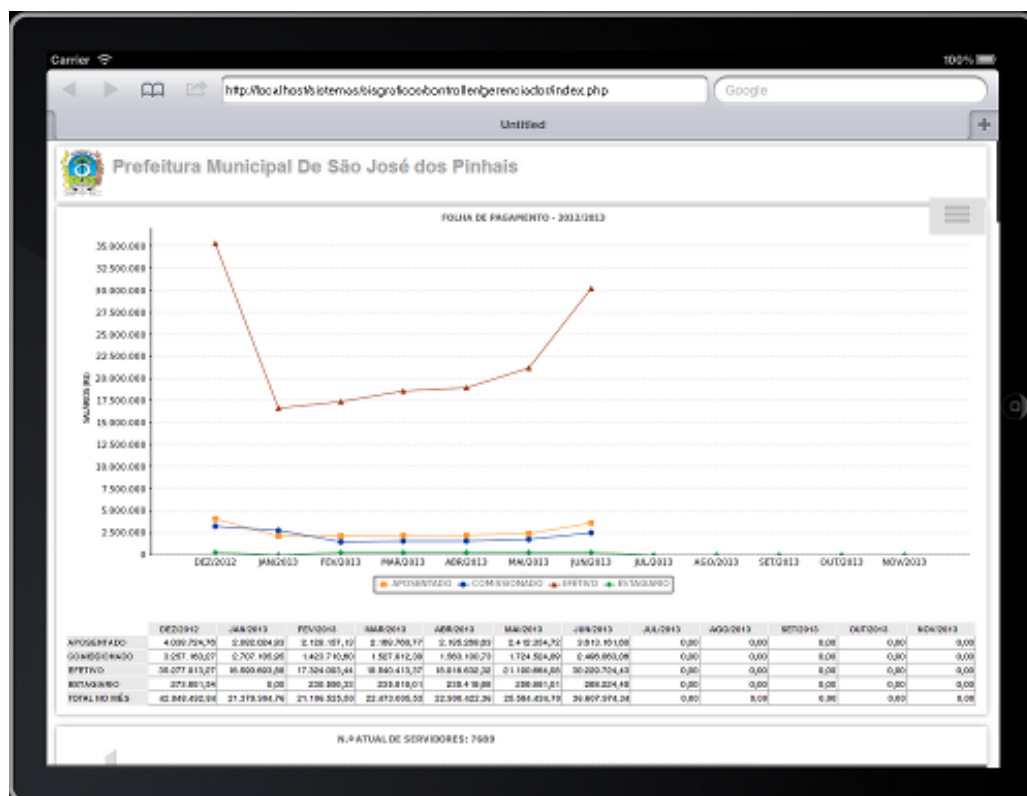


Figura 47 - Comparativo do layout sem design responsivo para tablets

Fonte: Autoria própria

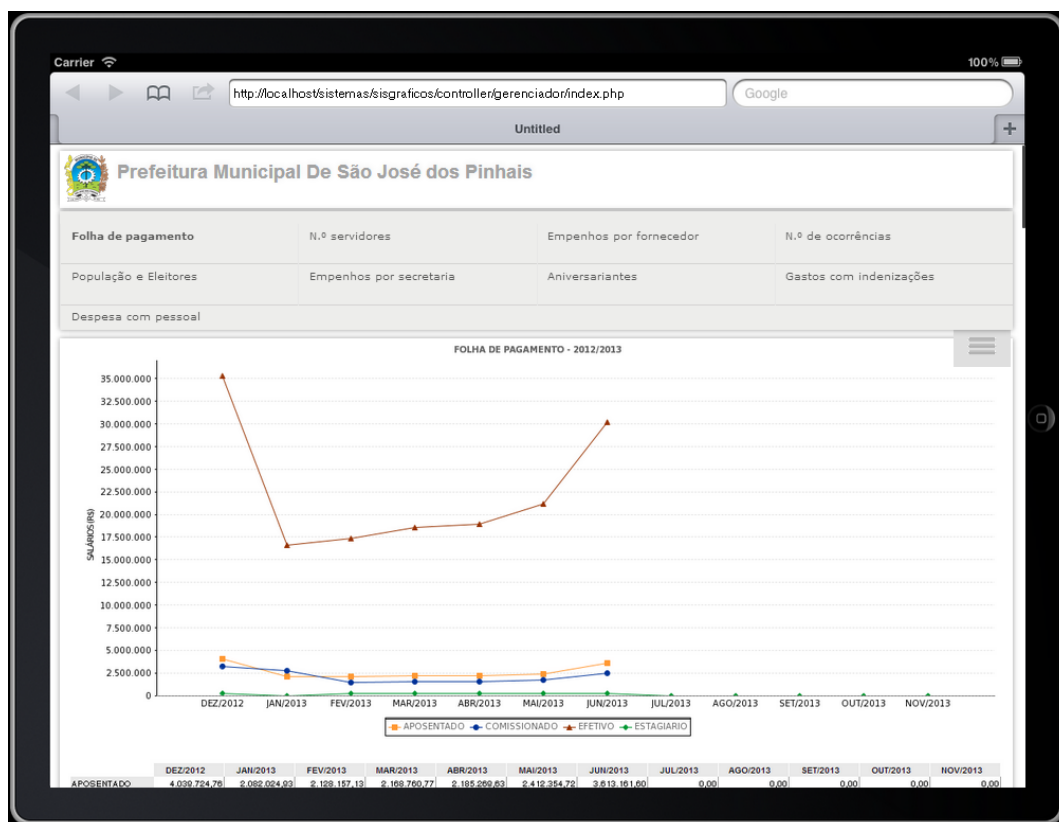
A Figura 47, representada sem as técnicas de responsividade, não só apresentou um corte no final dos dados, como também na lateral do gráfico. Percebe-se que este tipo de *layout* não é adequado para o dispositivo em questão. A largura disponível para manter os dados na íntegra, não é suficiente.



**Figura 48 - Comparativo do layout com design responsivo para tablets**

**Fonte: Autoria própria**

Aplicando a responsividade, observa-se na Figura 49 um *layout* melhor distribuído. O menu de opções deixa de ser exibido na lateral direita, dando mais espaço para os dados de relevância e, ganha apenas uma simbologia na parte superior direita da tela, que por sua vez representa-o de certa maneira. Deste modo há mais área disponível para utilização do gráfico. Nota-se também que há o início da apresentação do próximo gráfico ao final da página, que pode ser visualizado na íntegra, através da barra de rolagem do navegador.



**Figura 49 - Comparativo do layout com design responsivo para tablets – menu expandido**

**Fonte: Autoria própria**

A Figura 49 também apresenta as técnicas de *design* responsivo, idênticas à Figura 48, porém esta simula o clique na simbologia do menu, tornando suas opções visíveis. Observa-se também, que para o dispositivo em questão, as opções do menu são dispostas em quatro colunas.

As mesmas características e comportamentos são aplicados para dispositivos com navegadores de resoluções intermediárias, como por exemplo: Galaxy Note 8.0 GT-N5100<sup>15</sup> da Samsung, medindo aproximadamente 211 x 136 x 8 milímetros e o Galaxy Note II 4G GT-N7105T<sup>16</sup> também da Samsung com dimensões de 151.1 x 80.5 x 9.4 milímetros.

As Figuras 50, 51 e 52 apresentam testes obtidos através do simulador web para iPhone5 com orientação paisagem.

<sup>15</sup> Galaxy Note 8.0 - Disponível em <http://www.samsung.com/br/consumer/cellular-phone/cellular-phone-tablets/galaxy-note/GT-N5100ZWPZTA-spec>

<sup>16</sup> Galaxy Note II 4G - Disponível em: <http://www.samsung.com/br/consumer/cellular-phone/cellular-phone-tablets/galaxy-note/GT-N7105RWPZTA-spec>



**Figura 50 - Comparativo do layout sem design responsivo para smartphones**

**Fonte: Autoria própria**

Representada sem as técnicas de responsividade, a Figura 50 apresentou um mau aproveitamento de tela, nota-se que o menu lateral ocupou praticamente metade da área do navegador, o gráfico ficou ilegível e o único elemento que ganhou destaque foi o cabeçalho da página, informação esta de baixa relevância.



**Figura 51 - Comparativo do layout com design responsivo para smartphones**

**Fonte: Autoria própria**

Já com a responsividade, a Figura 51 proporcionou um *layout* mais agradável e legível. As opções do menu estão retraídas e representadas por uma imagem menor e numa posição intuitiva, dando mais visibilidade ao gráfico. E mesmo o gráfico não apresentando suas informações por completo, uma barra de rolagem vertical é gerada para que se possa navegar entre as imagens.

A Figura 52 a seguir, expõe as opções do menu referente à Figura 51.



**Figura 52 - Comparativo do layout com design responsivo para smartphones – menu expandido**

**Fonte: Autoria própria**

Quando as opções de menu são expandidas, estas são divididas em duas colunas, posicionadas na parte superior da página. Aparentemente tem-se a impressão de que o menu está incompleto e que os gráficos foram ocultados, mas há uma barra de rolagem vertical para acompanhar os dados na íntegra, como mostra a Figura 52.

Para o dispositivo iPhone5 na orientação retrato, também utilizando o simulador<sup>17</sup> web, obteve-se os seguintes efeitos apresentados nas Figuras 53, 54 e 55.

---

<sup>17</sup> Simulador disponível em: <http://ipadpeek.com/>





**Figura 53 - Comparativo do layout sem design responsivo para smartphones – orientação retrato**

**Fonte: Autoria própria**

Assim como na orientação paisagem, exibida na Figura 50, a falta das técnicas de *design* responsivo, permitiu que o menu de opções abrangesse quase toda a área de visualização do gráfico, tornando-o bastante ilegível, como mostra a Figura 53.



**Figura 54 - Comparativo do layout com design responsivo para smartphones – orientação retrato**

**Fonte: A autoria própria**

No entanto, a Figura 54, exibe o resultado dos testes com a implementação das técnicas responsivas. Observa-se que os elementos como o cabeçalho da página, tem tamanho proporcional e seus subelementos são ajustados adequadamente conforme a área disponível. O menu de opções está retraído e representado por uma imagem na parte superior direita da tela. Os gráficos são proporcionalmente redimensionados e exibidos um abaixo do outro também com opção de barra de rolagem vertical.

Caso o usuário ainda tenha dificuldade de analisar os valores dos gráficos, a opção padrão de *zoom* dos dispositivos móveis, comumente conhecidos como movimentos de pinça, estará disponível para ser utilizado.



**Figura 55 - Comparativo do layout com design responsivo para smartphones – orientação retrato, menu expandido**

**Fonte: Autoria própria**

Na Figura 55, os testes simularam o menu de opções expandido e dividido em duas colunas. Nesta, a responsividade também foi aplicada e as demais características e comportamentos dos elementos são reflexos da Figura 54.

Nas próximas seções, Seção 6 e 7, serão apresentadas os resultados obtidos no estudo de caso desenvolvido neste trabalho e suas considerações finais.

## 6 RESULTADOS OBTIDOS

Esta Seção apresenta os experimentos realizados e os resultados obtidos no estudo de caso apresentado neste trabalho.

Primeiramente observou-se um resultado bastante satisfatório na aplicação das técnicas de *design* responsivo, *mobile-first* e *viewport*, já que estas técnicas abrangem grande parte da solução do problema.

No início do desenvolvimento da adaptação da aplicação, a metodologia *mobile-first* foi descartada, porém no decorrer da adaptação do layout de desktop para *mobile*, identificou-se uma perda bastante considerável de tempo em relação ao desmembramento dos elementos de tela. Como os elementos já existiam e estes não foram planejados para serem flexíveis, estruturalmente comportavam-se de maneira dependente, ou seja, qualquer alteração implementada em um elemento, interferia em outro, pois os mesmos eram dependentes entre si.

Decidiu-se então aplicar a metodologia para identificar quais elementos do "todo" teriam características e comportamentos diferentes em dispositivos menores e quais seriam os "futuros" comportamentos destes elementos. A prática foi abordada no sentido do menor para o maior e a partir daí os resultados começaram a surgir de maneira mais eficiente.

O quadro 1 identifica os resultados obtidos, usados nas técnicas de *design* responsivo, referente às características e os comportamentos que cada elemento sofre conforme a mudança do tamanho do navegador do dispositivo:

Resolução	Descrição	Comportamento dos elementos de tela
1. 300px	Dispositivos com resoluções a partir de 300 pixels de largura.	Cabeçalho: a fonte do título principal é reduzida; Menu: as opções de menu são retraídas e simbolicamente representado por uma imagem pequena e intuitiva posicionada na parte superior a direita do navegador; Menu aberto: as opções de menu são expandidas e divididas em duas colunas posicionadas na parte superior do navegador ocupando 100% de largura e altura calculada <sup>18</sup> conforme a

<sup>18</sup> Em alguns casos houve a necessidade de complementar as técnicas de design responsivo com o uso da tecnologia jQuery para atender a necessidade de modificar o comportamento de alguns elementos de tela em tempo de execução.

		<p>quantidade de itens contidos na lista de menus;</p> <p>Conteúdo: a área disponibilizada para os gráficos ocupa 100% da largura do navegador; Todos os gráficos (imagens) disponíveis são exibidos um abaixo do outro, caso exista quantidades suficiente para aumentar a altura da página, este é feito automaticamente pelo navegador, usando scroll vertical; O tamanho de cada gráfico é proporcionalmente ajustado conforme a largura da área do conteúdo.</p>
2. 400px	Dispositivos com resoluções a partir de 400 pixels de largura.	<p>Cabeçalho: a fonte do título principal é ampliada de 1.2em para 1.3em.</p> <p>As demais características e comportamentos são mantidos como descritos no item anterior.</p>
3. 500px	Dispositivos com resoluções a partir de 500 pixels de largura.	<p>Cabeçalho: a fonte do título principal é ampliada de 1.3em para 1.4em.</p> <p>As demais características e comportamentos são mantidos como descritos no item 01.</p>
4. 600px	Dispositivos com resoluções a partir de 600 pixels de largura.	<p>Cabeçalho: a fonte do título principal é ampliada de 1.4em para 1.5em.</p> <p>Menu aberto: as opções de menu são expandidas e divididas em três colunas.</p> <p>As demais características e comportamentos são mantidos como descritos no item 01.</p>
5. 780px	Dispositivos com resoluções a partir de 780 pixels de largura.	<p>Cabeçalho: a fonte do título principal é ampliada de 1.5em para 1.6em.</p> <p>Menu aberto: as opções de menu são expandidas e divididas em quatro colunas.</p> <p>As demais características e comportamentos são mantidos como descritos no item 01.</p>
6. 1025px	Dispositivos com resoluções a partir de 1025 pixels de largura.	<p>Cabeçalho: a fonte do título principal é ampliada de 1.6em para 1.7em.</p> <p>Menu: a simbologia do menu retraído é ocultado; As opções de menu são sempre exibidas e em uma única coluna posicionada na lateral direita do navegador ocupando 18,4% de largura da área total. Caso a altura não seja o suficiente para a quantidade de itens do menu, o navegador gera automaticamente um scroll vertical.</p> <p>Conteúdo: a área disponibilizada para os gráficos ocupa 81.5% da</p>

		largura do navegador; Apenas um gráfico é exibido por vez. O tamanho do gráfico é proporcionalmente ajustado conforme a largura da área do conteúdo.
7. 1400px	Dispositivos com resoluções a partir de X pixels de largura.	Conteúdo: O tamanho do gráfico não excede seu tamanho original, evitando assim, que a imagem sofra distorções ao ultrapassar seu tamanho máximo; Toda a página é centralizada.
8. TV	Resolução específica para TV (recursos limitados)	Comporta-se idem ao item 06 deste quadro.

**Quadro 1 - Variações das características e comportamentos do layout conforme dispositivo móvel**  
**Fonte: Autoria própria**

Observou-se que estas dimensões obedecem a uma ordem sequencial de leitura, logo, para se obter os resultados corretos, estes foram implementados na mesma ordem que descritos acima.

Alguns testes também foram realizados acessando o projeto através de uma TV, mas especificamente uma *smart* TV (LG 50PM4700 47 polegadas).

O primeiro teste foi realizado acessando o projeto através do navegador da própria televisão, este por sua vez comportou-se do modo esperado, como descrito no item 08 da tabela acima.

O segundo caso foi apresentado através da projeção da imagem do ultrabook Dell Inspiron 14 polegadas, utilizando um cabo HDMI de transmissão, o resultado foi idêntico ao acesso pelo ultrabook, como descreve o item 06 da tabela acima.

A Figura 56 mostra o resultado obtido acessando a partir do navegador proprietário da TV.

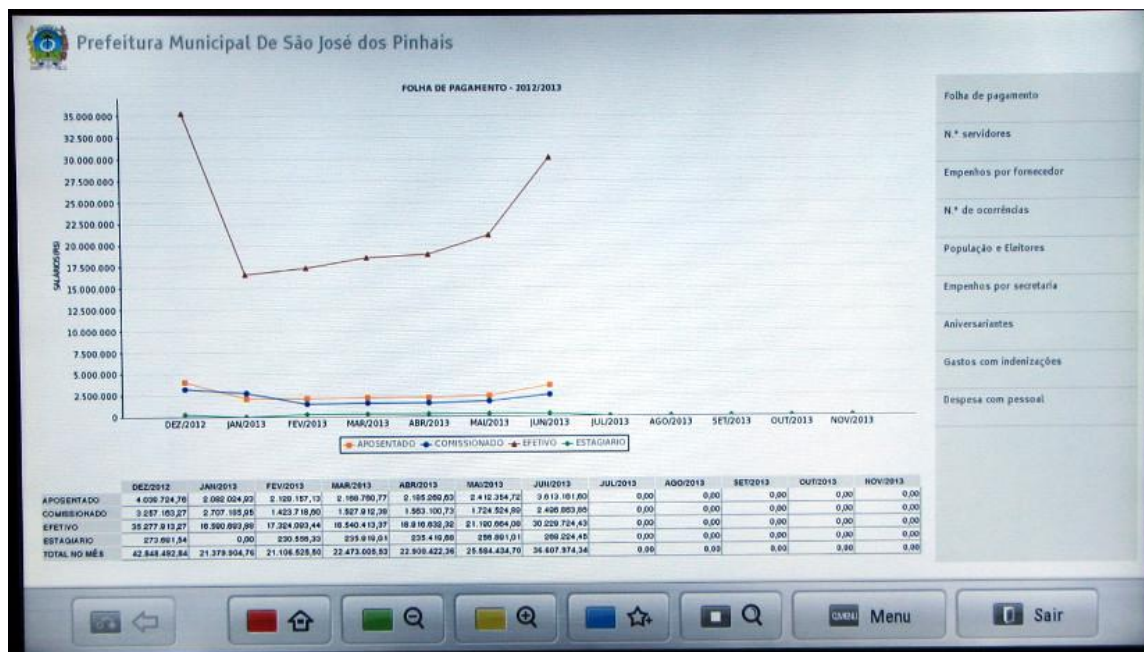


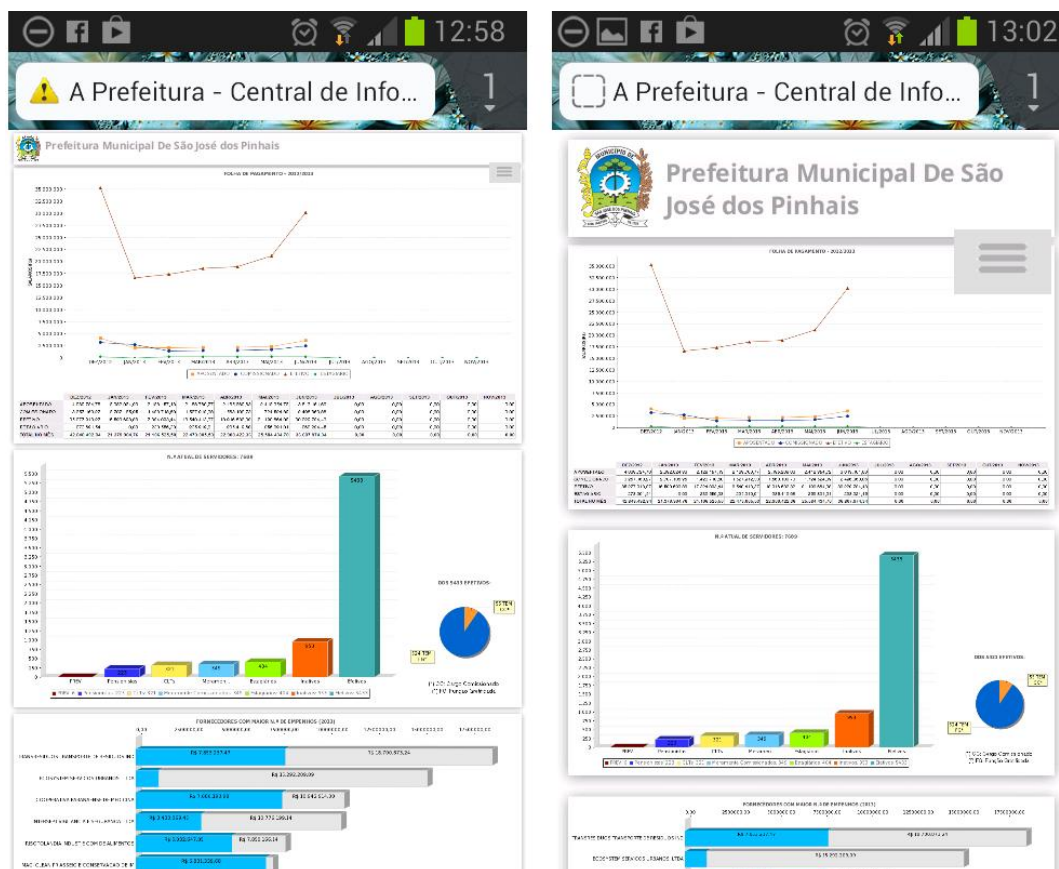
Figura 56 - Resultados obtidos a partir do navegador proprietário de uma smart TV

Fonte: Autoria própria

Outro resultado interessante foi o uso da *metatag viewport*, esta técnica implica diretamente na forma como o navegador renderiza a página web. O **não** uso desta técnica reduz o tamanho de todos os elementos da página para que se possa vê-los em um navegador menor.

Usando o *viewport* a resolução do navegador terá a mesma resolução de tela do dispositivo, deixando o conteúdo mais agradável de visualizar.

As Figuras 57 e 58 resultam num comparativo do protótipo com e sem a aplicação do *viewport* em um mesmo dispositivo - Samsung Galaxy S2 GT-I9100.



**Figura 57 - Comparativo do layout sem viewport (esquerda) e com viewport (direita)**

Fonte: Autoria própria

Onde a Figura 57 apresenta ao lado esquerdo, o *layout* acessado por um *smartphone* sem a implementação do *viewport* e, ao lado direito, o comparativo com *viewport*.





Figura 58 - Comparativo do menu sem viewport (esquerda) e com viewport (direita)

Fonte: Autoria própria

Onde a Figura 58 apresenta, ao lado esquerdo, o menu sem a implementação do *viewport* e, ao lado direito com o *viewport*.

Nota-se no comparativo ilustrado pelas Figuras 57 e 58 a redução do tamanho dos elementos como: fonte, imagens e menus sem o uso do *viewport* e o melhoramento visual com a especificação da *metatag viewport*.

Agupando todas essa técnicas e metodologias, podemos observar que o resultado atendeu o objetivo esperado, conforme Figura 59.

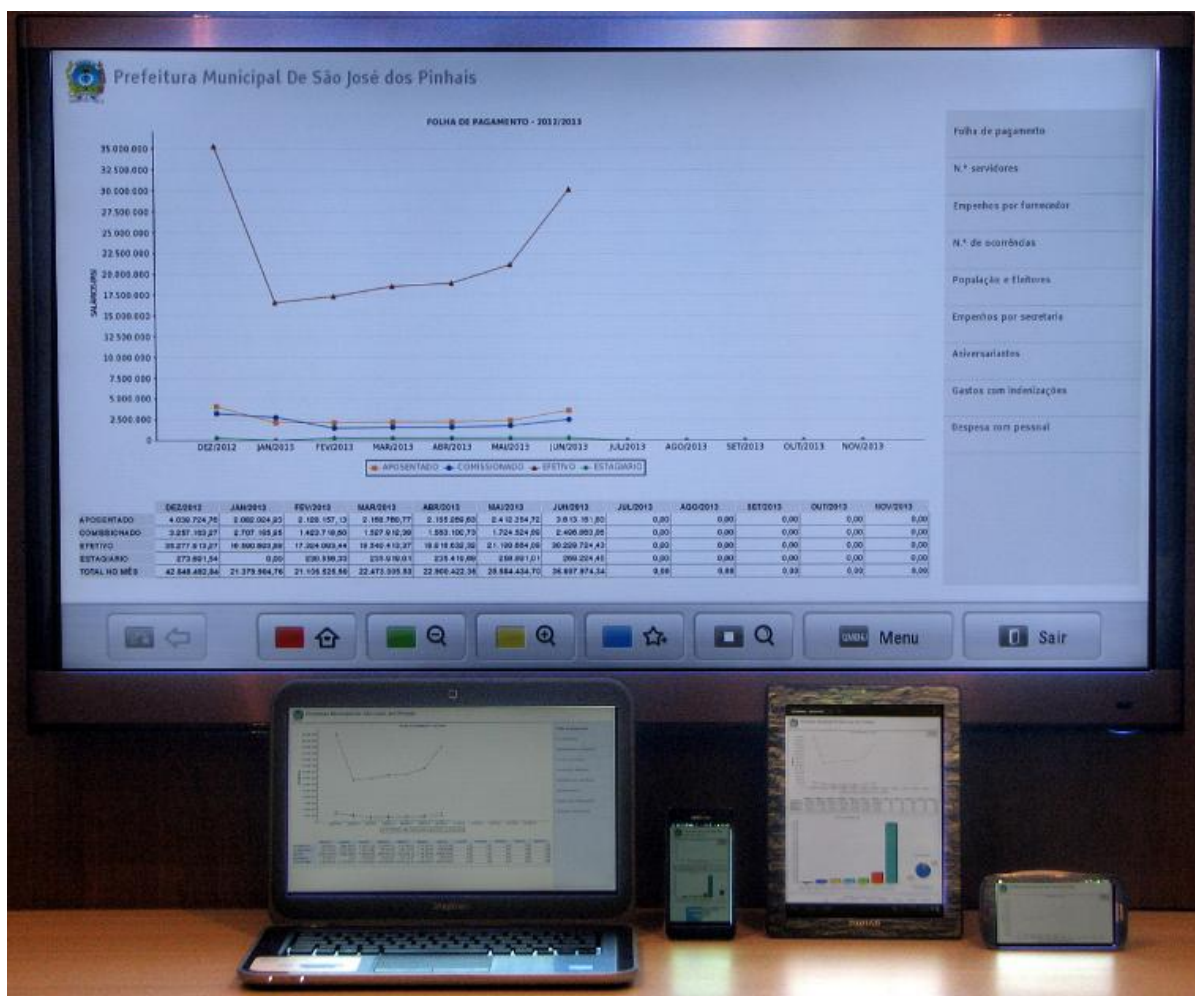


Figura 59 - Resultados obtidos em diversos dispositivos

Fonte: Autoria própria

Com relação às dificuldades encontradas e os resultados não satisfatórios, pode-se relatar que a ferramenta iReport, utilizada para criar os gráficos, exporta o trabalho final para vários tipos de arquivos (HTML, PDF, Excel, OpenOffice e Word), o formato escolhido para uso no projeto piloto foi o HTML, porém, ao usar as técnicas do *design* responsivo, para que os gráficos redimensionassem conforme a necessidade, não houve o resultado esperado, o redimensionamento do gráfico não ocorria, porque este era um HTML e necessariamente era preciso somente da imagem do gráfico. Solução: foi extraído do HTML a imagem do gráfico, porem este recurso foi feito manualmente, o ideal é procurar um recurso automático para fazer esta extração, podendo ser até uma implementação no próprio *servlet*, ou analisar a ferramenta iReport para exportar o trabalho final como imagem.

## 7 CONSIDERAÇÕES FINAIS

O propósito deste trabalho foi o de adaptar aplicações web, por meio de tecnologias e metodologias, que pudesse ser acessada por qualquer tipo de dispositivo eletrônico que tenha acesso a Internet, sendo eles: *smartphones*, *tablets*, computadores de mesa e televisores, sem ocorrer a perda de informações, o corte de imagens, a legibilidade dos dados e principalmente sem a necessidade de criar múltiplas versões de código fonte para cada dispositivo.

Para este fim, os assuntos teóricos que mais apontou para este tipo de abordagem, foi o tema *Design* responsivo. A partir deste descobriu-se outros temas, não menos importante como *Mobile first* e *Viewport*. Os três temas foram aplicados no estudo de caso deste trabalho.

Na fase de desenvolvimento, percebeu-se a importância da abordagem *mobile first*, assunto que no início foi descartado erroneamente por já existir um projeto web desenvolvido para *desktop*.

Também foi identificado, durante a implementação, a importância de se ter uma mesma imagem "replicada" no repositório, mas de tamanhos diferentes para atender dispositivos diferentes, minimamente uma imagem pequena para smartphones, outra de tamanho médio para tablets e uma maior para computadores de mesa, este é um conceito que pode ser aplicado para melhorar a desempenho da página web, item importante para dispositivos pequenos que normalmente tem menos recurso que um computador convencional.

O motivo de realizar este trabalho foi, além do interesse pelo assunto, a carência no mercado, ainda há muitas páginas web sem o mínimo de recurso para serem flexíveis e diante disso pode-se concluir que não é preciso soluções grandiosas e implementações complexas, o que o mercado de linguagens web oferece é o suficiente para se ter uma boa página web funcionando nos inúmeros dispositivos móveis existentes.

Por fim, cabe ressaltar que uma das principais observações, se não a mais importante, é o fato de que a web foi projetada para ser flexível desde o seu começo, o que tem de mudar é a cultura dos desenvolvedores e projetistas, um projeto web deve ser único e flexível desde a sua concepção.

## REFERÊNCIAS

ALMEIDA, R.; TERRA, J. C. C. **Varejo 2.0 - Um Guia Para Aplicar Redes Sociais Aos Negócios**. Rio de Janeiro, RJ - Brasil: Campus, 2011. 200p.

ARRIETA, T. V. **Uma filosofia do celular ou os avatares que este meio de comunicação está introduzindo em nossas vidas. Uma filosofia do celular**. Comunicação, Mídia e Consumo, São Paulo, SP - Brasil, v.5, n.12, p.151-166, 2008. Disponível em: <<http://www.revistas.univerciencia.org/index.php/comunicacaomidiaeconsumo/article/view/6863/6199>> Acesso em: 12 jul. 2013

DARIVA, R. **Gerenciamento de Dispositivos Móveis E Serviços de Telecom**. São Paulo, SP - Brasil: Campus, 2011. 160p.

FÉ, A. L. D. M. **Tecnologias móveis e vida pessoal. Uma pesquisa sobre o impacto da comunicação sem fio no tempo de trabalho e nas demais esferas da vida social**. 2008. 163f. Tese (Doutorado em Comunicação) - Pontifícia Universidade Católica de São Paulo, São Paulo, SP - Brasil, 2008. Disponível em: <[http://www.sapientia.pucsp.br/tde\\_arquivos/1/TDE-2008-06-17T08:11:30Z-5667/Publico/Ana%20Lucia%20Damasceno%20Moura%20Fe.pdf](http://www.sapientia.pucsp.br/tde_arquivos/1/TDE-2008-06-17T08:11:30Z-5667/Publico/Ana%20Lucia%20Damasceno%20Moura%20Fe.pdf)> Acesso em: 1 jan. 0001

FISHER, A.; SHARKIE, C. **Jump Start Responsive Web Design**. Victório - Austrália: SitePoint, 2013. 145p.

GOMES, A. L. **XHTML/CSS, Criação de Páginas Web**. São Paulo, SP - Brasil: Senac, 2010. 195p.

JOBSTRAIBIZER, F. **Criação de sites com CSS, desenvolva páginas Web mais leves e dinâmicas em menos tempo**. São Paulo, SP - Brasil: Digerati Books, 2009. 144p.

LOPES, S. **A Web Mobile. Programe para um mundo de muitos dispositivos.** São Paulo, SP - Brasil: Casa do Código, 2013. 217p.

LOUREIRO, A. A. F.; MATEUS, G. R. **Introdução à Computação Móvel.** 1. ed. Rio de Janeiro, RJ - Brasil: 11º Escola de Computação , 1998. 189p.

MARCOTTE, E. **Responsive Web Design.** New York, NY - USA : A book apart, 2011. 150p.

PELLANDA, E. C. **Comunicação móvel: das potencialidades aos usos e aplicações.** Em *Questão*, Brasil, v.15, n.1, p.89-98, 2009. Disponível em: <<http://www.revistas.univerciencia.org/index.php/revistaemquestao/article/view/6627/6010>> Acesso em: 12 jul. 2013

TELESP CELULAR. **A revolução da mobilidade. O celular no Brasil. De símbolo de status a instrumento de cidadania.** São Paulo, SP - Brasil: Plano Editorial, 2002. 136p.

WROBLEWSKI, L. **Mobile First.** New York, NY - USA: A Book Apart, 2011. 123p.

ZEMEL, T. **Web Design Responsivo: Páginas adaptáveis para todos os dispositivos.** São Paulo, SP - Brasil: Casa do código, 2013. 150p.

ANATEL, **Acesso dos Serviços Móveis Pessoais.** Disponível em: <<http://www.anatel.gov.br/Portal/verificaDocumentos/documento.asp?numeroPublicacao=296023&pub=principal&filtro=1&documentoPath=296023.pdf>> Acesso em: 12 jul. 2013

e-Gov, **Governo Eletrônico Brasileiro.** Disponível em: <<http://www.governoeletronico.gov.br/o-gov.br>> Acesso em: 19 mai. 2013

Master Studio Web, **Imagem demonstrando o comportamento de um site responsivo.** Disponível em: <<http://masterstudioweb.com/blog/wp->

content/uploads/2012/08/design\_responsivo.jpg> Acesso em: 14 jul. 2013

Google. **Nosso Planeta Mobile: Brasil. Como entender o usuário de celular.**

Disponível em:

<[http://services.google.com/fh/files/blogs/our\\_mobile\\_planet\\_brazil\\_pt\\_BR.pdf](http://services.google.com/fh/files/blogs/our_mobile_planet_brazil_pt_BR.pdf)>

Acesso em: 13 jul. 2013

SEDS DH. **Secretaria de Desenvolvimento Social e Direitos Humanos – Governo do Estado de Pernambuco** – Edital de Credenciamento de Pessoas com

Deficiência Auditiva. Disponível em:

<[http://www.portais.pe.gov.br/c/document\\_library/get\\_file?uuid=da8d54b6-1dc6-4c8b-8344-a7ef452e8d93&groupId=17459](http://www.portais.pe.gov.br/c/document_library/get_file?uuid=da8d54b6-1dc6-4c8b-8344-a7ef452e8d93&groupId=17459)> Acesso em: 25 jun. 2013

WCAG. **Web Content Accessibility Guidelines.** Disponível em:

<<http://www.w3.org/TR/WCAG20/>> Acesso em: 14 jul. 2013

MAZZA, L. **HTML5 e CSS3, Domine a Web do futuro.** São Paulo, SP - Brasil: Casa do código, 2012. 205p.