

**UNIVERSIDADE REGIONAL DE BLUMENAU**  
**CENTRO DE CIÊNCIAS EXATAS E NATURAIS**  
**CURSO DE CIÊNCIA DA COMPUTAÇÃO – BACHARELADO**

**USANDO RESPONSIVE WEB DESIGN PARA**  
**DESENVOLVIMENTO DE SISTEMAS WEB**

**RODRIGO EDUARDO BONI**

**BLUMENAU**  
**2012**

**2012/2-24**

**RODRIGO EDUARDO BONI**

**USANDO RESPONSIVE WEB DESIGN PARA  
DESENVOLVIMENTO DE SISTEMAS WEB**

Trabalho de Conclusão de Curso submetido à Universidade Regional de Blumenau para a obtenção dos créditos na disciplina Trabalho de Conclusão de Curso II do curso de Ciência da Computação — Bacharelado.

Prof. Jhony Alceu Pereira - Orientador

**BLUMENAU  
2012**

**2012/2-24**

# **USANDO RESPONSIVE WEB DESIGN PARA DESENVOLVIMENTO DE SISTEMAS WEB**

Por

**RODRIGO EDUARDO BONI**

Trabalho aprovado para obtenção dos créditos na disciplina de Trabalho de Conclusão de Curso II, pela banca examinadora formada por:

Presidente: \_\_\_\_\_  
Prof. Jhony Alceu Pereira – Orientador, FURB

Membro: \_\_\_\_\_  
Prof. Jacques Robert Heckmann, Mestre – FURB

Membro: \_\_\_\_\_  
Prof. Marcel Hugo, Mestre – FURB

Blumenau, 11 de dezembro de 2012

Dedico este trabalho a todos que acreditaram  
na capacidade de minha pessoa.

## **AGRADECIMENTOS**

A Deus, pelo seu imenso amor e graça.

À minha família e namorada, que sempre estiveram presentes.

Aos meus amigos, pela amizade e confiança.

Ao meu orientador, Jhony Alceu Pereira, por ter acreditado na conclusão deste trabalho.

Por fim, aos demais professores que contribuíram para minha formação.

Nunca desista sem antes ter tentado. E nunca tente pensando em desistir. Determinação é a chave de tudo.

Autor Desconhecido

## RESUMO

O *Responsive Web Design* (RWD) surgiu com o intuito de tornar-se um conceito fundamental na elaboração de sistemas web, pois sua principal preocupação é lidar com a diversidade e o crescente número de dispositivos com acesso à Internet. Em outras palavras, visa apresentar corretamente o conteúdo web de maneira automática e adaptável. Sendo assim, o objetivo deste trabalho é demonstrar um sistema web que aplique o RWD, potencializando-o em termos de escalabilidade e disponibilidade com a computação em nuvem da plataforma Windows Azure (WA). Como estudo de caso utiliza-se a Nota Fiscal eletrônica (NF-e).

Palavras-chave: *Responsive web design*. Sistemas adaptativos. Computação em nuvem. Windows Azure.

## **ABSTRACT**

The Responsive Web Design came up with the intention of becoming a key concept in the web systems development, because its main concern is dealing with diversity and the growing number of devices with Internet access. In other words, it aims to present the correct web content automatically and adaptable. Therefore, this work objective is to demonstrate a web system that applies the RWD, increasing it in terms of scalability and availability with cloud computing platform Windows Azure. As a case study the issue of electronic invoice is used.

Key-words: Responsive web design. Adaptive systems. Cloud computing. Windows Azure.



## LISTA DE ILUSTRAÇÕES

Figura 1 – Exemplo de um sistema web que aplica o conceito do RWD.....	18
Figura 2 – Exemplo de um elemento com unidade percentual calculada sobre um <i>grid</i> de 960px .....	19
Figura 3 – Exemplo de uma imagem flexível visualizada em diferentes resoluções de tela ...	19
Figura 4 – Controlador de malha.....	22
Figura 5 – Sistema web que aplica o RWD.....	24
Figura 6 – Sistema web que utiliza <i>media queries</i> visualizado em iPhone.....	25
Figura 7 – Protótipo que executa processamento em máquinas virtuais.....	26
Quadro 1 – Requisitos funcionais.....	27
Quadro 2 – Requisitos não funcionais.....	27
Figura 8 – Diagrama de casos de uso .....	28
Figura 9 – Diagrama de classes .....	30
Figura 10 – Diagrama de sequência .....	31
Figura 11 – Diagrama de estados .....	31
Quadro 3 – Cabeçalho do documento do leiaute principal.....	34
Quadro 4 – Corpo do documento do leiaute principal.....	35
Quadro 5 – Cabeçalho superior com menu do leiaute principal .....	35
Quadro 6 – Cabeçalho inferior com ilustração e legenda do leiaute principal.....	35
Quadro 7 – Conteúdo do leiaute principal.....	36
Quadro 8 – Rotina para definir a navegação estruturada, ilustração e legenda.....	36
Quadro 9 – Rodapé do leiaute principal .....	37
Quadro 10 – <i>Tag viewport</i> .....	37
Quadro 11 – CSS do <i>grid</i> flexível .....	38
Quadro 12 – Uso do <i>grid</i> flexível.....	38
Figura 13 – Rodapé utilizando <i>grid</i> flexível.....	38
Quadro 13 – <i>Media queries</i> do sistema web .....	39
Quadro 14 – <i>Media query</i> para resoluções de até 1040px de largura .....	39
Quadro 15 – <i>Media query</i> para resoluções de até 960px de largura .....	39
Quadro 16 – <i>Media query</i> para resoluções de até 810px de largura .....	40
Quadro 17 – <i>Media query</i> para resoluções de até 710px de largura .....	40
Quadro 18 – <i>Media query</i> para resoluções de até 480px de largura .....	41

Figura 14 – Aplicação de <i>media queries</i> no menu .....	42
Quadro 19 – CSS das imagens flexíveis.....	42
Figura 15 – Imagem flexível adaptando-se à resolução .....	42
Quadro 20 – Rotina para salvar o certificado digital.....	43
Quadro 21 – Rotina para realizar a persistência do certificado digital.....	43
Quadro 22 – <i>Schema</i> do certificado digital .....	43
Quadro 23 – Rotina para salvar a empresa emitente .....	44
Quadro 24 – Rotina para realizar a persistência da empresa emitente .....	44
Quadro 25 – <i>Schema</i> da empresa emitente.....	44
Quadro 26 – Trecho de código do método enviar .....	45
Quadro 27 – Trecho de código do método para assinar NF-e.....	45
Quadro 28 – Incremento da numeração de lotes da empresa emitente .....	46
Quadro 29 – Invoca o método de recepção da NF-e emitida do <i>webservice</i> do SEFAZ .....	46
Quadro 30 – Invoca o método de recepção do status de autorização de uso da NF-e do <i>webservice</i> do SEFAZ .....	46
Quadro 31 – Trecho de código que grava o status de autorização de uso da NF-e.....	47
Quadro 32 – Rotina de recebimento de NF-e.....	47
Quadro 33 – Rotina de recebimento de NF-e.....	47
Quadro 34 – Rotina de listar NF-es.....	48
Quadro 35 – Rotina de carregar NF-es.....	48
Quadro 36 – Trecho de código de leitura dos dados do emitente no XML da NF-e.....	49
Quadro 37 – Trecho de código que obtém a URL do <i>webservice</i> do SEFAZ.....	50
Quadro 38 – Trecho de código que instancia o <i>webservice</i> de consulta de status de autorização de uso.....	50
Quadro 39 – Trecho de código que converte o retorno de consulta de status de autorização de uso.....	51
Figura 16 – Avaliação gratuita do WA.....	51
Figura 17 – Criação de ASP.NET Web Role.....	52
Figura 18 – Criação de pacote para disponibilização no WA .....	52
Figura 19 – Adição dos arquivos gerados pelo pacote .....	53
Figura 20 – Máquinas virtuais executando no portal do WA.....	53
Figura 21 – Tela de autenticação.....	54
Figura 22 – Tela inicial adaptada em diferentes resoluções.....	55
Figura 23 – Tela de certificado digital adaptada em diferentes resoluções.....	56

Figura 24 – Tela de empresa emitente adaptada em diferentes resoluções .....	57
Figura 25 – Tela de emitir e receber adaptada em diferentes resoluções .....	58
Figura 26 – Tela de visualizar adaptada em diferentes resoluções.....	59
Figura 27 – Tela de visualizar com NF-es filtradas.....	59
Figura 28 – Tela de visualizar os dados da NF-e adaptada em diferentes resoluções.....	60
Figura 29 – Tela de autorização de uso adaptada em diferentes resoluções .....	60
Figura 30 – Tela de autorização de uso com os dados da NF-e filtrada.....	61
Figura 31 – Tela de autorização de uso exibindo o retorno da consulta .....	61
Quadro 40 – Teste realizados de múltiplos navegadores .....	62
Quadro 41 – Compatibilidade das <i>media queries</i> nos navegadores .....	63
Quadro 42 – Testes realizados em múltiplas resoluções de tela de <i>desktops</i> .....	63
Quadro 43 – Testes realizados em múltiplas resoluções de tela de <i>tablets</i> .....	64
Quadro 44 – Testes realizados em múltiplas resoluções de tela de <i>mobiles</i> .....	64
Quadro 45 – Testes realizados em múltiplas resoluções de tela de televisões .....	64
Quadro 46 – Testes realizados em múltiplas resoluções de tela de <i>mobiles</i> .....	65
Quadro 47 – Teste de emissão de NF-e .....	65
Quadro 48 – Detalhamento do caso de uso UC01 .....	73
Quadro 49 – Detalhamento do caso de uso UC02 .....	73
Quadro 50 – Detalhamento do caso de uso UC03 .....	74
Quadro 51 – Detalhamento do caso de uso UC04 .....	74
Quadro 52 – Detalhamento do caso de uso UC05 .....	74
Quadro 53 – Detalhamento do caso de uso UC06 .....	75

## LISTA DE SIGLAS

ASP – *Active Server Page*

CPU – *Central Processing Unit*

CSS – *Cascading Style Sheets*

DOM – *Document Object Model*

HTML – *HyperText Markup Language*

ICMS – Imposto sobre Circulação de Mercadorias e Serviços

IDE – *Integrated Development Environment*

IIS – *Internet Information Service*

LINQ – *Language Integrated Query*

NF-e – Nota Fiscal eletrônica

RFB – Receita Federal do Brasil

RWD – *Responsive Web Design*

SEFAZ – SEcretaria da FAZenda

SDK – *Software Development Kit*

SQL – *Structured Query Language*

TCC – Trabalho de Conclusão de Curso

UML – *Unified Modeling Language*

URL – *Uniform Resource Locator*

VB.NET – Visual Basic .NET

WA – Windows Azure

WAE – *Web Application Extension*

WCF – *Windows Communication Foundation*

WSDL – *Web Services Description Language*

*XML – eXtensible Markup Language*

## **LISTA DE SÍMBOLOS**

% – por cento

" – polegadas

px – pixel

# SUMÁRIO

<b>1 INTRODUÇÃO.....</b>	<b>15</b>
1.1 OBJETIVOS DO TRABALHO .....	16
1.2 ESTRUTURA DO TRABALHO .....	16
<b>2 FUNDAMENTAÇÃO TEÓRICA .....</b>	<b>17</b>
2.1 <i>RESPONSIVE WEB DESIGN</i> .....	17
2.1.1 <i>Grids</i> flexíveis.....	18
2.1.2 Imagens flexíveis .....	19
2.1.3 <i>Media queries</i> .....	20
2.2 COMPUTAÇÃO EM NUVEM .....	20
2.3 WINDOWS AZURE.....	21
2.4 NOTA FISCAL ELETRÔNICA .....	22
2.5 TRABALHOS CORRELATOS .....	23
2.5.1 <i>Responsive web design</i> .....	23
2.5.2 <i>Adaptive layouts with media queries</i> .....	24
2.5.3 Integração de aplicações e serviços utilizando computação na nuvem com a plataforma Microsoft Windows Azure.....	25
<b>3 DESENVOLVIMENTO DO SISTEMA WEB.....</b>	<b>27</b>
3.1 REQUISITOS DO SISTEMA WEB .....	27
3.2 ESPECIFICAÇÃO .....	28
3.2.1 Diagrama de casos de uso .....	28
3.2.2 Diagrama de classe.....	29
3.2.3 Diagrama de sequência .....	30
3.2.4 Diagrama de estados .....	31
3.3 VISÃO GERAL DO SISTEMA WEB.....	32
3.4 IMPLEMENTAÇÃO .....	33
3.4.1 Técnicas e ferramentas utilizadas.....	34
3.4.2 Codificação .....	34
3.4.2.1 Leiaute principal .....	34
3.4.2.2 <i>Tag meta viewport</i> .....	37
3.4.2.3 Aplicação de <i>grid</i> flexível .....	37
3.4.2.4 Aplicação de <i>media queries</i> .....	39

3.4.2.5	Aplicação de imagens flexíveis .....	42
3.4.2.6	Tela de certificado digital .....	43
3.4.2.7	Tela de empresa emitente .....	43
3.4.2.8	Tela de emitir e receber .....	44
3.4.2.9	Tela de visualizar NF-e.....	47
3.4.2.10	Tela de autorização de uso.....	49
3.4.2.11	Processo de disponibilização no Windows Azure .....	51
3.4.3	Operacionalidade da implementação .....	54
3.4.3.1	Tela de autenticação .....	54
3.4.3.2	Tela inicial .....	54
3.4.3.3	Tela de certificado digital .....	56
3.4.3.4	Tela de empresa emitente .....	56
3.4.3.5	Tela de emitir e receber .....	57
3.4.3.6	Tela de visualizar .....	58
3.4.3.7	Tela de autorização de uso.....	60
3.5	RESULTADOS E DISCUSSÃO .....	61
3.5.1	Testes realizados em múltiplos navegadores .....	62
3.5.2	Testes realizados em múltiplas resoluções de tela .....	63
3.5.3	Testes realizados de emissões de NF-e .....	65
3.5.4	Vantagens e desvantagens do RWD .....	65
3.5.5	Comparativo dos trabalhos correlatos .....	68
<b>4</b>	<b>CONCLUSÕES.....</b>	<b>69</b>
4.1	EXTENSÕES .....	70
	<b>REFERÊNCIAS BIBLIOGRÁFICAS .....</b>	<b>71</b>
	<b>APÊNDICE A – Descrição dos casos de uso .....</b>	<b>73</b>

## 1 INTRODUÇÃO

A Internet, desde a sua criação há aproximadamente 40 anos, propagou-se e atingiu proporções extraordinárias, interligando as pessoas do mundo inteiro e tornando cada vez mais evidente a ideia de que a cooperação e a liberdade de informação podem ser mais propícias à inovação do que a competição e os direitos de propriedade (CASTELLS, 2003, p. 13). Em outras palavras, tornou-se a rede da vida da maioria das pessoas. Como consequência deste acelerado crescimento aliado à facilidade de acesso e uso da Internet, surgiu uma gama enorme e diversificada de dispositivos (*laptops, netbooks, tablets, celulares, entre outros*) capazes de conectarem-se em rede. Entretanto, as informações disponíveis nas páginas web em sua maioria não estão preparadas para apresentar o seu conteúdo de forma adequada e acessível a todos estes dispositivos. Foi pensando nisso que Marcotte (2010) introduziu o conceito do *Responsive Web Design* (RWD).

A abordagem principal faz referência ao processo de concepção e desenvolvimento de sistemas web que são capazes de reagir às ações do usuário e detectar o meio pelo qual está sendo visualizado, a fim de proporcionar a melhor experiência possível em termos de navegabilidade e legibilidade (WEBDESIGNSHOCK, 2011). De maneira geral, a estrutura do sistema web deverá adaptar-se automaticamente. A teoria por trás disto envolve a utilização de *grids*, *layouts* e imagens flexíveis, além de otimização de imagens, *media queries*, entre outros recursos. Portanto, não importa quantos dispositivos surgirão no futuro, o RWD sempre será capaz de fornecer uma resposta adequada.

Além desta questão de acessibilidade, ao longo dos anos, os sistemas web foram tornando-se complexos e, por consequência, houve um aumento da demanda por recursos de armazenamento e processamento. Tendo em vista esta situação e outros fatores, surgiu a computação em nuvem com o intuito de prover, segundo Taurion (2009, p. 2), "um conjunto de recursos com capacidade de processamento, armazenamento, conectividade, plataformas, aplicações e serviços disponibilizados na Internet". Com isso cria-se uma escalabilidade sem limites, capaz de atender os mais variados tipos de sistemas, desde os pequenos até os grandes. Tanto as aplicações locais como as aplicações em nuvem são construídas em algum tipo de plataforma que inclui um sistema operacional, uma forma de armazenamento de dados e outros componentes. O objetivo do Windows Azure (WA) é fornecer esta plataforma em nuvem (VERAS, 2011, p. 36).

Diante deste cenário, o presente trabalho apresenta o desenvolvimento de um sistema



web aplicando os conceitos abordados pelo RWD, potencializando-os em termos de escalabilidade e disponibilidade com a computação em nuvem da plataforma WA e como estudo de caso foi desenvolvido um sistema web para emissão de Nota Fiscal eletrônica (NF-e).

## 1.1 OBJETIVOS DO TRABALHO

O objetivo geral deste trabalho é utilizar os conceitos do RWD no desenvolvimento de sistemas web acessíveis e adequados aos diversos meios de acesso.

Os objetivos específicos do trabalho são:

- a) identificar vantagens e desvantagens na utilização do RWD;
- b) adaptar o sistema web a ser desenvolvido para os navegadores de diferentes dispositivos e resoluções de tela no qual serão visualizados;
- c) desenvolver um sistema web para emissão de NF-e, usando o conceito do RWD;
- d) disponibilizar o sistema web na plataforma em nuvem WA.

## 1.2 ESTRUTURA DO TRABALHO

No primeiro capítulo apresenta-se a introdução, expondo o cenário em que o sistema web desenvolvido insere-se, uma síntese dos temas abordados e os objetivos do trabalho.

No segundo capítulo expõe-se a fundamentação teórica sobre o tema estudado, elucidando os conhecimentos necessários para compreensão do trabalho. A fundamentação consiste em RWD, computação em nuvem, WA, NF-e e trabalhos correlatos.

No terceiro capítulo apresentam-se os detalhes inerentes ao desenvolvimento e programação do sistema web, detalhando as especificações e implementações aplicadas durante o processo de elaboração do trabalho. Também são expostos os detalhes da operacionalidade, bem como os resultados alcançados com o trabalho.

No quarto e último capítulo expõem-se as conclusões e possíveis extensões deste trabalho.

## 2 FUNDAMENTAÇÃO TEÓRICA

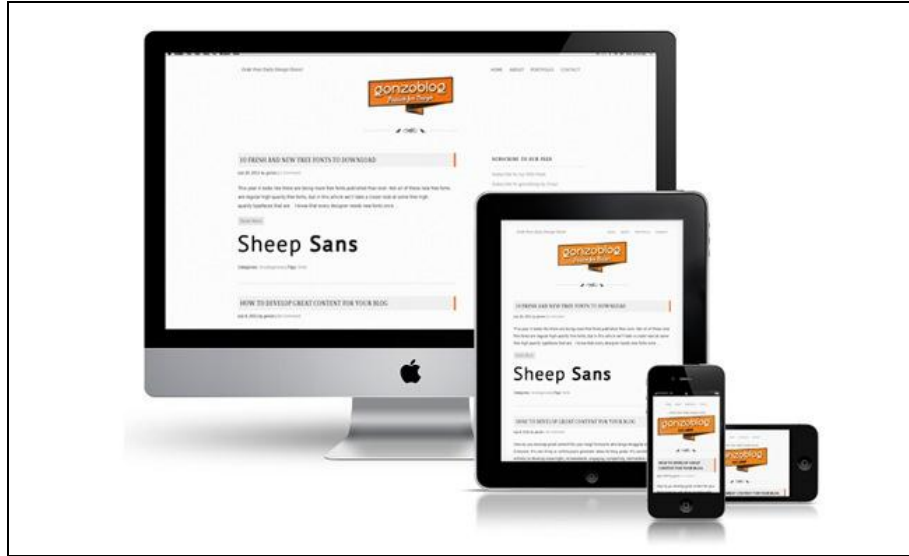
Neste capítulo é apresentado o conceito do RWD e suas principais técnicas para desenvolvimento de sistemas web. Em seguida, é abordada a computação em nuvem e a plataforma WA. Na seção subsequente é apresentada a NF-e. Por último, são descritos os trabalhos correlatos.

### 2.1 *RESPONSIVE WEB DESIGN*

A área de desenvolvimento web está rapidamente chegando ao ponto de ser incapaz de acompanhar os dispositivos que surgem no mercado. Torna-se impossível ou impraticável criar uma nova versão de sistema web para cada um destes dispositivos (KNIGHT et al., 2011, p. 110). Além disto, assim que os projetos são publicados, imediatamente estão à mercê das pessoas que os visualizam com suas diferentes configurações de fonte, cores de exibição, forma e tamanho da janela do navegador (MARCOTTE, 2011, p. 3-4). Para contornar este cenário e acompanhar o ritmo da tecnologia, surgiu o RWD criado pelo *web designer* e desenvolvedor Ethan Marcotte.

A abordagem sugere que o *design* e o desenvolvimento devem responder ao comportamento do usuário e do ambiente, baseado no tamanho da tela, plataforma e orientação. A prática consiste principalmente em uma mistura de *grids*, leiautes e imagens flexíveis e no uso inteligente de *media queries*, com o objetivo de tornar o sistema web tecnologicamente preparado para responder automaticamente às preferências do usuário. Isso elimina a necessidade de desenvolvimento específico para cada novo dispositivo no mercado (KNIGHT et al., 2011, p. 110).

A Figura 1 apresenta um exemplo de um sistema web que aplica o conceito do RWD.



Fonte: Webdesignshock (2011).

Figura 1 – Exemplo de um sistema web que aplica o conceito do RWD

A seguir são apresentados os principais itens que compõem o conceito do RWD.

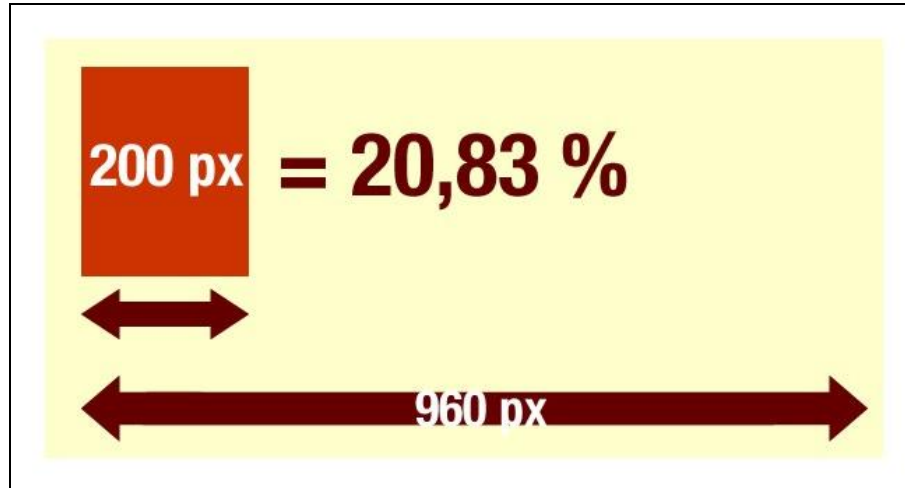
### 2.1.1 *Grids* flexíveis

Um *grid* flexível consiste no uso inteligente de camadas, porcentagens e matemática simples para criação de layouts (MARCOTTE, 2010). Permite que os elementos sejam redimensionados e reorganizados, pois faz uma revisão completa em termos de proporções (WEBDESIGNSHOCK, 2011).

Quando é criado um elemento na marcação, a largura torna-se fixa no *Cascading Style Sheets* (CSS), mas para tornar-se uma forma flexível é preciso de algo mais fluido e mais proporcional (MARCOTTE, 2011, p. 15).

A essência do RWD baseia-se em parar de pensar em pixels fixos e começar a considerar unidades percentuais. Para calcular as proporções de um elemento é preciso dividir a largura pelo tamanho total do *grid* e em seguida multiplicar o resultado por 100 e aplicar o valor percentual obtido sobre o elemento, a fim de realizar um redimensionamento correto (WEBDESIGNSHOCK, 2011).

A Figura 2 apresenta um exemplo de um elemento com unidade percentual calculada sobre um *grid* de 960px.



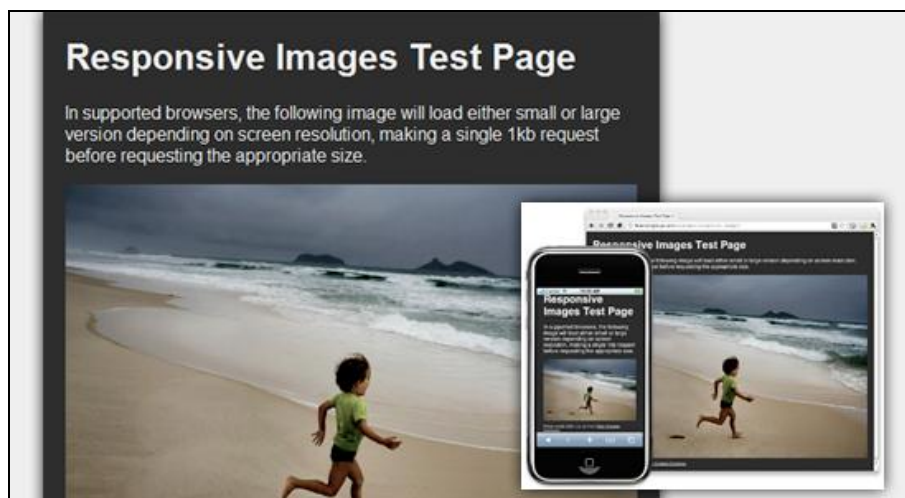
Fonte: Webdesignshock (2011).

Figura 2 – Exemplo de um elemento com unidade percentual calculada sobre um *grid* de 960px

### 2.1.2 Imagens flexíveis

Há inúmeras técnicas para redimensionar proporcionalmente imagens e muitas delas podem ser feitas facilmente. A opção mais popular define o tamanho das imagens pela propriedade *max-width* do CSS (KNIGHT et al., 2011, p. 117). Através dela a imagem assumirá qualquer tamanho, desde que seja menor do que o elemento que o contém. Entretanto, caso seja maior, a propriedade obriga que a largura da imagem coincida com a largura do elemento que a contém (MARCOTTE, 2011, p. 45).

A Figura 3 apresenta um exemplo de uma imagem flexível visualizada em diferentes resoluções de tela.



Fonte: Webdesignshock (2011).

Figura 3 – Exemplo de uma imagem flexível visualizada em diferentes resoluções de tela

Embora as imagens sejam simplesmente redimensionadas através de atributos *width* e

*height*, os quais ajudam a manter a informação organizada, o tempo para renderizar não é otimizado. Sendo assim, é aconselhável criar várias versões de uma imagem e dependendo do tamanho da tela atual deverá ser chamada a imagem mais apta. Esta técnica pode ser feita usando um recurso de detecção em conjunto com a manipulação do *Document Object Model* (DOM) (WEBDESIGNSHOCK, 2011).

### 2.1.3 *Media queries*

Marcotte (2011, p. 74) descreve as *media queries* como um mecanismo extremamente robusto para identificar não só os tipos de mídia, mas para realmente inspecionar as características físicas dos dispositivos e navegadores que renderizam o conteúdo. Através delas, os *designers* podem criar vários layouts usando documentos únicos de *HyperText Markup Language* (HTML) e seletivamente fornecer CSS com base em características diferentes, tais como o tamanho do navegador, resolução, orientação ou até mesmo a cor (WEBDESIGNSHOCK, 2011).

São comumente utilizadas pelos desenvolvedores web para estabelecer as propriedades *min-width* e *max-width*. A primeira define a janela de um navegador ou largura de tela mínima para que um determinado conjunto de estilos seja aplicável, enquanto a segunda faz exatamente o contrário. Se alguma coisa está abaixo deste limite, os estilos serão ignorados (KNIGHT et al., 2011, p. 126).

Embora as *media queries* forneçam uma grande solução para o desenvolvimento de sistemas web, ainda há muito a trabalhar, a fim de oferecer uma solução completa para dispositivos móveis (WEBDESIGNSHOCK, 2011).

## 2.2 COMPUTAÇÃO EM NUVEM

A computação em nuvem é uma tendência recente de tecnologia que visa aprimorar o uso da capacidade computacional do mundo inteiro, seguindo o princípio da computação em grade. Segundo Veras (2011, p. 27), computação em nuvem "é um conjunto de recursos virtuais facilmente utilizáveis e acessíveis tais como hardware, software, plataformas de desenvolvimento e serviços". Algumas características importantes da computação em nuvem

podem ser resumidas da seguinte maneira:

**A computação em nuvem cria uma ilusão** da disponibilidade de recursos infinitos, acessáveis sob demanda.

**A computação em nuvem elimina a necessidade** de adquirir e provisionar recursos antecipadamente.

**A computação em nuvem oferece elasticidade**, permitindo que as empresas usem os recursos na quantidade que forem necessários, aumentando e diminuindo a capacidade computacional de forma dinâmica.

**O pagamento dos serviços em nuvem** é pela quantidade de recursos utilizados (*pay-per-use*). (TAURION, 2009, p. 2-3, grifo do autor).

De maneira geral, esta é uma forma poderosa para fornecer recursos de computação, podendo ser acessado de todo o mundo, a qualquer hora e com qualquer computador ou dispositivo com acesso à Internet. As ferramentas para criar e manter sistemas em nuvem e seus serviços estão tornando-se cada vez mais flexíveis e fáceis de usar, fornecendo ao usuário além do acesso sob demanda, o imenso poder de computação e armazenamento (ANTONOPOULOS; GILLAM, 2010, p. 17).

### 2.3 WINDOWS AZURE

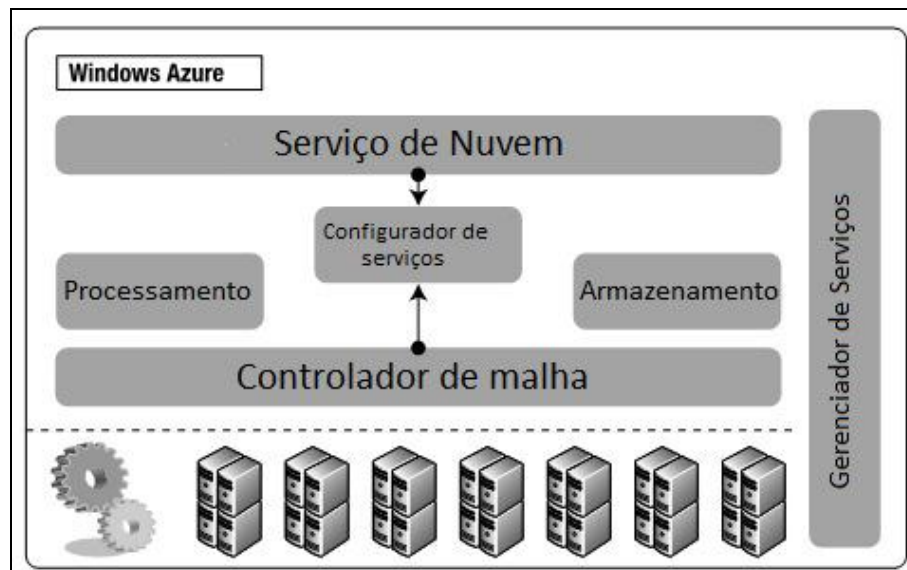
WA é uma plataforma em nuvem que permite criar, implantar e gerenciar aplicativos, usando qualquer linguagem, ferramenta ou estrutura (KOMMALAPATI, 2010).

O Windows AZURE é executado em máquinas presentes nos *DATACENTERS* da Microsoft. Em vez de fornecer o software para que os clientes instalem e executem as aplicações em seus próprios computadores, o Windows AZURE é um serviço: os clientes o utilizam para executar aplicações e armazenar dados em máquinas que pertencem à Microsoft. Essas aplicações podem fornecer serviços para empresas, consumidores ou ambos. (VERAS, 2011, p. 37).

Foi projetado para oferecer elasticidade, garantindo a disponibilidade e escalabilidade de recursos sob demanda, em ambientes totalmente virtualizados. Por ser um serviço, o custo é baseado no seu uso (KOMMALAPATI, 2010).

O serviço de processamento é dividido em duas categorias: *worker role application* e *web role application*. *Worker role application* são processos que rodam em segundo plano continuamente na nuvem. *Web role application* é uma aplicação web que roda em um ambiente *Internet Information Service* (IIS), normalmente uma aplicação *Active Server Page* (ASP) ou *Windows Communication Foundation* (WCF). Tanto uma *worker role application* quanto uma *web role application* podem ser instanciadas em diversas máquinas virtuais, permitindo assim um alto nível de escalabilidade para o serviço hospedado na nuvem (REDKAR, 2009, p. 110-111).

A abstração entre os serviços essenciais do WA e o hardware é gerenciada pelo controlador de malha, o qual monitora e gerencia servidores e recursos coordenados para software aplicativos. Além disto, este controlador de malha lê as informações de configuração fornecidas pelo serviço em nuvem para gerar as máquinas virtuais dos servidores necessários para implantar um serviço em nuvem. Enquanto isso, o processamento e o armazenamento são oferecidos como principais serviços do WA, assim como a computação. A Figura 4 exibe esta visão geral da arquitetura do WA.



Fonte: Redkar (2009, p. 106).

Figura 4 – Controlador de malha

Para executar uma aplicação é preciso acessar o portal do WA no navegador e escolher se quer criar uma conta de hospedagem para executar as aplicações, ou uma conta de armazenamento, ou ambas (VERAS, 2011, p. 38).

## 2.4 NOTA FISCAL ELETRÔNICA

O projeto de NF-e, segundo Primak (2009, p. 14), “tem como principal objetivo a implantação de um único modelo nacional para a documentação fiscal que venha a substituir a atual emissão de documento fiscal em papel”. A sua obrigatoriedade atinge a quase todos os contribuintes de Imposto sobre Circulação de Mercadorias e Serviços (ICMS).

Segundo a SEcretaria da FAZenda (SEFAZ), órgão responsável pela receita e despesa dos estados, a NF-e é “um documento de existência apenas digital, emitido e armazenado eletronicamente, com o intuito de documentar, para fins fiscais, uma operação de circulação

de mercadorias ou uma prestação de serviços, ocorrida entre as partes” (SECRETARIA DA FAZENDA, 2012). Deve ser assinado com certificado digital do remetente e enviado à SEFAZ de sua unidade federativa, para validação e autorização (DUARTE, 2010, p. 74).

A emissão e a assinatura digital devem ser feitas nota a nota, no entanto, o processo de transmissão da NF-e pode ser realizado em lotes, contendo uma única NF-e ou até 50 NF-es (SECRETARIA DA FAZENDA, 2012).

A SEFAZ e a Receita Federal do Brasil (RFB) disponibilizam a consulta da NF-e para o destinatário e outros legítimos interessados que tenham a chave de acesso (PRIMAK, 2009, p. 78). Além disto, o emitente e o destinatário deverão manter as NF-e em arquivo digital pelo prazo estabelecido na legislação tributária para a guarda dos documentos fiscais, devendo ser apresentadas à administração tributária quando solicitado (SECRETARIA DA FAZENDA, 2012).

## 2.5 TRABALHOS CORRELATOS

A seguir serão apresentados o trabalho de Marcotte (2010), que introduziu o conceito do RWD, o trabalho de Gustafson (2012), que mostra o uso das *media queries*, e, por último, o trabalho de Linhares (2011), que explorou a computação em nuvem oferecida pela plataforma WA.

### 2.5.1 *Responsive web design*

Marcotte (2010) descreve que o desenvolvimento web é definido pela sua transitoriedade, visto que as mudanças ocorrem dentro de um ou dois anos. Há uma grande preocupação em apresentar as informações de maneira acessível e adequada para diferentes larguras de janelas, resoluções de tela, preferências de usuários, fontes, navegadores, entre outros. Entretanto, esta preocupação está tornando-se cada vez mais acentuada, pois a cada dia que passa surgem mais e mais dispositivos com acesso à Internet e diferentes tipos de resoluções.

Tendo em vista este cenário, Marcotte (2010) criou o conceito RWD que aborda a questão da flexibilidade e adaptação da estrutura dos sistemas web ao meio pelo qual está



sendo visualizado. Para exemplificar o uso do RWD construiu um sistema web simples para uma revista hipotética, com leiaute de duas colunas sobre um *grid* e imagens flexíveis. Para conseguir o seu objetivo, utilizou as *media queries* do CSS3 para obter a resolução das telas. Com isso, pôde alterar o leiaute, aumentar a área de destinos dos *links*, embutir *touchscreen* (se o dispositivo permitir), mostrar ou esconder elementos, alterar posição de texto e otimizar apresentação.

A Figura 5 apresenta o sistema web desenvolvido por Marcotte (2010) que aplica o RWD.



Fonte: Knight (2011, p. 114).

Figura 5 – Sistema web que aplica o RWD

### 2.5.2 Adaptive layouts with media queries

Gustafson (2012) apresenta as *media queries* e demonstra o seu uso em uma página web já existente.

As *media queries* são muito semelhantes aos filtros CSS, que exploram o modo de como um navegador analisa as instruções CSS e são usados para filtrar o conteúdo, a apresentação ou o código. No entanto elas são muito mais robustas. Dentre as principais informações disponibilizadas pelas *media queries* destacam-se (GUSTAFSON, 2012):

- a) dimensões do navegador (largura, altura e razão de aspecto);
- b) dimensões do dispositivo (largura do dispositivo, altura do dispositivo e razão de aspecto do dispositivo);
- c) orientação do navegador;

- d) informação de cor (cor, índice de cor e monocromático);
- e) detalhes específicos de dispositivos tais como a sua resolução, se a tela é de *grid* ou baseado em bitmap, e o tipo de verificação (progressivo ou entrelaçado).

De maneira geral, no exemplo do autor, é demonstrado como tornar o leiaute adaptativo para duas dimensões do navegador e também para iPhone e iPad, com o uso das *media queries*.

A Figura 6 apresenta o sistema web desenvolvido por Gustafson (2012) que utiliza *media queries* visualizado em iPhone.



Fonte: Gustafson (2012).

Figura 6 – Sistema web que utiliza *media queries* visualizado em iPhone

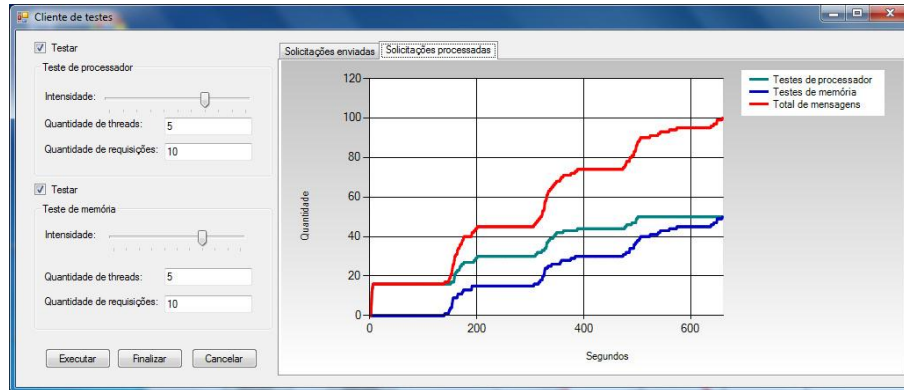
### 2.5.3 Integração de aplicações e serviços utilizando computação na nuvem com a plataforma Microsoft Windows Azure

A constante evolução das tecnologias e o aumento de complexidade dos sistemas, com conseqüente incremento nas necessidades de recursos de armazenamento e processamento, contribuem para a crescente busca de novos meios para disponibilizar serviços e aplicações (LINHARES 2011, p. 14). Sendo assim, o tema abordado em questão foi a computação em nuvem, que visa suprir a necessidade de escalabilidade e disponibilidade, utilizando a plataforma WA que oferece computação em nuvem em forma de serviço pago pelo uso.

O principal objetivo foi desenvolver um protótipo que executasse pesados algoritmos de matemática e grafos para efetuar o consumo de recursos em nuvem, como memória e processamento. Com isso, apresenta os resultados obtidos, identificando vantagens e

desvantagens.

A Figura 7 apresenta o protótipo desenvolvido por Linhares (2011) para o processamento de requisições em máquinas virtuais.



Fonte: Linhares (2011, p. 63).

Figura 7 – Protótipo que executa processamento em máquinas virtuais

Em suma, pelos resultados obtidos, identificou-se facilidade para escalar a estrutura, a fim de adaptar-se às necessidades momentâneas. Entretanto, segundo Linhares (2011, p. 65), “escalando o protótipo periodicamente, foi obtido um desempenho superior diante da utilização dele de maneira estática, sem a escalabilidade de recursos”.

### 3 DESENVOLVIMENTO DO SISTEMA WEB

Neste capítulo é apresentado o desenvolvimento do sistema web, sendo dividido em quatro etapas. Na seção 3.1 é apresentada a especificação dos requisitos funcionais e não funcionais do sistema web. Na seção 3.2 é apresentada a especificação do sistema web, tais como diagramas de casos de uso, de classes, de sequência e de estados. Na seção 3.3 é apresentada a visão geral do sistema web. Por fim na seção 3.4 é apresentada a implementação do sistema web.

#### 3.1 REQUISITOS DO SISTEMA WEB

Com o objetivo de aplicar os conceitos do RWD no estudo de caso de NF-e, o sistema web atenderá os requisitos relacionados no Quadro 1 e Quadro 2. Além disto, cada RF está vinculado a um caso de uso, facilitando a identificação do propósito e justificando a sua existência.

REQUISITOS FUNCIONAIS	CASOS DE USO
RF01: O sistema web deve permitir a emissão da NF-e.	UC01
RF02: O sistema web deve permitir o cadastro de empresas emitentes de NF-e.	UC02
RF03: O sistema web deve permitir o cadastro de certificado digital.	UC03
RF04: O sistema web deve permitir o recebimento da NF-e via <i>upload</i> .	UC04
RF05: O sistema web deve permitir a visualização da NF-e emitida e recebida.	UC05
RF06: O sistema web deve permitir consultar a autorização de uso da NF-e.	UC06

Quadro 1 – Requisitos funcionais

REQUISITOS NÃO FUNCIONAIS
RNF01: Implementar o sistema web aplicando os conceitos do RWD.
RNF02: Implementar o sistema web utilizando a plataforma de computação em nuvem WA.
RNF03: Implementar o sistema web utilizando o ambiente de desenvolvimento Visual Studio 2012.
RNF04: Implementar o sistema web utilizando a linguagem de programação Visual Basic .NET (VB.NET.)
RNF05: Implementar o sistema web utilizando a plataforma de desenvolvimento web ASP.NET.

Quadro 2 – Requisitos não funcionais

## 3.2 ESPECIFICAÇÃO

Em todo o desenvolvimento foi utilizada a orientação a objetos e para a especificação da implementação foi utilizada a *Unified Modeling Language* (UML) através de diagramas de casos de uso, de classe, de sequência e de estados. Para a modelagem dos diagramas foi utilizada a ferramenta Enterprise Architect.

### 3.2.1 Diagrama de casos de uso

No diagrama de casos de uso apresentado na Figura 8 estão ligados os requisitos funcionais especificados para o sistema web, enquanto que o seu detalhamento encontra-se no Apêndice A.

Através deste diagrama são apresentadas as principais funcionalidades pertinentes ao usuário, como emissão e recepção de NF-e, cadastro de empresas emitentes e certificados digitais, visualização e consulta de autorização de uso da NF-e.

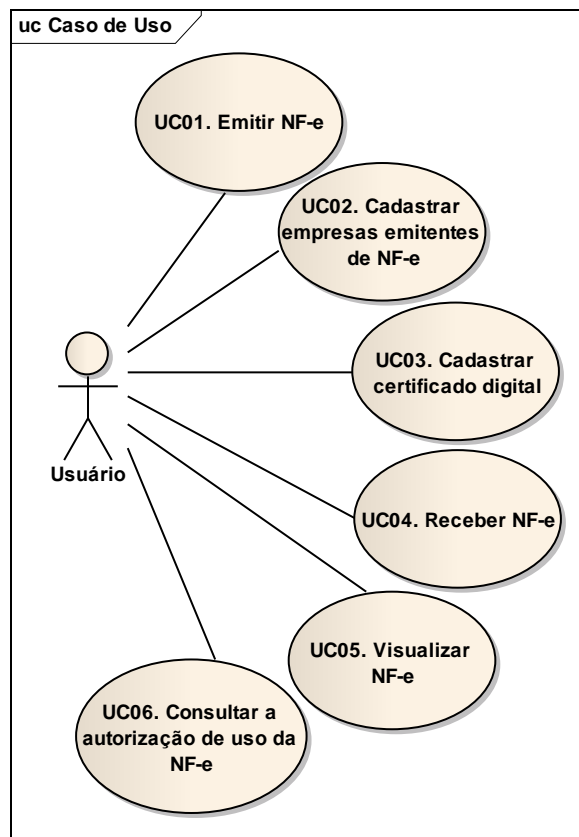


Figura 8 – Diagrama de casos de uso

### 3.2.2 Diagrama de classe

A Figura 9 apresenta o diagrama de classes do sistema web. A fim de facilitar o aspecto de legibilidade deste diagrama foram abstraídos alguns atributos, como os de relacionamento e conjuntos de dados que pertençam a um mesmo tipo. Além disto, este diagrama apenas mostra o modelo de dados utilizado para implementar as principais funcionalidades do sistema web.

Os dados da NF-e são armazenados em instâncias da classe `XNFe` e as demais classes apenas agrupam conjuntos de dados da NF-e que representam respectivamente:

- a) `XNFeDup`: as duplicatas;
- b) `XNFeProcRef`: a estrutura das NF-es de referência do *eXtensible Markup Language* (XML);
- c) `XNFeObsCont`: as informações de observações do contribuinte;
- d) `XNFeObsFisco`: as informações financeiras, tributárias, e económicas;
- e) `XNFeRef`: a estrutura das NF-e referenciadas;
- f) `XNFeTranspReboque`: a estrutura das informações de reboque do transporte;
- g) `XNFeTranspVol`: a estrutura dos volumes de transporte;
- h) `XNFeTranspVolLacres`: a estrutura dos lacres dos volumes;
- i) `XRetConsSitNFe`: a estrutura do retorno do *webservice* de consulta da SEFAZ;
- j) `XNFeDet`: a estrutura para os itens (detalhes) da NF-e;
- k) `XNFeDetMed`: a estrutura para dados específicos do comércio de medicamentos do item da NF-e;
- l) `XNFeDetImposto`: a estrutura dos impostos dos itens da NF-e;
- m) `XNFeDetArma`: a estrutura para dados específicos do comércio de armas do item da NF-e;
- n) `XNFeDetDI`: a estrutura para dados de importação do item da NF-e;
- o) `XNFeDetDIadi`: a estrutura para dados de adição dos documentos de importação do item da NF-e;
- p) `XNFeDetVeicProd`: a estrutura para os dados do veículo do item da NF-e;
- q) `XNFeDetComb`: a estrutura para dados de cobrança do item da NF-e.

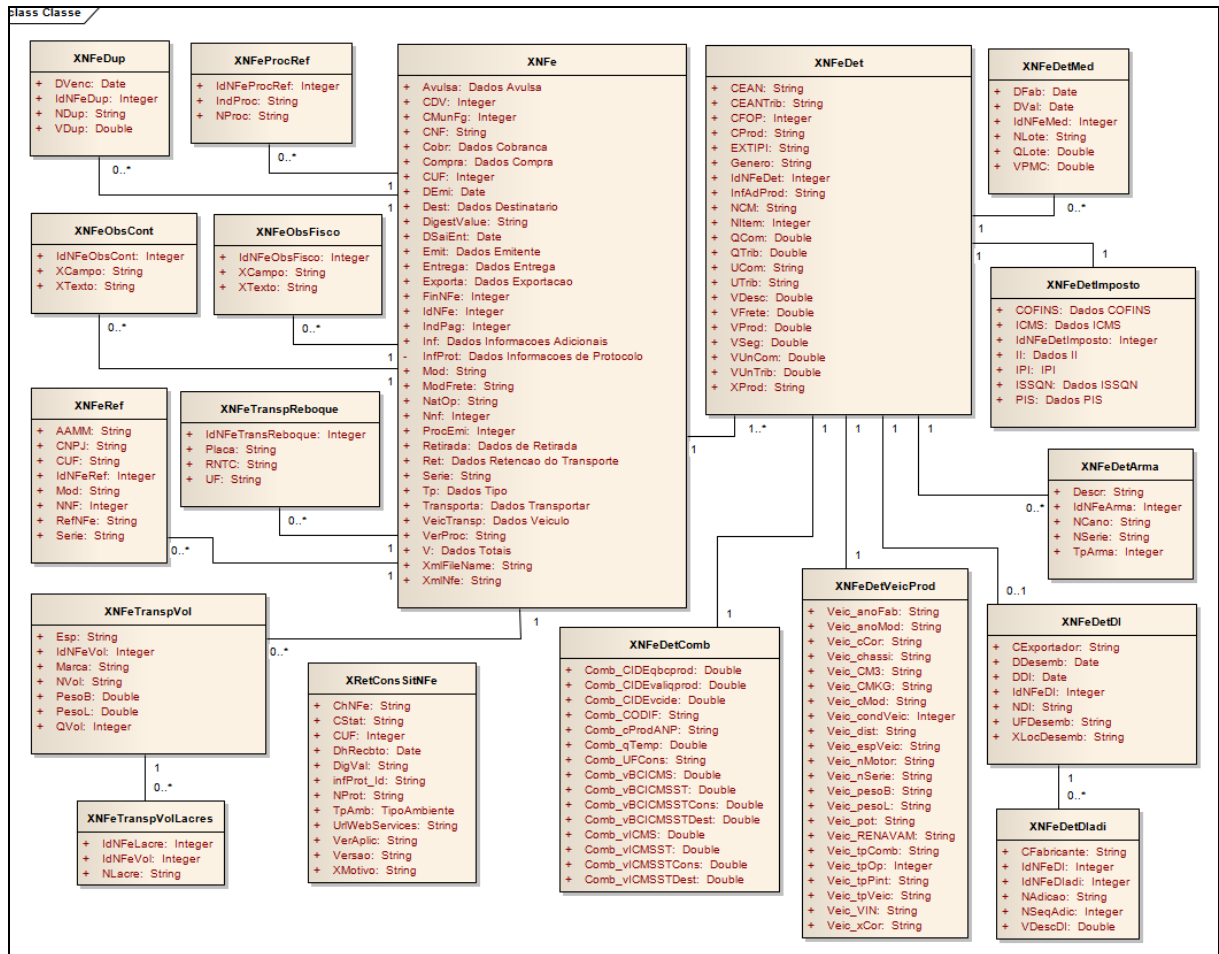


Figura 9 – Diagrama de classes

### 3.2.3 Diagrama de sequência

A Figura 10 apresenta o diagrama de sequência do UC01 - Emitir NF-e. Inicialmente o usuário envia a NF-e para o sistema web, posteriormente será assinada digitalmente e incluída em um único lote de NF-e para envio ao SEFAZ. Em seguida é obtido o retorno do lote de NF-e enviado, por conseguinte será consultado o status de autorização de uso da NF-e, por fim atualizado no arquivo XML original.

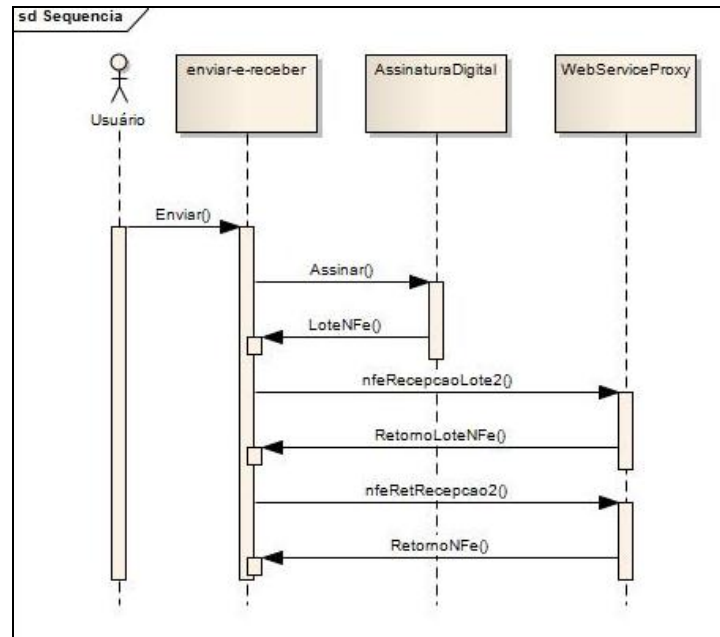


Figura 10 – Diagrama de sequência

### 3.2.4 Diagrama de estados

A Figura 11 apresenta o diagrama de estados da NF-e que foram implementados pelo sistema web.

A NF-e ao ser emitida assume o estado Em Processamento na SEFAZ, depois de processada a NF-e assume o estado de acordo com o retorno do SEFAZ que pode ser Rejeitada, Autorizada para uso e Denegada para uso.

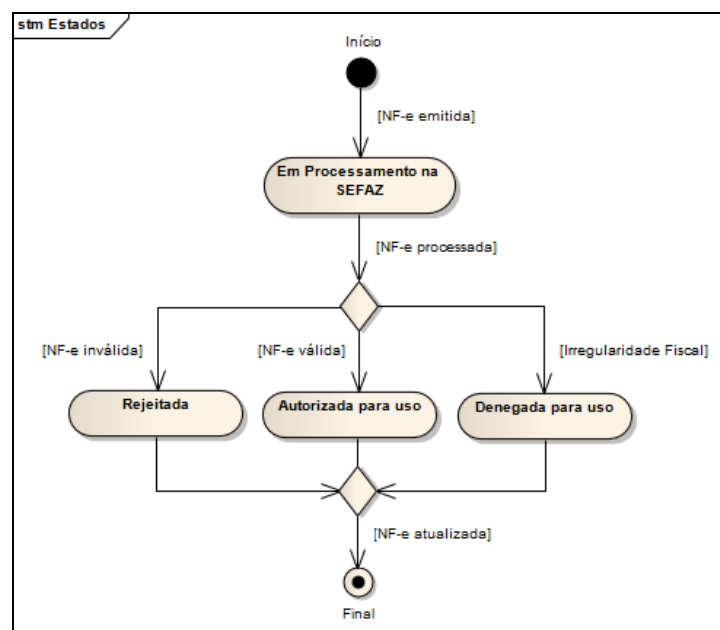


Figura 11 – Diagrama de estados



### 3.3 VISÃO GERAL DO SISTEMA WEB

O sistema web desenvolvido tem por finalidade demonstrar os conceitos do RWD através de um sistema de emissão de NF-e.

A arquitetura web é composta por onze páginas ASP.NET listadas abaixo:

- a) `login`: solicita a autenticação do usuário para acessar a área restrita do sistema;
- b) `inicial`: tela principal a qual permite o acesso às demais telas do sistema;
- c) `proposta`: apresenta informações e detalhes sobre a proposta deste Trabalho de Conclusão de Curso (TCC);
- d) `responsive-web-design`: apresenta informações e detalhes sobre o conceito do RWD;
- e) `windows-azure`: apresenta informações e detalhes sobre a plataforma na nuvem WA;
- f) `nota-fiscal-eletronica`: apresenta informações e detalhes sobre a NF-e;
- g) `certificado-digital`: permite o cadastro de certificados digitais no sistema e implementa o RF03;
- h) `empresa-emitente`: permite o cadastro de empresas emitentes no sistema e implementa o RF02;
- i) `emitir-e-receber`: permite a emissão e recepção de NF-e no sistema e implementa o RF01 e RF04;
- j) `visualizar`: permite a visualização de NF-es emitidas e recebidas no sistema e implementa o RF05;
- k) `autorizacao-de-uso`: permite a consulta da autorização de uso de NF-es emitidas e recebidas no sistema e implementa o RF06.

A visão geral do sistema web é representada pelo diagrama *Web Application Extension* (WAE), exposto na Figura 12, o qual é uma extensão da notação UML com semânticas adicionais e restrições, a fim de permitir a modelagem arquitetural de elementos web. Além disto, com esta extensão é possível realizar a captura, desenvolvimento e a análise de como será a execução das regras de negócio nas páginas web do sistema.

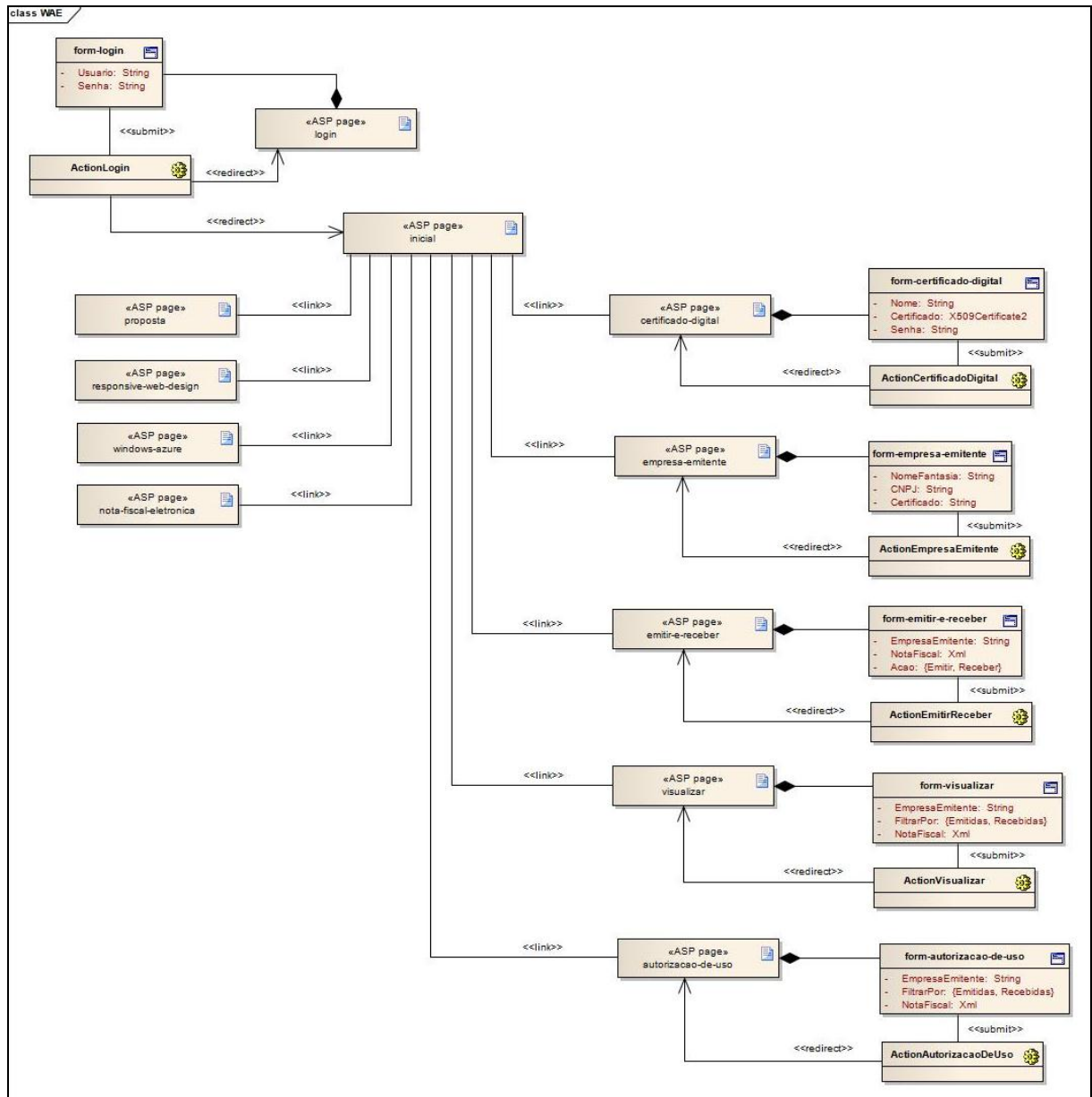


Figura 12 – Diagrama de WAE

### 3.4 IMPLEMENTAÇÃO

Nesta seção serão apresentadas informações sobre técnicas e ferramentas utilizadas para a implementação, bem como a própria implementação e cada etapa de desenvolvimento. Ao final, será descrita a operacionalidade do sistema web desenvolvido.

### 3.4.1 Técnicas e ferramentas utilizadas

A linguagem de programação utilizada para o desenvolvimento do sistema web foi o VB.NET juntamente com o *framework* .NET 4.0 e a plataforma de desenvolvimento web ASP.NET. O ambiente de desenvolvimento utilizado foi o Visual Studio 2012.

### 3.4.2 Codificação

As seções a seguir irão detalhar a codificação do sistema web para emissão de NF-e.

#### 3.4.2.1 Leiaute principal

O leiaute principal faz uso de um recurso do ASP.NET chamado *master pages* que permite criar uma página padrão que poderá ser utilizada por todo o sistema web. Com isto, todas as demais páginas podem herdar da *master page* a aparência visual. Portanto, qualquer mudança feita no leiaute principal altera imediatamente as demais páginas do sistema web.

O cabeçalho do documento da página, exposto no Quadro 3, contém o título e outras informações, dentre elas os *links* de acesso ao CSS, JavaScript e as bibliotecas do jQuery.

```
<head>
<meta charset="utf-8">
<title>TCC II - Usando Responsive Web Design para Desenvolvimento de Sistemas Web
</title>
<meta name="viewport" content="width=device-width; initial-scale=1.0; maximum-scale=1.0; user-scalable=0;" />
<meta name="description" content="TCCII - Usando Responsive Web Design para Desenvolvimento de Sistemas Web. Por Rodrigo Eduardo Boni.">
<link rel="shortcut icon" href=" ../Images/icone.ico" />
<link rel="apple-touch-icon-precomposed" href=" ../Images/icone.ico" />
<script type="text/javascript" src=" ../Scripts/jquery-1.7.2.min.js"></script>
<script type="text/javascript" src=" ../Scripts/scripts.min.js"></script>
<script type="text/javascript" src=" ../Scripts/jquery-1.3.2.min.js"></script>
<script type="text/javascript" src=" ../Scripts/jquery-ui-1.7.3.custom.min.js"></script>
<script type="text/javascript" src=" ../Scripts/ui.core.js"></script>
<script type="text/javascript" src=" ../Scripts/ui.tabs.js"></script>
<script type="text/javascript" src=" ../Scripts/jquery.js"></script>
<script type="text/javascript" src=" ../Scripts/jquery-ui.js"></script>
<script type="text/javascript" src=" ../Scripts/jquery.scrollabletab.js"></script>
<link type="text/css" href=" ../Styles/ui.tabs.css" rel="stylesheet" media="screen">
<link type="text/css" href=" ../Styles/ui.core.css" rel="stylesheet" media="screen">
<link type="text/css" href=" ../Styles/ui.theme.css" rel="stylesheet" media="screen">
<link type="text/css" href=" ../Styles/styles.css" rel="stylesheet" media="screen">
<!--[if lt IE 9]>
<script src="//html5shiv.googlecode.com/svn/trunk/html5.js"></script>
<![endif]-->
</head>
```

Quadro 3 – Cabeçalho do documento do leiaute principal

O corpo do documento da página, exposto no Quadro 4, é dividido em quatro partes, o cabeçalho superior com o menu, o cabeçalho inferior com ilustração e legenda, o conteúdo e o rodapé.

```

<body class="page page-id-10 page-parent page-template-default">
  <!-- Menu -->
  <div id="navMain">...</div>
  <!-- Ilustração & Legenda -->
  <header id="siteHeader" class="imageContainer">...</header>
  <!-- Conteúdo -->
  <div id="content" class="clear">...</div>
  <!-- Rodapé -->
  <footer>...</footer>
</body>

```

Quadro 4 – Corpo do documento do leiaute principal

No cabeçalho superior localiza-se o menu disposto horizontalmente, com acesso às demais páginas do sistema web, visualizado no Quadro 5.

```

<!-- Menu -->
<div id="navMain">
  <nav>
    <h1>
      <a href="inicial.aspx" rel="inicial" title="Página Inicial">inicial </a>
    </h1>
    <ul id="menu-navmain" class="menu">
      <li id="menu-item-17" class="menu-item menu-item-type-post_type menu-item-object-page menu-item-17">
        <a href="empresa-emitente.aspx">Empresa Emitente</a></li>
      <li id="menu-item-587" class="menu-item menu-item-type-custom menu-item-object-custom menu-item-587">
        <a href="certificado-digital.aspx">Certificado Digital</a></li>
      <li id="menu-item-1520" class="menu-item menu-item-type-post_type menu-item-object-page menu-item-1520">
        <a href="emitir-e-receber.aspx">Emitir & Receber</a></li>
      <li id="menu-item-19" class="menu-item menu-item-type-post_type menu-item-object-page menu-item-19">
        <a href="visualizar.aspx">Visualizar</a></li>
      <li id="menu-item-20" class="external menu-item menu-item-type-custom menu-item-object-custom menu-item-20">
        <a href="autorizacao-de-uso.aspx">Autorização de Uso</a></li>
    </ul>
    <a href="#" id="showHideNav">Mostrar Navegação</a>
  </nav>
</div>

```

Quadro 5 – Cabeçalho superior com menu do leiaute principal

No cabeçalho inferior localiza-se a ilustração e a legenda, as quais desempenham papel informativo da página visitada, visualizada no Quadro 6.

```

<!-- Ilustração & Legenda -->
<header id="siteHeader" class="imageContainer">
  <div class="headerImage" id="divHeader" runat="server">
    <div class="container">
      <div class="headerDescription">
        <hgroup>
          <h1>
            <a id="lnkHeader" runat="server"></a>
          </h1>
          <h2 class="col4">
            <asp:Label runat="server" ID="lblHeader"></asp:Label>
          </h2>
        </hgroup>
      </div>
    </div>
  </div>
</header>

```

Quadro 6 – Cabeçalho inferior com ilustração e legenda do leiaute principal

O conteúdo é dividido em duas partes, expostas no Quadro 7, a primeira corresponde à navegação estruturada, ou seja, registra a hierarquia da localização da página, a segunda corresponde à parte que será preenchida pelo conteúdo das páginas que herdaram do leiaute

principal.

```

<!-- Conteúdo -->
<div id="content" class="clear">
  <div class="container clear">
    <nav>
      <div id="breadcrumb">
        <a href="inicial.aspx" class="homelink">Inicial</a> &raquo;
        <a id="lnkNavegation" runat="server"></a>
      </div>
    </nav>
    <div class="articleContainer">
      <article class="col9">
        <!-- Conteúdo específico da página que herdar -->
        <asp:ContentPlaceHolder ID="ContentPlaceHolder1" runat="server">
        </asp:ContentPlaceHolder>
      </article>
    </div>
  </div>
</div>

```

Quadro 7 – Conteúdo do leiaute principal

Tanto a navegação estruturada, quanto a ilustração e a legenda necessitam ser alteradas de acordo com a página visualizada, o Quadro 8 exibe o trecho de código responsável por esta tarefa.

```

Protected Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Load
  Try
    Select Case Me.Parent.GetType.Name.Replace("pages_", "").Replace("_aspx", "")
      Case "autorizacao_de_uso"
        Me.DefinirPagina("autorizacao-de-uso.aspx", "Autorização de Uso",
          "Permite consultar no SEFAZ o status de Autorização de Uso da NF-e", "headerImage headerAutorizacaoDeUso")
      Case "certificado_digital"
        Me.DefinirPagina("certificado-digital.aspx", "Certificado Digital",
          "Validade jurídica garantida pela assinatura digital, padrão ICP-Brasil (tipos A1 ou A3)", "headerImage headerCertificadoDigital")
      Case "emitir_e_receber"
        Me.DefinirPagina("emitir-e-receber.aspx", "Emitir & Receber",
          "Realizar a emissão ou o recebimento de Nota Fiscal Eletrônica", "headerImage headerEmitirEReceber")
      Case "empresa_emitente"
        Me.DefinirPagina("empresa-emitente.aspx", "Empresa Emitente",
          "Responsável pela emissão de uma Nota Fiscal Eletrônica", "headerImage headerEmpresaEmitente")
      Case "nota_fiscal_eletronica"
        Me.DefinirPagina("nota-fiscal-eletronica.aspx", "Nota Fiscal Eletrônica",
          "Documento digital, emitido e armazenado eletronicamente", "headerImage headerNotaFiscalEletronica")
      Case "proposta"
        Me.DefinirPagina("proposta.aspx", "Proposta", "Usando Responsive Web Design para Desenvolvimento de Sistemas Web", "headerImage headerProposta")
      Case "responsive_web_design"
        Me.DefinirPagina("responsive-web-design.aspx", "Responsive Web Design",
          "Adaptar a experiência de acordo com o dispositivo", "headerImage headerResponsiveWebDesign")
      Case "visualizar"
        Me.DefinirPagina("visualizar.aspx", "Visualizar", "Visualizar a Nota Fiscal Eletrônica emitida ou recebida", "headerImage headerVisualizar")
      Case "windows_azure"
        Me.DefinirPagina("windows-azure.aspx", "Windows Azure", "Plataforma na nuvem da Microsoft", "headerImage headerWindowsAzure")
    End Select
  Catch wrkErro As Exception
    UtilsRWD.ExibirAlerta(CType(sender, Control), String.Format("Erro: {0}", wrkErro.Message))
  End Try
End Sub

Private Sub DefinirPagina(ByVal pPagina As String, ByVal pTitulo As String, ByVal pTexto As String, ByVal pImagem As String)
  InkHeader.Title = pTitulo
  InkHeader.HRef = pPagina
  InkHeader.InnerText = pTexto
  divHeader.Attributes("class") = pImagem
  lblHeader.Text = pTexto
  InkNavegation.Title = pTitulo
  InkNavegation.HRef = pPagina
  InkNavegation.InnerText = pTitulo
End Sub

```

Quadro 8 – Rotina para definir a navegação estruturada, ilustração e legenda

Por último, o rodapé que contém algumas informações pessoais do autor da proposta, visualizado no Quadro 9.

```

<!-- Rodapé -->
<footer>
  <div class="container clear">
    <div class="col9 alignleft withSeperator">
      <div class="vcard">
        <div class="col2">
          <strong>Rodrigo Eduardo Boni.</strong><br />
          <span class="street-address">Rua Fides Deeke, 250. Apto 156.</span><br />
          <span class="postal-code">89035-200</span> <span class="locality">Blumenau/SC - Brasil</span>
        </div>
        <div class="col2">
          <span class="tab60">Telefone: </span><span class="tel">(47) 3237-7274</span><br />
          <span class="tab60">Celular: </span><span class="fax">(47) 9208-9228</span><br />
          <span class="tab60">E-Mail: </span><a class="email" href="mailto:drigoboni@hotmail.com.br">drigoboni@hotmail.com.br</a><br />
        </div>
      </div>
    </div>
    <div class="col7 alignleft">
      <nav>
        <ul id="navFooter">
          <li>Todos os direitos reservados &copy; 2012. Rodrigo Eduardo Boni. <a href="http://www.windowsazure.com/pt-br/"
            title="Hospedado por Windows Azure" target="_blank">Hospedado por Windows Azure</a>.</li>
        </ul>
      </nav>
    </div>
  </div>
</footer>

```

Quadro 9 – Rodapé do leiaute principal

### 3.4.2.2 Tag meta viewport

Inicialmente, no leiaute principal foi adicionada a *tag meta viewport*, responsável por customizar a resolução inicial que o navegador deve renderizar o tamanho da página no dispositivo, exposta no Quadro 10.

```
<meta name="viewport" content="width=device-width; initial-scale=1.0; maximum-scale=1.0; user-scalable=0;" />
```

Quadro 10 – Tag viewport

No atributo `width` é definido a largura do `viewport` para `device-width`, ou seja, determina automaticamente um valor igual à largura da tela do dispositivo. No atributo `initial-scale` é definido a escala inicial do `viewport` para 1.0, ou seja, trata-se do estado normal, sem aumento ou diminuição do *zoom*. No atributo `user-scalable` é definido a possibilidade de fazer *zoom* na tela para 0, ou seja, desabilitado o *zoom*.

### 3.4.2.3 Aplicação de *grid* flexível

O *grid* flexível escolhido tem tamanho de 960px, devido à resolução tornar-se adequada para a maioria nos *desktops*, além disto, o *grid* foi dividido proporcionalmente em nove colunas. Portanto, todo o conteúdo do sistema web acontece dentro desta divisão, impedindo que tudo espalhe-se desordenadamente pela tela. O Quadro 11 apresenta o CSS do

*grid* flexível.

```
.col1, .col2, .col3, .col4, .col5, .col6, .col7, .col8, .col9 { float:left; }
.col1 { width:88px; }
.col2 { width:197px; }
.col3 { width:306px; }
.col4 { width:415px; }
.col5 { width:524px; }
.col6 { width:633px; }
.col7 { width:742px; }
.col8 { width:851px; }
.col9 { width:960px; }
```

Quadro 11 – CSS do *grid* flexível

O Quadro 12 exibe o trecho de código que faz uso do *grid* flexível em todas as páginas do sistema web e que conseqüentemente delimita o espaço do conteúdo.

```
<div class="articleContainer">
  <article class="col9">
    <!-- Conteudo específico da página que herdar -->
    <asp:ContentPlaceHolder ID="ContentPlaceHolder1" runat="server">
    </asp:ContentPlaceHolder>
  </article>
</div>
```

Quadro 12 – Uso do *grid* flexível

Quando os limites de largura de alguma coluna do *grid* são atingidos, tanto a coluna, quanto o conteúdo contido quebrará para uma nova linha. A Figura 13 ilustra o que acontece com o conteúdo do rodapé quando estes limites são atingidos, neste caso o conteúdo está dividido em duas partes que estão contidas cada uma em duas colunas das nove estabelecidas pelo *grid*.

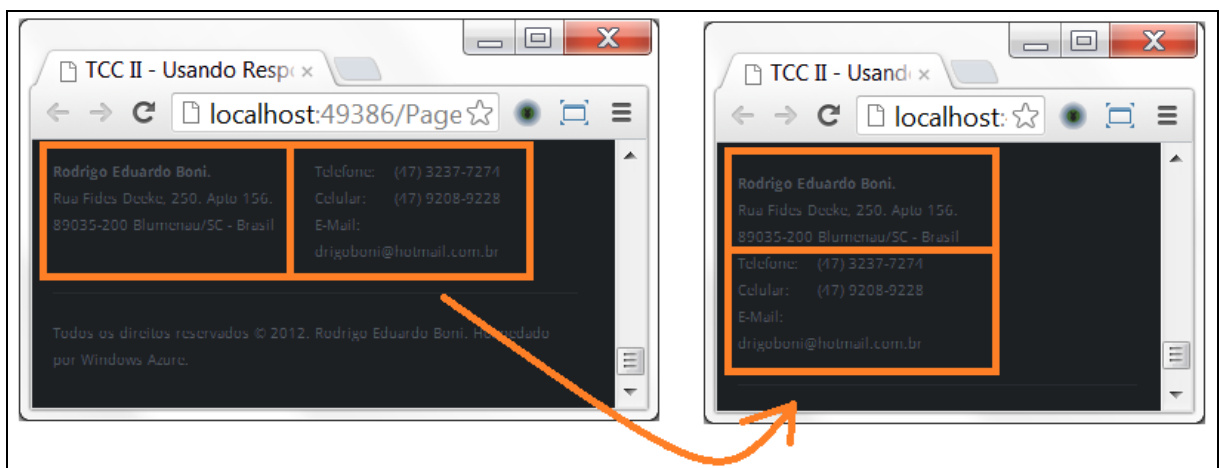


Figura 13 – Rodapé utilizando *grid* flexível

Todos os componentes utilizados assumem como largura máxima a coluna do *grid* pela qual estão contidas. Sendo assim, caso houver qualquer redimensionamento ou reposicionamento de alguma das colunas do *grid*, automaticamente o conteúdo acompanhará.

### 3.4.2.4 Aplicação de *media queries*

As *media queries* escolhidas foram para as resoluções com largura menor que 1040px, 960px, 810px, 710px e 480px, a fim de tornar o conteúdo do sistema web adaptável também para *netbooks*, *tablets*, *smartphones*, entre outros dispositivos de baixa resolução. O Quadro 13 apresenta as *media queries* do sistema web.

```

/* Breakpoint < 1040px -----
@media only screen and (max-width: 1040px) {...

/* Breakpoint < 960px -----
@media only screen and (max-width: 960px) {...

/* Breakpoint < 810px -----
@media only screen and (max-width: 810px) {...

/* Breakpoint < 710px -----
@media only screen and (max-width: 710px) {...

/* Breakpoint < 480px -----
@media only screen and (max-width: 480px) {...

```

Quadro 13 – *Media queries* do sistema web

Na *media query* para resoluções de até 1040px de largura apenas é redefinido no menu os limites da margem e preenchimento, como pode ser observado no Quadro 14.

```

@media only screen and (max-width: 1040px) {
#navMain h1 a { margin: 0 1px 0 -16px; }
#navMain nav ul li a { padding: 25px 15px 20px 15px; }
}

```

Quadro 14 – *Media query* para resoluções de até 1040px de largura

Na *media query* para resoluções de até 960px de largura, também é redefinido no menu os limites da margem, preenchimento e posicionamento. Além disto, a disponibilidade da largura das colunas do *grid* flexível é alterada, torna as imagens flexíveis e reajusta o posicionamento do conteúdo da tela inicial, como pode ser observado no Quadro 15.

```

@media only screen and (max-width: 960px) {
body { overflow-x:hidden; width:100%; }
.col9 { width:100%; }
.col7 { width:100%; }
.col5 { width:95%; }
#navMain h1 a { margin: 0 1px 0 0px; }
#navMain, #navMain nav { width:100%; position:absolute; }
.headerDescription { left:15px; }
.container { width:100%; float:left; }
#mainTeaser { float:left; background:white !important; width: 100%; }
#mainTeaser .teaser { width:27%; margin:5% 3%; padding:0; background-position:top center; }
#content { width: 100%; margin-left: 15px; }
#content .container { width:96%; }
#content .articleContainer { width:100%; }
#content .col6 { width:66%; }
#content .col3 { width:33%; }
#content .col6 .col6, #content .col6 .col3 { width:100%; }
#content img { width:100%; height:auto; max-width:inherit; }
footer { float:left; margin:0; width:100%;}
footer .col9 { width:95%; }
footer .container { margin:0 0 0 15px; }
}

```

Quadro 15 – *Media query* para resoluções de até 960px de largura



Na *media query* para resoluções de até 810px também é redefinido no menu os limites da margem, preenchimento e posicionamento. Além disto, todas as ilustrações das páginas devem carregar a versão de resolução 810px das respectivas imagens, como pode ser observado no Quadro 16.

```
@media only screen and (max-width: 810px) {
#navMain, #navMain nav, #menu-navmain { width: 100%; position: relative; float: left; overflow:visible;
height:auto; }
#siteHeader { float:left; margin-top:-60px; height:auto; }
#siteHeader .headerImage { background:url(img/header-imagens/default-810px.jpg) top center no-repeat }
.home #siteHeader { height:auto; }
.home .headerImage { width:100%; padding-bottom:30px; float:left; background:url(..Images/
inicial-810px.png) top center no-repeat !important; }
.home .headerDescriptionContainer { width:100%; text-align:center; float:left; margin-top: 0; }
.home .headerDescription { position: relative; left: 0; top: 0; margin-top:290px; width:100% !
important; }
.home .headerDescription .button { float: left; margin: 30px auto 0 auto; left: 50%; margin-left:
-90px; }
.headerImage { width:100%; padding-bottom:30px; float:left; }
#siteHeader .container { width:100%; text-align:center; float:left; margin-top: 60px; }
.headerDescription { position: relative; left: 0; top: 0; margin-top:150px; width:100% !important; }
.headerDescription .col4 { width:95%; margin: 0 auto; float: none; }
#siteHeader:after { display:none; }
.headerEmpresaEmitente { background:url(..Images/empresa-emitente-810px.png) top center no-repeat !
important; }
.headerCertificadoDigital { background:url(..Images/certificado-digital-810px.png) top center no-
repeat !important; }
.headerEmitirEReceiver { background:url(..Images/emitir-e-receber-810px.png) top center no-repeat !
important; }
.headerVisualizar { background:url(..Images/visualizar-810px.png) top center no-repeat !important; }
.headerAutorizacaoDeUso { background:url(..Images/autorizacao-de-uso-810px.png) top center no-repeat !
important; }
.headerResponsiveWebDesign { background:url(..Images/responsive-web-design-810px.png) top center no-
repeat !important; }
.headerWindowsAzure { background:url(..Images/windows-azure-810px.png) top center no-repeat !
important; }
.headerNotaFiscalEletronica { background:url(..Images/nota-fiscal-eletronica-810px.png) top center no-
repeat !important; }
.headerProposta { background:url(..Images/proposta-810px.png) top center no-repeat !important; }
}
```

Quadro 16 – *Media query* para resoluções de até 810px de largura

Na *media query* para resoluções de até 710px de largura é reajustado o posicionamento do conteúdo da tela inicial individualmente e não mais lado a lado. Além disto, são redefinidos nos botões os limites da margem e posicionamento, como pode ser observado no Quadro 17.

```
@media only screen and (max-width: 710px) {
#mainTeaser .teaser { width: 95%; margin: 20px 0 0 15px; text-align:center; }
#mainTeaser .responsiveWebDesign { background: url(..Images/responsive-web-design-
teaser.png) center -40px no-repeat; padding-top: 100px !important; }
#mainTeaser .windowsAzure { background: url(..Images/windows-azure-teaser.png) center
0px no-repeat; padding-top: 147px !important; }
#mainTeaser .notaFiscalEletronica { background: url(..Images/nota-fiscal-eletronica-
teaser.png) center 0px no-repeat; padding-top: 141px !important; margin-top:0; }
#mainTeaser h3 a { padding-top:0; height:auto; }
#mainTeaser .button { left:50%; margin-left:-75px; margin-top:0; }
#content .articleContainer { background:none; }
#content .col6 { width: 95%; }
}
```

Quadro 17 – *Media query* para resoluções de até 710px de largura

Na *media query* para resoluções de até 480px de largura o menu é oculto e sua orientação é alterada de horizontal para vertical. Além disso, todas as ilustrações das páginas devem carregar a versão de resolução 480px das respectivas imagens, como pode ser observado no Quadro 18.

```

@media only screen and (max-width: 480px) {
#navMain h1 { width:100%; float:left; }
#navMain h1 a { margin:0 auto; float:left; background:url(..Images/home.png) no-repeat 14px; }
#navMain h1 a:hover { background:url(..Images/home-hover.png) no-repeat 14px; }
#navMain li, #navMain li a { width:100%; display:block; text-align:center; background:#1d2023 url(..Images/sprite-x.jpg) top center repeat-x; }
#navMain nav ul li a { padding: 20px 0 15px 0; border-bottom:1px solid black; }
#showHideNav, #showHideNav:active, #showHideNav:visited { margin:0 auto; position: absolute; left: 100%; bottom: 10px; width: 39px; height: 39px; background: url(..Images/menu.png) no-repeat !important; border:none !important; display: block; margin-left: -54px; text-indent: -9999px; padding:0 !important; opacity:1; }
#showHideNav:hover { background: url(..Images/menu-hover.png) no-repeat !important; }
#siteHeader .headerImage { background:url(img/header-images/default-480px.jpg) top center no-repeat }
.headerDescription { margin-top:120px; }
.headerDescription h1 { line-height:110%; font-size:1.857em; }
.headerDescription h2 { font-size:1.286em; }
.home .headerImage { width:100%; padding-bottom:30px; float:left; background:url(..Images/inicial-480px.png) top center no-repeat !important; }
.home .headerDescription { margin-top:150px; }
.headerEmpresaEmitente { width:100%; padding-bottom:30px; float:left; background:url(..Images/empresa-emitente-480px.png) top center no-repeat !important; }
.headerCertificadoDigital { width:100%; padding-bottom:30px; float:left; background:url(..Images/certificado-digital-480px.png) top center no-repeat !important; }
.headerEmitirEReceiver { width:100%; padding-bottom:30px; float:left; background:url(..Images/emitir-e-receber-480px.png) top center no-repeat !important; }
.headerVisualizar { width:100%; padding-bottom:30px; float:left; background:url(..Images/visualizar-480px.png) top center no-repeat !important; }
.headerAutorizacaoDeUso { width:100%; padding-bottom:30px; float:left; background:url(..Images/autorizacao-de-uso-480px.png) top center no-repeat !important; }
.headerResponsiveWebDesign { width:100%; padding-bottom:30px; float:left; background:url(..Images/responsive-web-design-480px.png) top center no-repeat !important; }
.headerWindowsAzure { width:100%; padding-bottom:30px; float:left; background:url(..Images/windows-azure-480px.png) top center no-repeat !important; }
.headerNotaFiscalEletronica { width:100%; padding-bottom:30px; float:left; background:url(..Images/nota-fiscal-eletronica-480px.png) top center no-repeat !important; }
.headerProposta{ width:100%; padding-bottom:30px; float:left; background:url(..Images/proposta-480px.png) top center no-repeat !important; }
.container { width:95%; }
#content { width: 95%; }
#navMain nav ul { display:none; }
}

```

Quadro 18 – *Media query* para resoluções de até 480px de largura

O resultado da utilização das *media queries* no menu pode ser visto na Figura 14.



Figura 14 – Aplicação de *media queries* no menu

### 3.4.2.5 Aplicação de imagens flexíveis

Todas as imagens do sistema web que estão contidas em algum elemento que faz referência ao CSS `content`, exposto no Quadro 19, são flexíveis.

```
#content img { width:100%; height:auto; max-width:inherit; }
```

Quadro 19 – CSS das imagens flexíveis

No entanto, o principal conceito utilizado de imagens flexíveis está relacionado diretamente ao uso das *media queries*. Cada imagem possui três resoluções diferentes, 1300px, 810px e 480px. De acordo com a resolução visualizada a imagem correspondente será carregada, pois as *media queries* encarregam-se de identificar a resolução e definir o CSS e a imagem.

A Figura 15 ilustra a principal mudança que ocorre nas imagens flexíveis de acordo com a resolução visualizada.

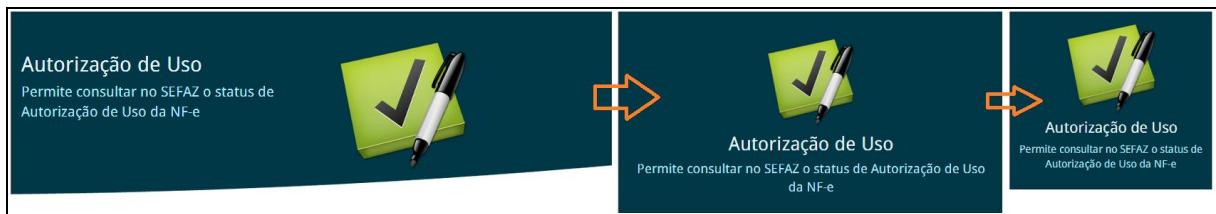


Figura 15 – Imagem flexível adaptando-se à resolução

### 3.4.2.6 Tela de certificado digital

Na tela de certificado digital é informado um nome para o certificado, o arquivo do certificado e por último a senha, então realiza-se o cadastro. O Quadro 20 ilustra o método `btnConcluir_Click` em que é salvo o certificado digital.

```
Private Sub btnConcluir_Click(sender As Object, e As System.EventArgs) Handles btnConcluir.Click
    Try
        'Verifica se foi informado o certificado digital e se a extensão é válida
        If txtUploadCertificadoDigital.HasFile AndAlso txtUploadCertificadoDigital.FileName.EndsWith(Constants.EXTENSAO_CERTIFICADO_DIGITAL) Then
            'Salva certificado
            CertificadoDigitalXml.Salvar(New CertificadoDigital() With {.Nome = txtNome.Text,
                .Conteudo = Convert.ToBase64String(txtUploadCertificadoDigital.FileBytes),
                .Senha = txtSenha.Text})

            'Informativo
            UtilsRWD.ExibirAlerta(txtUploadCertificadoDigital, String.Format("O Certificado Digital '{0}' foi salvo com sucesso!", txtNome.Text))
            'Limpa o formulário
            UtilsRWD.LimparControles(Me.Controls)
        Else
            UtilsRWD.ExibirAlerta(txtUploadCertificadoDigital, "Nenhum Certificado Digital foi selecionado.")
        End If
    Catch wrkErro As Exception
        UtilsRWD.ExibirAlerta(CType(sender, Control), String.Format("Erro: {0}", wrkErro.Message))
    End Try
End Sub
```

Quadro 20 – Rotina para salvar o certificado digital

A persistência do certificado digital é realizada em arquivo XML, através do método `Salvar` da classe `CertificadoDigitalXml`, exposto no Quadro 21.

```
Public Shared Sub Salvar(ByVal pCertificadoDigital As Objects.CertificadoDigital)
    CriarDiretorioPadrao()
    Dim wrkXmlCertificadoDigital As String = Path.Combine(DiretorioPadrao, pCertificadoDigital.Nome & ".xml")
    UtilsXml.SerializeObject(wrkXmlCertificadoDigital, pCertificadoDigital, New UTF8Encoding)
End Sub
```

Quadro 21 – Rotina para realizar a persistência do certificado digital

O modelo de como os dados são salvos no XML segue o *schema* do arquivo `CertificadoDigital.xsd`, visualizado no Quadro 22.

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="CertificadoDigital">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Nome" type="xs:string"/>
        <xs:element name="Conteudo" type="xs:string"/>
        <xs:element name="Senha" type="xs:string"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Quadro 22 – *Schema* do certificado digital

### 3.4.2.7 Tela de empresa emitente

Na tela de empresa emitente são informados o nome fantasia da empresa, o CNPJ e, por último, um certificado digital associado à empresa. Então realiza-se o cadastro. O Quadro 23 ilustra o método `btnConcluir_Click` em que é salva a empresa emitente.

```

Private Sub btnConcluir_Click(sender As Object, e As System.EventArgs) Handles btnConcluir.Click
    Try
        'Salva empresa
        EmpresaEmitenteXml.Salvar(New EmpresaEmitente() With {.NomeFantasia = txtNomeFantasia.Text,
                                                                .CNPJ = txtCNPJ.Text,
                                                                .CertificadoDigital = cboCertificadoDigital.Text})

        'Informativo
        UtilsRWD.ExibirAlerta(btnConcluir, String.Format("A Empresa Emitente '{0}' foi salva com sucesso!", txtNomeFantasia.Text))
        'Limpa o formulário
        UtilsRWD.LimparControles(Me.Controls)
    Catch wrkErro As Exception
        UtilsRWD.ExibirAlerta(CType(sender, Control), String.Format("Erro: {0}", wrkErro.Message))
    End Try
End Sub

```

Quadro 23 – Rotina para salvar a empresa emitente

A persistência da empresa emitente é realizada em arquivo XML, através do método Salvar da classe EmpresaEmitenteXml, exposto no Quadro 24.

```

Public Shared Sub Salvar(ByVal pEmpresaEmitente As Objects.EmpresaEmitente)
    CriarDiretorioPadrao()
    If Not Directory.Exists(Path.Combine(DiretorioPadrao, UtilsCampo.LimpaString(pEmpresaEmitente.CNPJ))) Then
        Directory.CreateDirectory(Path.Combine(DiretorioPadrao, UtilsCampo.LimpaString(pEmpresaEmitente.CNPJ)))
    End If
    Dim wrkXmlEmpresaEmitente As String = Path.Combine(DiretorioPadrao, UtilsCampo.LimpaString(pEmpresaEmitente.CNPJ), "Info.xml")
    UtilsXml.SerializeObject(wrkXmlEmpresaEmitente, pEmpresaEmitente, New UTF8Encoding)
End Sub

```

Quadro 24 – Rotina para realizar a persistência da empresa emitente

O modelo de como os dados são salvos no XML segue o *schema* do arquivo EmpresaEmitente.xsd, visualizado no Quadro 25.

```

<?xml version="1.0" encoding="utf-8"?>
<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="EmpresaEmitente">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="NomeFantasia" type="xs:string"/>
        <xs:element name="CNPJ" type="xs:string"/>
        <xs:element name="CertificadoDigital" type="xs:string"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>

```

Quadro 25 – Schema da empresa emitente

#### 3.4.2.8 Tela de emitir e receber

Para realizar a emissão de NF-e é selecionada a empresa emitente e o arquivo de NF-e. Então realiza-se a emissão. O Quadro 26 ilustra o trecho de código do método Enviar, disparado pelo método btnConcluir\_Click em que é recebida a NF-e.

```

Public Sub Enviar()
    Me.AssinarValidarXMLNFe()

    Dim wrkCertificadoDigital As CertificadoDigital = wrkListCertificadosDigitais.Find(
        Function(f) f.Nome = wrkListEmpresasEmitentes(cboEmpresaEmitente.SelectedIndex - 1).CertificadoDigital)

    Dim wrkCertificado As New X509Certificate2(Convert.FromBase64String(wrkCertificadoDigital.Conteudo), wrkCertificadoDigital.Senha)

    Dim oRecepcao As Object = Nothing
    Dim oCabecMsg As Object = Nothing

    'Instancia proxy
    Dim wsProxy As New WebServiceProxy(Path.Combine(Request.PhysicalApplicationPath, Constantes.DIRETORIO_SEFAZ_WEB_SERVICES, Constantes.WEB_SERVICES_RECEPCAO), wrkCertificado)

    'Define propriedades do proxy
    With wsProxy
        oRecepcao = .CriarObjeto("NfeRecepcao2")
        oCabecMsg = .CriarObjeto("nfeCabecMsg")
        .SetProp(oCabecMsg, "cUF", "42")
        .SetProp(oCabecMsg, "versaoDados", "2.00")
        .SetProp(oRecepcao, "Timeout", 60000)
        .SetProp(oRecepcao, "nfeCabecMsgValue", oCabecMsg)
    End With

    Dim docXML As New XmlDocument()
    docXML.Load(wrkLoteXml)

    'Envia a NF-e e obtém retorno do lote
    Dim XmlRetorno As XmlNode = wsProxy.InvokeXML(oRecepcao, "nfeRecepcaoLote2", New Object() {docXML})

```

Quadro 26 – Trecho de código do método enviar

Inicialmente é obtida a NF-e e o certificado digital da empresa emitente selecionada pelo método `AssinarValidarXMLNFe`. Por conseguinte, é chamado o método `Assinar` da classe `AssinaturaDigital` para realizar a assinatura de NF-e, como pode ser observado no trecho de código do Quadro 27.

```

'Criar um objeto SignedXml
Dim signedXml As SignedXml = New SignedXml(doc)
'Adicionar a chave para o documento SignedXml
signedXml.SigningKey = x509Cert.PrivateKey
'Adicionar uma transformação envolvida com a referência
Dim env As XmlDsigEnvelopedSignatureTransform = New XmlDsigEnvelopedSignatureTransform
reference.AddTransform(env)
Dim c14 As XmlDsigC14NTransform = New XmlDsigC14NTransform
reference.AddTransform(c14)
'Adicionar a referência para o objeto SignedXml
signedXml.AddReference(reference)
'Criar um novo objeto KeyInfo
Dim keyInfo As KeyInfo = New KeyInfo
'Carregar o certificado em um objeto KeyInfoX509Data e adicionar ao objeto KeyInfo
keyInfo.AddClause(New KeyInfoX509Data(x509Cert))
'Adicionar o objeto KeyInfo ao objeto SignedXml
signedXml.KeyInfo = keyInfo
signedXml.ComputeSignature()
'Obter a representação XML da assinatura e salvar para um objeto XmlElement
' it to an XmlElement object.
Dim xmlDigitalSignature As XmlElement = signedXml.GetXml
' Gravar o elemento no documento XML
nodes.AppendChild(doc.ImportNode(xmlDigitalSignature, True))

```

Quadro 27 – Trecho de código do método para assinar NF-e

Depois de assinado o arquivo de NF-e, ele é encaminhado para o envio ao *webservice* da SEFAZ, dentro de um lote de NF-es que segue o manual de integração do contribuinte da SECRETARIA DA FAZENDA (2012). A numeração dos lotes é controlada por empresa emitente, sendo incrementada a cada envio, exposto no método do Quadro 28.

```

Private Sub FinalizacaoLote(ByVal numeroLote As Integer, ByVal arquivosNFe As List(Of String))
    Dim wrkEmpresaEmitenten As EmpresaEmitente = wrkListEmpresasEmitentes(cboEmpresaEmitente.SelectedIndex - 1)
    wrkEmpresaEmitenten.NrLote += 1
    EmpresaEmitenteXml.Salvar(wrkEmpresaEmitenten)
End Sub

```

Quadro 28 – Incremento da numeração de lotes da empresa emitente

A classe `WebServiceProxy` instancia dinamicamente o *proxy* responsável pela emissão de NF-e, de acordo com o arquivo *Web Services Description Language* (WSDL) correspondente, que neste caso é `HSVRSNfeRecepcao2`. Com isto, é invocado o método do *webservice* do SEFAZ para a recepção da NF-e emitida, como pode ser observado no trecho de código do Quadro 29.

```

'Instancia proxy
Dim wsProxy As New WebServiceProxy(Path.Combine(Request.PhysicalApplicationPath,
                                                Constantes.DIRETORIO_SEFAZ_WEB_SERVICES, Constantes.WEB_SERVICES_RECEPCAO), wrkCertificado)
'Define propriedades do proxy
With wsProxy
    oRecepcao = .CriarObjeto("NfeRecepcao2")
    oCabecMsg = .CriarObjeto("nfeCabecMsg")
    .SetProp(oCabecMsg, "cUF", "42")
    .SetProp(oCabecMsg, "versaoDados", "2.00")
    .SetProp(oRecepcao, "Timeout", 60000)
    .SetProp(oRecepcao, "nfeCabecMsgValue", oCabecMsg)
End With

Dim docXML As New XmlDocument()
docXML.Load(wrkLoteXml)

'Envia a NF-e e obtém retorno do lote
Dim XmlRetorno As XmlNode = wsProxy.InvokeXML(oRecepcao, "nfeRecepcaoLote2", New Object() {docXML})

```

Quadro 29 – Invoca o método de recepção da NF-e emitida do *webservice* do SEFAZ

Depois de enviada a NF-e obtém-se o retorno do envio do lote correspondente e logo em seguida é consultado o status de retorno da autorização de uso da NF-e. Instancia-se o *proxy* novamente, de acordo com o arquivo WSDL, desta vez o `HSVRSNfeRetRecepcao2`. Com isto, é invocado o método do *webservice* do SEFAZ para a recepção do status de autorização de uso da NF-e, como pode ser observado no trecho de código do Quadro 30.

```

'Instancia proxy
wsProxy = New WebServiceProxy(Path.Combine(Request.PhysicalApplicationPath,
                                                Constantes.DIRETORIO_SEFAZ_WEB_SERVICES, Constantes.WEB_SERVICES_RET_RECEPCAO), wrkCertificado)

With wsProxy
    oRecepcao = .CriarObjeto("NfeRetRecepcao2")
    oCabecMsg = .CriarObjeto("nfeCabecMsg")
    .SetProp(oCabecMsg, "cUF", "42")
    .SetProp(oCabecMsg, "versaoDados", "2.00")
    .SetProp(oRecepcao, "Timeout", 60000)
    .SetProp(oRecepcao, "nfeCabecMsgValue", oCabecMsg)
End With

docXML.LoadXml("<?xml version='1.0' encoding='UTF-8'?'><consReciNFe xmlns='http://www.portalfiscal.inf.br/nfe' versao='2.00'><tpAmb>2</tpAmb> <nRec> &
    XmlRetorno.ChildNodes(6).ChildNodes(0).InnerXml & "</nRec> </consReciNFe>")

'Obtém retorno do status de NF-e
XmlRetorno = wsProxy.InvokeXML(oRecepcao, "nfeRetRecepcao2", New Object() {docXML})

```

Quadro 30 – Invoca o método de recepção do status de autorização de uso da NF-e do *webservice* do SEFAZ

Por fim, obtém-se o retorno da recepção do status de autorização de uso da NF-e, monta-se a *tag* correspondente e grava-se na NF-e assinada e enviada, como pode ser observado no trecho de código do Quadro 31.

```

'ProtNfe
Dim infConsSitList As XmlNodeList = docXML.GetElementsByTagName("infProt")
Dim strProtNfe As String = GeraStrProtNfe(CType(infConsSitList(0), XmlElement))

'Nfe
Dim wrkDiretorioAssinado As String = Path.Combine(EmpresaEmitenteXml.DiretorioPadrao,
    UtilsCampo.LimpaString(wrkListEmpresasEmitentes(cboEmpresaEmitente.SelectedIndex - 1).CNPJ),
    Constantes.DIRETORIO_NOTA_FISCAL_ELETRONICA,
    Constantes.DIRETORIO_NOTA_FISCAL_ELETRONICA_ENVIO, Constantes.DIRETORIO_NOTA_FISCAL_ELETRONICA_ASSINADA)

Dim wrkArquivoXmlAssinado As String = Path.Combine(wrkDiretorioAssinado, txtUploadNfe.FileName)

docXML.Load(wrkArquivoXmlAssinado)

Dim NFelist As XmlNodeList = docXML.GetElementsByTagName("NFe")
Dim NFeNode As XmlNode = NFelist(0)
Dim strNfe As String = NFeNode.OuterXml

If Not Directory.Exists(NotaFiscalEletronicaXml.DiretorioPadrao) Then Directory.CreateDirectory(NotaFiscalEletronicaXml.DiretorioPadrao)

'Grava retorno do status da NF-e no arquivo assinado
Using wrkStreamWriter As New StreamWriter(Path.Combine(NotaFiscalEletronicaXml.DiretorioPadrao, txtUploadNfe.FileName))
    wrkStreamWriter.Write("<?xml version=""1.0"" encoding=""UTF-8"" ?><NfeProc xmlns=""http://www.portalfiscal.inf.br/nfe"" versao=""2.00"">" &
        strNfe & strProtNfe & "</NfeProc>")
End Using

```

Quadro 31 – Trecho de código que grava o status de autorização de uso da NF-e

Para realizar o recebimento de NF-e também é informada a empresa emitente e o arquivo de NF-e. Então realiza-se o recebimento. O Quadro 32 ilustra método `Receber` disparado pelo método `btnConcluir_Click` em que é recebido a NF-e.

```

Public Sub Receber()
    NotaFiscalEletronicaXml.Salvar(txtUploadNfe.FileName, txtUploadNfe.FileBytes)
    UtilsRWD.ExibirAlerta(txtUploadNfe, String.Format("A Nota Fiscal Eletronica '{0}' foi 'Recebida' com sucesso!", txtUploadNfe.FileName))
End Sub

```

Quadro 32 – Rotina de recebimento de NF-e

A persistência do recebimento de NF-e é realizada em arquivo XML. Como o arquivo recebido já é um XML, então é realizada apenas uma cópia do arquivo original, através do método `Salvar` da classe `NotaFiscalEletronicaXml`, exposto no Quadro 33.

```

Public Shared Sub Salvar(ByVal pNomeArquivo As String, ByVal pConteudoArquivo As Byte())
    CriarDiretorioPadrao()
    Using wrkFileStream As New FileStream(Path.Combine(DiretorioPadrao, pNomeArquivo), FileMode.CreateNew)
        wrkFileStream.Write(pConteudoArquivo, 0, pConteudoArquivo.Count)
        wrkFileStream.Flush()
    End Using
End Sub

```

Quadro 33 – Rotina de recebimento de NF-e

### 3.4.2.9 Tela de visualizar NF-e

Na tela de visualizar NF-e é informada a empresa emitente e o tipo de NF-e, emitida ou recebida. Então realiza-se a listagem de acordo com as opções informadas. O Quadro 34 ilustra o método `CarregarGridView`, o qual lista as NF-es filtradas.



```

Private Sub CarregarGridView()

    'Carrega as NF-es
    wrkListXNFe = NotaFiscalEletronicaXml.Carregar()

    If wrkListXNFe IsNot Nothing AndAlso wrkListXNFe.Count > 0 Then

        'Monta estrutura de dados
        Dim dtbXNFe As New DataTable
        dtbXNFe.Columns.Add("InfProt_chNFe", GetType(String))
        dtbXNFe.Columns.Add("Emit_XNome", GetType(String))
        dtbXNFe.Columns.Add("Dest_XNome", GetType(String))

        Dim wrkDataRow As DataRow = Nothing

        'Adiciona as NF-es na estrutura
        For Each wrkXNFe As XNFe In wrkListXNFe
            wrkDataRow = dtbXNFe.NewRow()
            wrkDataRow.ItemArray = {wrkXNFe.InfProt_chNFe, wrkXNFe.Emit_XNome, wrkXNFe.Dest_XNome}
            dtbXNFe.Rows.Add(wrkDataRow)
        Next

        'Define a estrutura para o gridview
        dtgNotaFiscalEletronica.DataSource = dtbXNFe
        dtgNotaFiscalEletronica.DataBind()

        'Habilita controle
        btnVisualizar.Visible = True
    Else
        'Habilita controle
        btnVisualizar.Visible = False
    End If

End Sub

```

Quadro 34 – Rotina de listar NF-es

No entanto, o acesso físico do XML da NF-e é realizado pelo método `Carregar` da classe `NotaFiscalEletronicaXml`, exposto no Quadro 35. Cada XML é lido e convertido para o objeto `XNFe`, permitindo assim a manipulação dos dados da NF-e.

```

Public Shared Function Carregar() As List(Of XNFe)

    Dim wrkListXNFe As List(Of XNFe) = Nothing

    If Directory.Exists(DiretorioPadrao) Then

        Dim wrkArquivosXml() As String = Directory.GetFiles(DiretorioPadrao, "*.xml")

        If wrkArquivosXml IsNot Nothing AndAlso wrkArquivosXml.Length > 0 Then

            wrkListXNFe = New List(Of XNFe)
            Dim wrkNFeInfo As New MapXmlToNFe(Enumeradores.VersaoNFe.v2_00)
            Dim wrkXNFe As XNFe = Nothing

            For Each wrkArquivoXml As String In wrkArquivosXml
                wrkNFeInfo.GetXNFeFromFile(wrkArquivoXml)
                wrkXNFe = wrkNFeInfo.GetXNFe
                If wrkXNFe IsNot Nothing Then
                    wrkListXNFe.Add(wrkXNFe)
                End If
            Next
        End If

    End If

    Return wrkListXNFe
End Function

```

Quadro 35 – Rotina de carregar NF-es

No momento da leitura do XML da NF-e é utilizado o recurso *Language Integrated Query* (LINQ) do *framework* .NET, o qual permite realizar consultas diretamente em documentos XML usando uma sintaxe parecida com a linguagem *Structured Query Language* (SQL). O Quadro 36 ilustra o trecho de código de leitura dos dados do emitente do XML da NF-e.

```
Dim wrkNFfeEmit = From nfeEmit In xmlDocNfe...<nfe:emit> _
                  Select nfeEmit...<nfe:CNPJ>.Value, nfeEmit...<nfe:CPF>.Value, nfeEmit...<nfe:xNome>.Value, nfeEmit...<nfe:xFant>.Value,
                          nfeEmit...<nfe:xlgr>.Value, nfeEmit...<nfe:nro>.Value, nfeEmit...<nfe:xCpl>.Value, nfeEmit...<nfe:xBairro>.Value,
                          nfeEmit...<nfe:cMun>.Value, nfeEmit...<nfe:xMun>.Value, nfeEmit...<nfe:UF>.Value, nfeEmit...<nfe:CEP>.Value,
                          nfeEmit...<nfe:cPais>.Value, nfeEmit...<nfe:xPais>.Value, nfeEmit...<nfe:fone>.Value, nfeEmit...<nfe:IE>.Value,
                          nfeEmit...<nfe:IEST>.Value, nfeEmit...<nfe:IM>.Value, nfeEmit...<nfe:CNAE>.Value, nfeEmit...<nfe:CRT>.Value

With wrkNFfeEmit(0)
    wrkXNFfe.Emit_CNPJ = .CNPJ
    wrkXNFfe.Emit_CPF = .CPF
    wrkXNFfe.Emit_XNome = .xNome
    wrkXNFfe.Emit_XFant = .xFant
    wrkXNFfe.Emit_XLgr = .xlgr
    wrkXNFfe.Emit_Nro = .nro
    wrkXNFfe.Emit_XCpl = .xCpl
    wrkXNFfe.Emit_XBairro = .xBairro
    wrkXNFfe.Emit_CMun = UtilsXml.GetIntegerFromXmlWithMinValue(.cMun)
    wrkXNFfe.Emit_XMun = .xMun
    wrkXNFfe.Emit_UF = .UF
    wrkXNFfe.Emit_Cep = .CEP
    wrkXNFfe.Emit_CPais = UtilsXml.GetIntegerFromXmlWithMinValue(.cPais)
    wrkXNFfe.Emit_XPais = .xPais
    wrkXNFfe.Emit_Fone = .fone
    wrkXNFfe.Emit_IE = .IE
    wrkXNFfe.Emit_IEST = .IEST
    wrkXNFfe.Emit_IM = .IM
    wrkXNFfe.Emit_CNAE = .CNAE
    wrkXNFfe.emit_CRT = UtilsXml.GetIntegerFromXmlWithMinValue(.CRT)
End With
```

Quadro 36 – Trecho de código de leitura dos dados do emitente no XML da NF-e

#### 3.4.2.10 Tela de autorização de uso

Na tela de autorização de uso o usuário informa a empresa emitente e o tipo de NF-e, emitida ou recebida. Então realiza-se a listagem de acordo com as opções informadas. Esta funcionalidade é a mesma utilizada pela tela de visualizar.

Depois de selecionada a NF-e e o seu status de autorização de uso consultado, também é obtido o certificado digital da empresa emitente selecionada. Com isto, faz-se a leitura de alguns dados da NF-e para obter-se o endereço do *webservice* correspondente a ser utilizado, nos arquivos *UrlWebServicesSefaz\_v1\_10.xml* e *UrlWebServicesSefaz\_v2\_00.xml*. O Quadro 37 ilustra o trecho de código que obtém a *Uniform Resource Locator* (URL) do *webservice* do SEFAZ.

```

Dim wrkWsSefaz As WebServiceSefazXml = Nothing

Dim linqWsSefaz = From wsSefaz In xmlDocWebServices...<UrlWebServices:WebServicesSefaz>
                  Select wsSefaz.@versao

If linqWsSefaz IsNot Nothing AndAlso linqWsSefaz.Count > 0 Then
    wrkWsSefaz = New WebServiceSefazXml
    wrkWsSefaz.Versao = linqWsSefaz(0)
End If

'Obtem o portal de acordo com a unidade da federacao
Dim linqWsPortal = From wsPortal In xmlDocWebServices...<UrlWebServices:Portal>
                   Where wsPortal.@id = pUF
                   Select wsPortal.@id, wsPortal.@idPortalUtilizarWS, wsPortal.@UF

```

Quadro 37 – Trecho de código que obtém a URL do *webservice* do SEFAZ

Por conseguinte, com a URL do *webservice* do SEFAZ é instanciado o *proxy* correspondente e também a versão do XML da NF-e correspondente, conforme exposto no Quadro 38.

```

'Verifica a versao do layout
If pVersao = Enumeradores.VersaoNFe.v1_10 Then

    Dim wrkTConsSitNFe As New Xml.XmlTConsSitNFe_v1_07
    wrkTConsSitNFe.DadosMsg.ChNFe = pChaveNFe
    wrkTConsSitNFe.DadosMsg.TpAmb = pTpAmbiente
    wrkNfeCabecMsg = wrkTConsSitNFe.GetXmlDadosCabec
    wrkNfeDadosMsg = wrkTConsSitNFe.GetXmlDadosMsg
    wrkClientConsultaNFe = New SefazWebServices.ClientConsultaNFe_v1_07

    'Realiza consulta utilizando o certificado digital passado como parametro
    wrkRet = wrkClientConsultaNFe.ConsultaSituacaoNFe(wrkURL, pCUF, wrkNfeDadosMsg, pCertificado)

ElseIf pVersao = Enumeradores.VersaoNFe.v2_00 Then

    Dim wrkTConsSitNFe As New Xml.XmlTConsSitNFe_v2_00
    wrkTConsSitNFe.DadosMsg.ChNFe = pChaveNFe
    wrkTConsSitNFe.DadosMsg.TpAmb = pTpAmbiente
    wrkNfeDadosMsg = wrkTConsSitNFe.GetXmlDadosMsg
    wrkClientConsultaNFe = New SefazWebServices.ClientConsultaNFe_v2_00

    'Realiza consulta utilizando o certificado digital passado como parametro
    wrkRet = wrkClientConsultaNFe.ConsultaSituacaoNFe(wrkURL, pCUF, wrkNfeDadosMsg, pCertificado)

End If

```

Quadro 38 – Trecho de código que instancia o *webservice* de consulta de status de autorização de uso

Por fim, o retorno obtido da consulta de status de autorização de uso em formato XML é convertido para o objeto *XRetConsSitNFe*, permitindo assim a manipulação dos dados, conforme exposto no Quadro 39.

```

'Verifica se conseguiu retorno do SEFAZ
If Not String.IsNullOrEmpty(wrkRetConsSitNFe) Then

    'Define a versao do XML de consulta de situacao da NF-e
    Dim wrkVersaoConsSitNFe As Enumeradores.VersaoConsultaSitNFe
    If pVersao = Enumeradores.VersaoNFe.v1_10 Then
        wrkVersaoConsSitNFe = Enumeradores.VersaoConsultaSitNFe.v1_07
    ElseIf pVersao = Enumeradores.VersaoNFe.v2_00 Then
        wrkVersaoConsSitNFe = Enumeradores.VersaoConsultaSitNFe.v2_00
    End If

    'Obtem objeto com dados de retorno da consulta situação da NF-e
    Dim wrkMapXRetConsSitNFe As New MapXmlToRetConsSitNFe(wrkVersaoConsSitNFe, wrkRetConsSitNFe)
    Dim wrkXRetConsSitNFe As XRetConsSitNFe = wrkMapXRetConsSitNFe.GetXRetConsSitNFe

    'Armazena a URL do web services SEFAZ
    wrkXRetConsSitNFe.UrlWebServices = Me.UrlWebServices

    Return wrkXRetConsSitNFe
End If

```

Quadro 39 – Trecho de código que converte o retorno de consulta de status de autorização de uso

### 3.4.2.11 Processo de disponibilização no Windows Azure

Por último, foi realizado o processo de disponibilização do sistema web de emissão de NF-e no WA. Para isto tornou-se necessária a criação de uma conta no WA, pois somente usuários autenticados poderão usufruir do serviço. Embora a cobrança ocorra de acordo com os recursos utilizados, há a opção de criar-se uma conta gratuita válida por três meses, exposta na Figura 16.



Avaliação gratuita do Windows Azure por 90 dias

Teste o Windows Azure gratuitamente sem nenhum compromisso.

teste gratuitamente\*

sua avaliação gratuita contém	computação	750 horas de computação pequena por mês**
	sites	10 Sites Compartilhados***
	banco de dados relacional	1 Unidade de Banco de Dados SQL Web Edition e Business Edition****
	armazenamento	35 GB com 50.000.000 transações de armazenamento
	largura de banda	20 GB de transferência de dados de saída/transferência de dados de entrada ilimitada

Figura 16 – Avaliação gratuita do WA

Existem diversas formas de publicar uma aplicação ASP.NET no WA, sendo as mais comuns por meio do próprio portal, escolhida pelo presente trabalho, ou pelo Visual Studio. Além disto, tornou-se necessário instalar o Software *Development Kit* (SDK) do WA que

pode ser obtido através do portal, ou pelo Visual Studio.

Inicialmente na *Integrated Development Environment* (IDE) foi criado um projeto Windows Azure Cloud Service e escolhido o serviço ASP.NET Web Role, conforme visto na Figura 17.

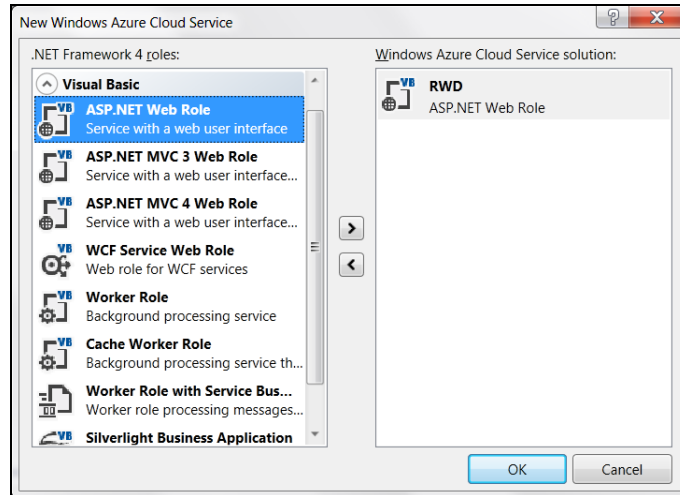


Figura 17 – Criação de ASP.NET Web Role

Em seguida foi criado um pacote (*package*) para ser importado no portal do WA. A Figura 18 detalha o menu onde é possível criar o pacote que gera os arquivos para a disponibilização no WA.

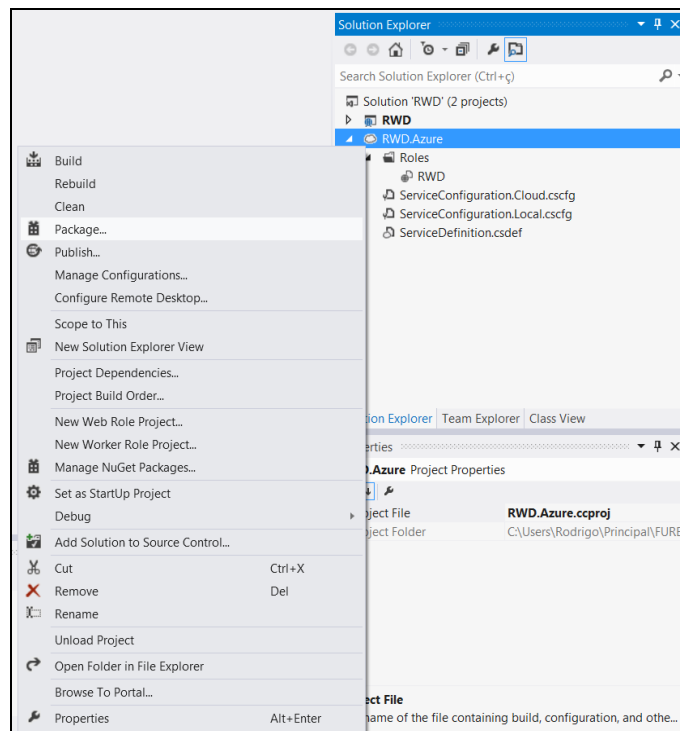


Figura 18 – Criação de pacote para disponibilização no WA

Após executado o comando de criação do pacote, o Visual Studio criará dois arquivos o `RWD.Azure.cspkg` e `ServiceConfiguration.Cloud.cscfg`. Para carregar a aplicação no

WA, basta acessar o portal e criar um serviço hospedado. Em seguida será exibida a tela de configurações do serviço hospedado, o qual solicitará os arquivos gerados pelo pacote, conforme mostrado na Figura 19.

Figura 19 – Adição dos arquivos gerados pelo pacote

Depois de feito o *upload* dos arquivos, o WA iniciará o processo de publicação e as instâncias de máquinas virtuais estarão disponíveis para utilização. Ao final deste processo a URL gerada foi <http://testrwd.cloudapp.net/pages/Login.aspx>. A Figura 20 ilustra a publicação ocorrida com sucesso do sistema web.

Figura 20 – Máquinas virtuais executando no portal do WA

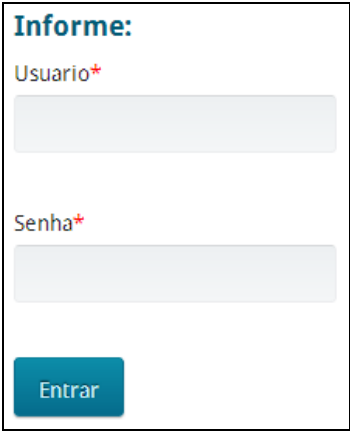
### 3.4.3 Operacionalidade da implementação

O sistema web tem como finalidade a disponibilização de um ambiente de emissão de NF-e, capaz de adaptar-se aos navegadores de diferentes dispositivos e resoluções de tela no qual serão visualizados, juntamente com alguns recursos básicos para a gestão e a organização de NF-e, focando a usabilidade, ergonomia e experiência do usuário com o sistema. Além disto, agrega a acessibilidade, proporcionando acesso simples em qualquer lugar que tenha acesso à Internet.

A seguir serão apresentadas as telas em formato tutorial, de forma a elucidar o funcionamento e o entendimento das áreas disponíveis no sistema web.

#### 3.4.3.1 Tela de autenticação

Na tela de acesso inicial do sistema, apresentada na Figura 21, é possível acessar as funções restritas do sistema, informando o nome do usuário e a senha de acesso e, posteriormente, fazendo-se o clique em `Entrar`. Após isto, caso o usuário tenha se autenticado com sucesso, ele é redirecionado para área restrita.



A imagem mostra uma interface de autenticação com o seguinte conteúdo:

- Título: **Informe:**
- Campos de entrada: **Usuario\*** e **Senha\***
- Botão: **Entrar**

Figura 21 – Tela de autenticação

#### 3.4.3.2 Tela inicial

Depois de efetuada a ação de autenticação, o usuário tem acesso à tela inicial da área restrita e as opções para seguir acessando as demais telas do sistema web. Na Figura 22 pode-

se observar a tela adaptando-se para diferentes tipos de resolução.

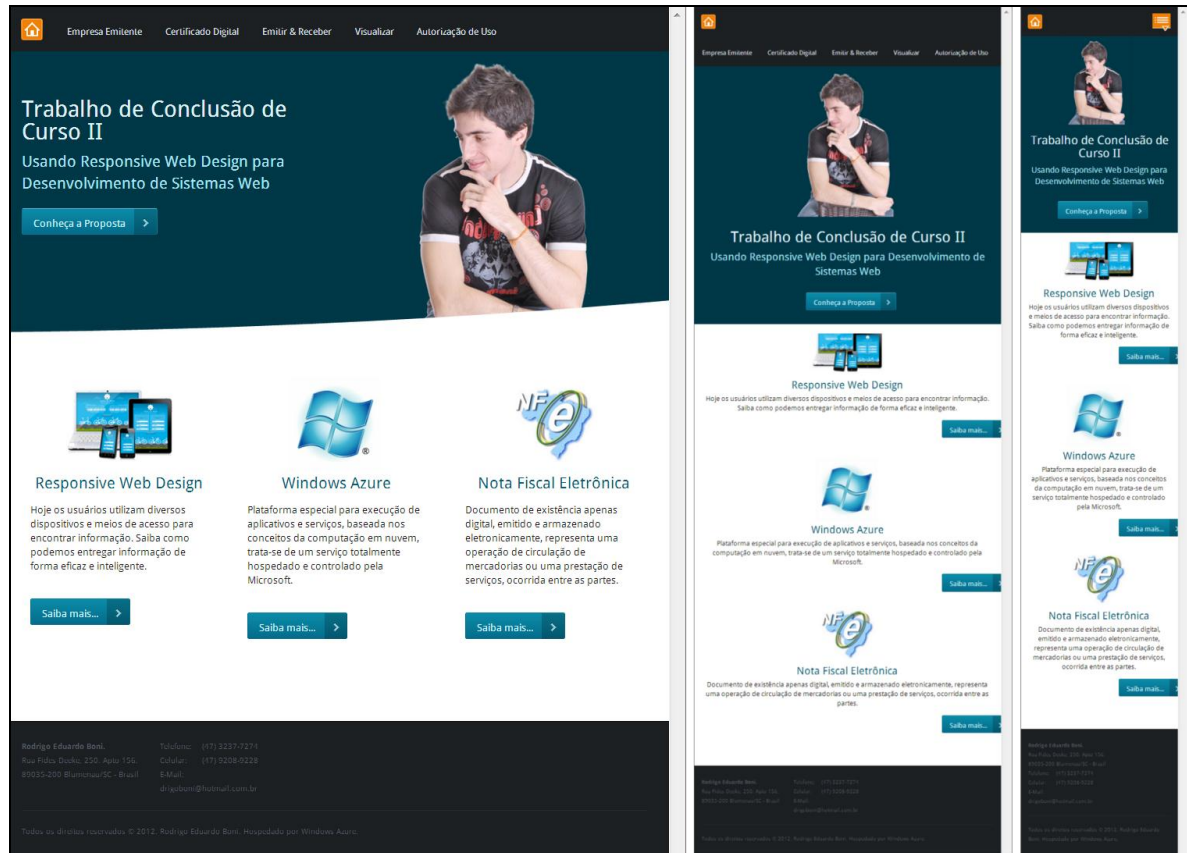


Figura 22 – Tela inicial adaptada em diferentes resoluções

Focando-se em uma interface simples e de acesso fácil, dispõe-se de um menu superior global durante toda a experiência do usuário. O ícone representado por uma casa redireciona para a tela inicial, enquanto o outro ícone representado por um menu, habilita ou desabilita a visibilidade do mesmo.

Escolheu-se esta estrutura de menu em virtude da ergonomia do sistema. Tratando-se de dispositivos maiores, como *desktops*, os quais têm a tela de tamanho avantajado, o menu superior é sempre visível e apresenta todas as opções de navegação. Caso contrário, tratando-se de dispositivos móveis, os quais têm a tela de tamanho reduzido, o menu pode ser oculto. Desta forma, as opções disponibilizadas são facilmente acessíveis a, no máximo, dois toques na tela, tornando mais viável e menos morosa a experiência homem – computador, tendo em vista os mais variados ambientes e contextos em que são feitos os acessos com os dispositivos móveis.

Além disto, é possível conhecer alguns detalhes da proposta de TCC fazendo-se o clique em *Conheça a Proposta*, assim como, os conceitos de RWD, WA e NF-e fazendo-se o clique em *Saiba mais* de cada tema.



### 3.4.3.3 Tela de certificado digital

Acessando a opção de certificado digital, no menu, o usuário visualiza a tela com alguns campos, conforme apresentado na Figura 23.

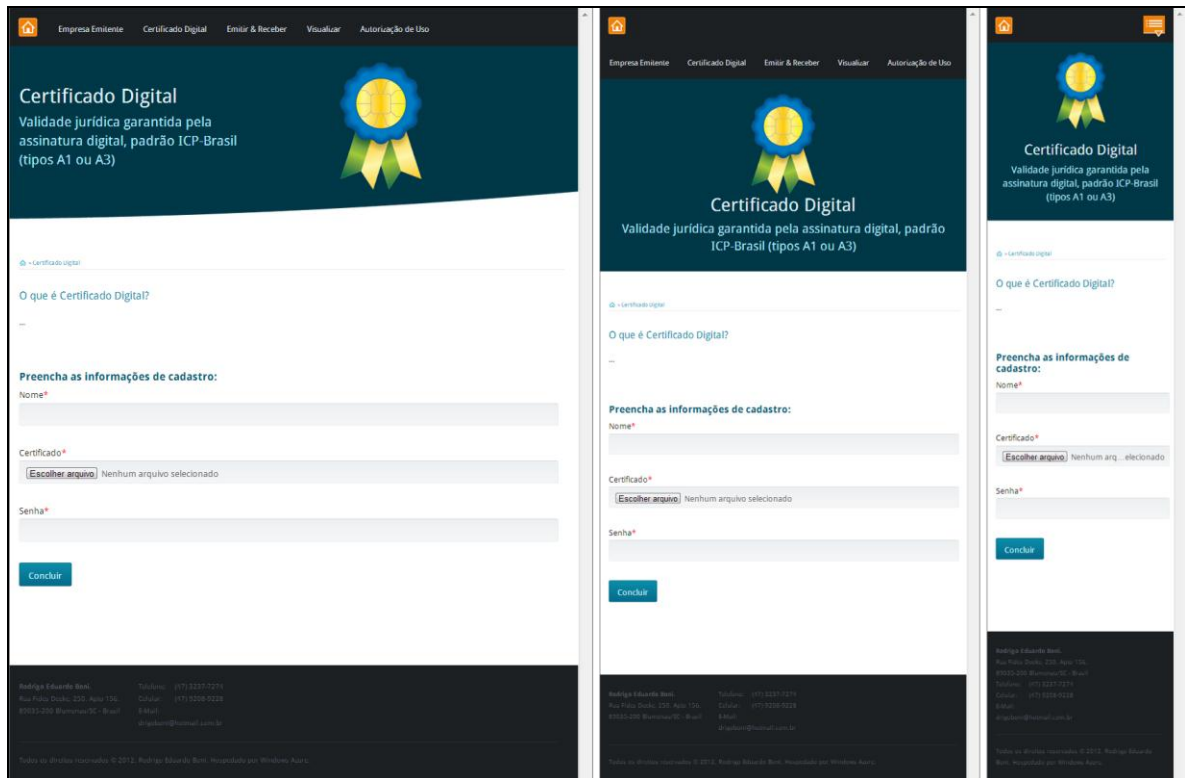


Figura 23 – Tela de certificado digital adaptada em diferentes resoluções

Esta área é destinada ao cadastro do certificado digital, que deve ser do tipo A1 ou A3, conforme determina a SECRETARIA DA FAZENDA (2012). Posteriormente, ele será utilizado na assinatura da NF-e, juntamente com o processo de emissão, ou na consulta de autorização de uso de alguma NF-e.

Inicialmente, deve-se informar o nome para o certificado digital, o qual serve como seu identificador. Em seguida, é necessário selecionar o certificado digital através do botão *Escolher arquivo*, que abre uma nova janela para seleção do arquivo físico. Por último, deve-se informar a senha do certificado e finalizar o cadastro fazendo-se o clique em *Concluir*. Como resultado, será apresentada uma mensagem de sucesso ou falha ao usuário.

### 3.4.3.4 Tela de empresa emitente

Acessando a opção de empresa emitente, no menu, o usuário visualiza a tela com

alguns campos, conforme apresenta a Figura 24.

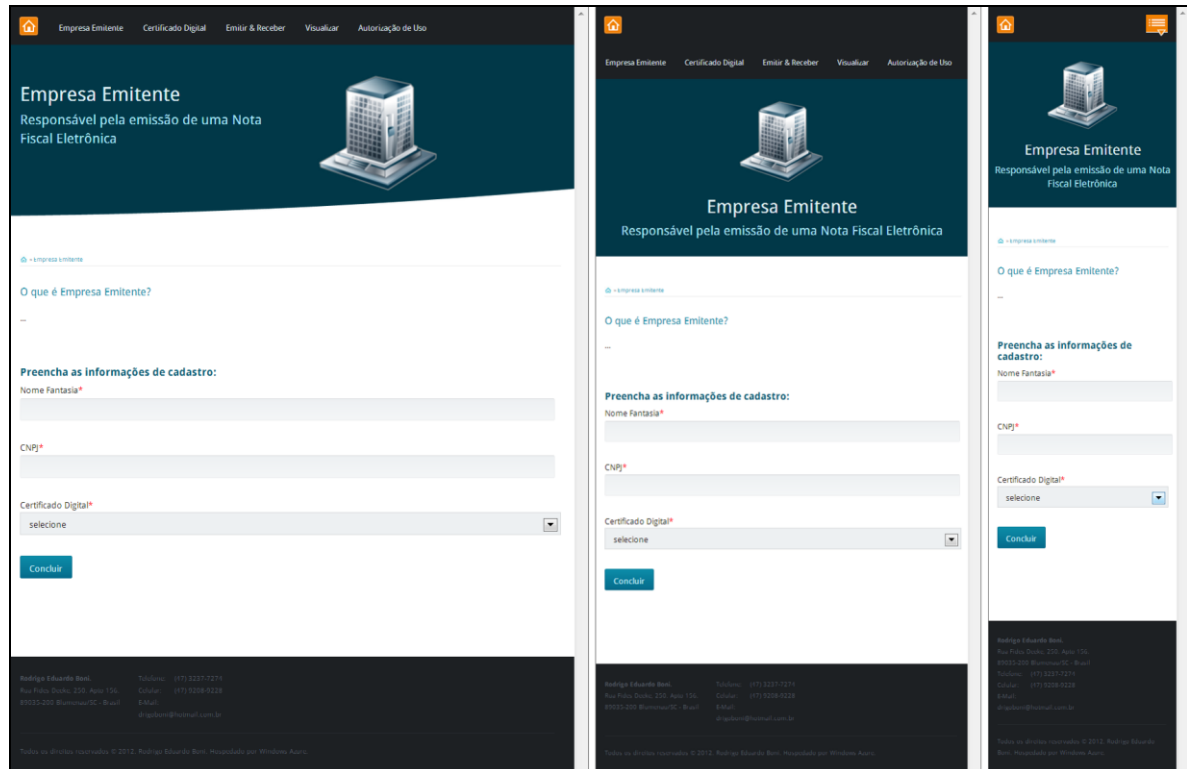


Figura 24 – Tela de empresa emitente adaptada em diferentes resoluções

Esta área é destinada ao cadastro de empresa emitente, que deve ter certificado digital correspondente, e posteriormente será utilizada no processo de emissão de NF-e, recepção, visualização e na consulta de autorização de uso de alguma NF-e.

Inicialmente, deve-se informar o nome fantasia para a empresa emitente. Em seguida, informar o CNPJ da empresa emitente, o qual serve como seu identificador. Por último, é necessário selecionar o certificado digital que havia sido gravado e finalizar o cadastro fazendo-se o clique em `Concluir`. Como resultado, será apresentada uma mensagem de sucesso ou falha ao usuário.

#### 3.4.3.5 Tela de emitir e receber

Acessando a opção de emitir e receber, no menu, o usuário visualiza a tela com alguns campos, conforme apresenta a Figura 25.

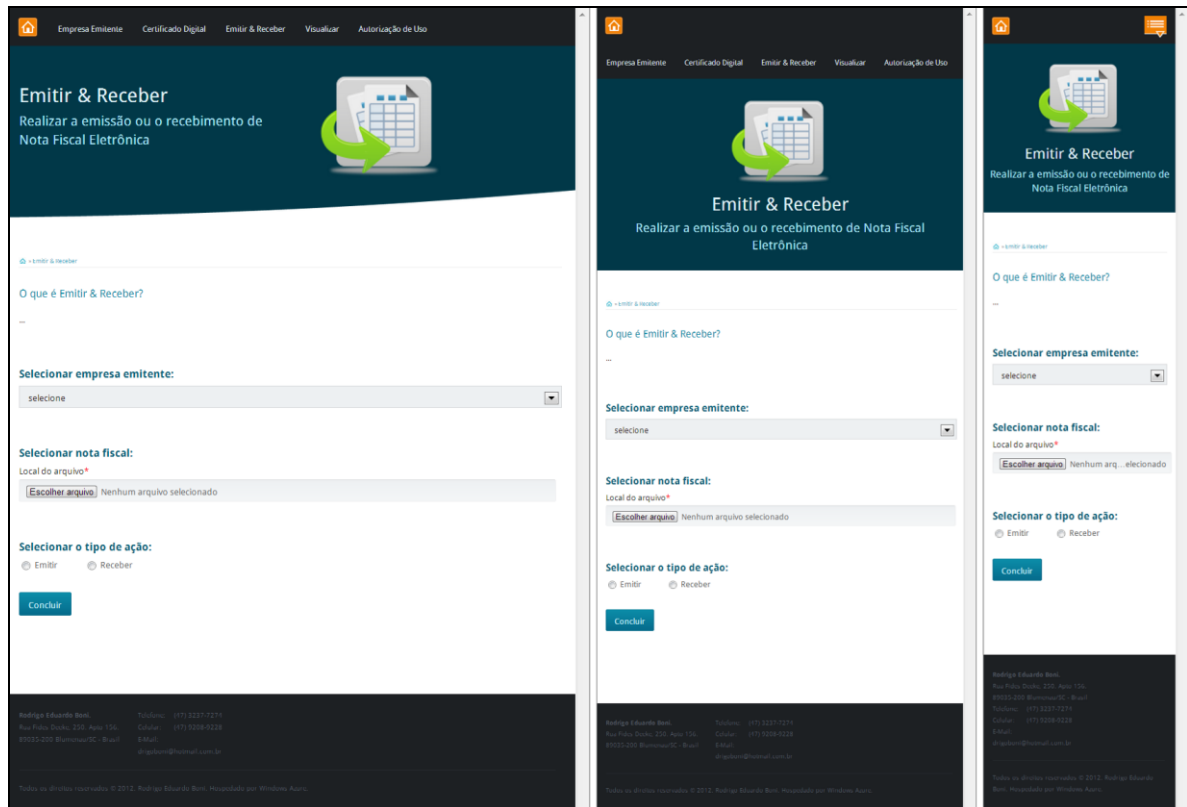


Figura 25 – Tela de emitir e receber adaptada em diferentes resoluções

Esta área é destinada a emissão e recepção de NF-e, que deve ter empresa emitente correspondente. Posteriormente será utilizada no processo de emissão e recepção de NF-e.

Inicialmente, deve-se selecionar a empresa emitente que havia sido gravada. Em seguida, selecionar a NF-e através do botão *Escolher arquivo*, que abre uma nova janela para seleção do arquivo físico. Por último, selecionar a ação, que pode ser emitir ou receber, e finalizar o processo fazendo-se o clique em *Concluir*. Como resultado, será apresentada uma mensagem de sucesso ou falha ao usuário.

#### 3.4.3.6 Tela de visualizar

Acessando a opção de visualizar, no menu, o usuário visualiza a tela com alguns campos, conforme apresenta a Figura 26.

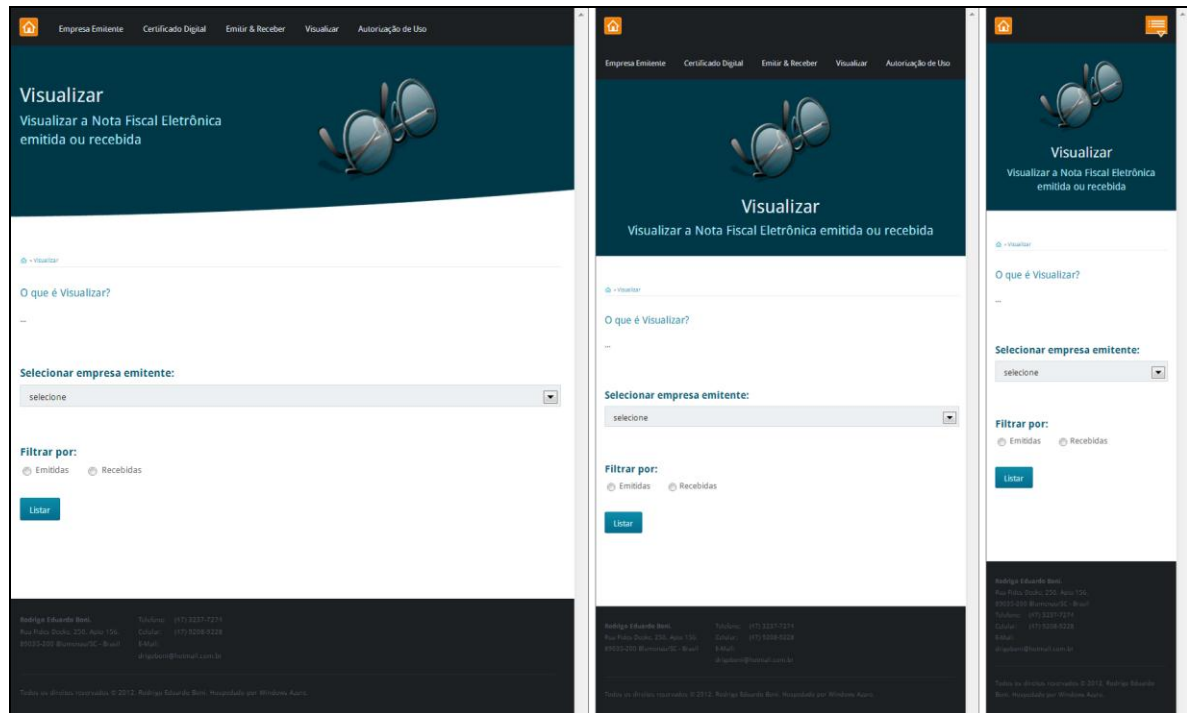


Figura 26 – Tela de visualizar adaptada em diferentes resoluções

Esta área é destinada a visualização dos dados da NF-e emitida ou recebida, que deve ter empresa emitente correspondente, e posteriormente será utilizada para visualização dos dados da NF-e emitida ou recebida.

Inicialmente, deve-se selecionar a empresa emitente que havia sido gravada. Em seguida, selecionar tipo de NF-e, que pode ser emitida ou recebida, e listar as NF-es fazendo-se o clique em **Listar**. Com isso, serão listadas as NF-es conforme os campos informados anteriormente, de maneira semelhante à Figura 27.

 The image shows three side-by-side screenshots of the web application interface displaying a list of filtered electronic tax notes (NF-es). Each screenshot shows the 'Selecionar empresa emitente:' dropdown menu with 'Empresa RWD - (15281420000241)' selected. The 'Filtrar por:' section has 'Recebidas' selected. Below this is a table titled 'Notas Fiscais Eletrônicas' with columns for 'Chave de Acesso', 'Emitente', and 'Destinatário'. The table contains two rows of data. At the bottom of each screenshot, there are 'Listar' and 'Visualizar' buttons.
 

Chave de Acesso	Emitente	Destinatário
42120301539775000100550010000000071950860484	EMPRESA RWD	NF-E EMITIDA EM AMBIENTE DE HOMOLOGACAO - SEM VALOR FISCAL
42120301539775000100550010000000091881347080	EMPRESA RWD	NF-E EMITIDA EM AMBIENTE DE HOMOLOGACAO - SEM VALOR FISCAL

Figura 27 – Tela de visualizar com NF-es filtradas

Por último, deve-se selecionar a NF-e desejada na listagem e finalizar fazendo-se o clique em **Visualizar**. Como resultado será apresentado os dados da NF-e tabulados, conforme a Figura 28.

Chave de Acesso: 4212030153975000109550010000000071950 Número da NF-e: 7 Versão XML: 2.00				Chave de Acesso: 4212030153975000109550010000000071950 Número da NF-e: 7 Versão XML: 2.00				Chave de Acesso: 4212030153975000109550010000000071950 Número da NF-e: 7 Versão XML: 2.00									
NF-e	Emitente	Destinatário	Produtos/Serviços	Totais	Transporte	Cobrança	NF-e	Emitente	Destinatário	Produtos/Serviços	Totais	NF-e	Emitente	Destinatário	Produto		
Data de Emissão	Valor Total	Situação	Protocolo				Data de Emissão	Valor Total	Situação	Protocolo		Data de Emissão	Valor Total	Situação	Protocolo		
05/03/2012	-1.79769313486232E+108	Autorizado o uso da NF-e	342120000178934				05/03/2012	-1.79769313486232E+108	Autorizado o uso da NF-e	342120000178934		05/03/2012	-1.79769313486232E+108	Autorizado o uso da NF-e	342120000178934		
<b>Emitente</b>						<b>Emitente</b>						<b>Emitente</b>					
CNPJ	Nome / Razão Social	Inscrição Estadual	UF				CNPJ	Nome / Razão Social	Inscrição Estadual	UF		CNPJ	Nome / Razão Social	Inscrição Estadual	UF		
15281420000241	Empresa RWD	256151660	SC				15281420000241	Empresa RWD	256151660	SC		15281420000241	Empresa RWD	256151660	SC		
<b>Destinatário</b>						<b>Destinatário</b>						<b>Destinatário</b>					
CNPJ	Nome / Razão Social	NF-E EMITIDA EM AMBIENTE DE HOMOLOGACAO - SEM VALOR FISCAL		UF		CNPJ	Nome / Razão Social	NF-E EMITIDA EM AMBIENTE DE HOMOLOGACAO - SEM VALOR FISCAL		UF		CNPJ	Nome / Razão Social	NF-E EMITIDA EM AMBIENTE DE HOMOLOGACAO - SEM VALOR FISCAL		UF	
99999999000191				SP		99999999000191				SP		99999999000191				SP	
<b>Emissão</b>						<b>Emissão</b>						<b>Emissão</b>					
Processo	Versão do Processo	Tipo de Emissão	Finalidade				Processo	Versão do Processo	Tipo de Emissão	Finalidade		Processo	Versão do Processo	Tipo de Emissão	Finalidade		
0	EmitNF-e 2.0.22.18	1	2				0	EmitNF-e 2.0.22.18	1	2		0	EmitNF-e 2.0.22.18	1	2		
Natureza da Operação						Natureza da Operação						Natureza da Operação					
Venda Interestadual		Tipo de Operação		Forma de Pagamento		Digest Value		Venda Interestadual		Tipo de Operação		Forma de Pagamento		Digest Value		Venda Interestadual	
		1		PmM9B		YidYSmzBwvCQ8YdPmM9B				1		PmM9B		YidYSmzBwvCQ8YdPmM9B			

Figura 28 – Tela de visualizar os dados da NF-e adaptada em diferentes resoluções

### 3.4.3.7 Tela de autorização de uso

Acessando a opção de autorização de uso, no menu, o usuário visualiza a tela com alguns campos, conforme apresenta a Figura 29.

Figura 29 – Tela de autorização de uso adaptada em diferentes resoluções

Esta área é destinada a consulta da autorização de uso de NF-e emitida ou recebida, que deve ter empresa emitente correspondente, e posteriormente será utilizada para consulta de autorização de uso da NF-e emitida ou recebida.

Inicialmente, deve-se selecionar a empresa emitente que havia sido gravada. Em seguida, selecionar tipo de NF-e, que pode ser emitida ou recebida, e listar as NF-es fazendo-

se o clique em Listar. Com isso, serão listadas as NF-es conforme os campos informados anteriormente, de maneira semelhante à Figura 30.

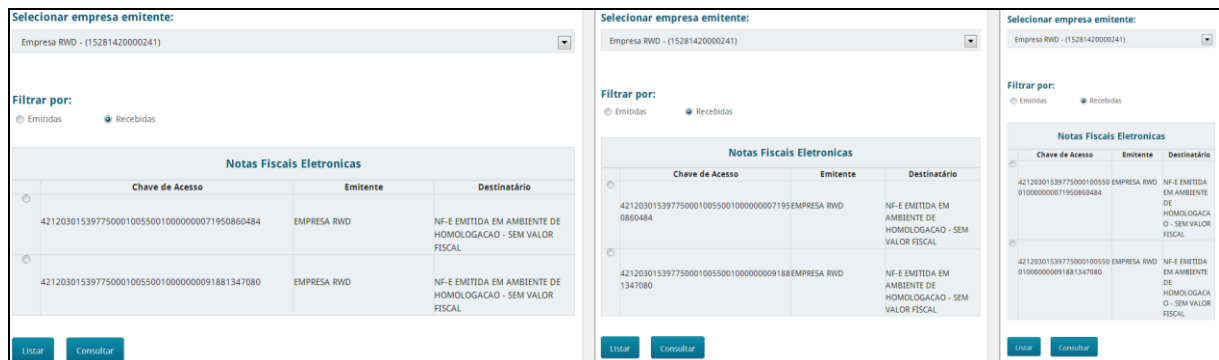


Figura 30 – Tela de autorização de uso com os dados da NF-e filtrada

Por último, deve-se selecionar a NF-e desejada na listagem e finalizar fazendo-se o clique em Consultar. Como resultado será apresentado a autorização de uso da NF-e, conforme a Figura 31.

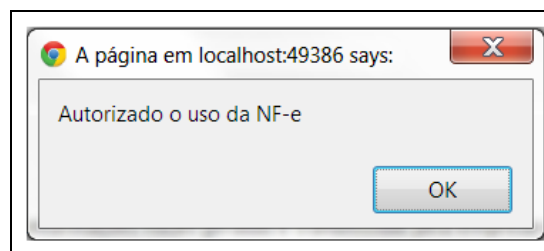


Figura 31 – Tela de autorização de uso exibindo o retorno da consulta

### 3.5 RESULTADOS E DISCUSSÃO

Para demonstrar o uso do RWD no sistema web, foram realizados testes em múltiplos navegadores e múltiplas resoluções de tela, a fim de analisar o aspecto de adaptação. Optou-se por utilizar tanto simuladores, como dispositivos reais, proporcionando uma maior realidade nos testes.

Foram realizados testes de emissão de NF-e, pois trata-se da principal funcionalidade do sistema web. Optou-se por utilizar NF-es válidas e inválidas para emissão, a fim de analisar o comportamento do sistema web para execução desta tarefa.

Por último, os resultados obtidos são confrontados com os trabalhos correlatos apresentados na fundamentação teórica.

### 3.5.1 Testes realizados em múltiplos navegadores

Os testes realizados em múltiplos navegadores consistiram em verificar se o sistema web é suportado por múltiplos navegadores indiferentemente da tecnologia. Sendo assim, o teste focou em avaliar se os principais itens que compõem o conceito do RWD funcionam nestes navegadores.

De acordo com a W3C (2012) os navegadores mais populares da atualidade são o Firefox, Google Chrome, Internet Explorer, Opera e Safari. Portanto, o sistema web desenvolvido foi testado nestes navegadores, pois são utilizados pela maioria dos usuários.

O Quadro 40 apresenta os dados obtidos nos testes realizados em múltiplos navegadores e indica se foram aceitáveis.

<b>NAVEGADOR</b>	<b>GRIDS FLEXÍVEIS</b>	<b>IMAGENS FLEXÍVEIS</b>	<b>MEDIA QUERIES</b>
Internet Explorer 7	Não	Não	Não
Internet Explorer 8	Não	Não	Não
Internet Explorer 9	Sim	Sim	Sim
Firefox 13	Sim	Sim	Sim
Firefox 14	Sim	Sim	Sim
Firefox 15	Sim	Sim	Sim
Chrome 20	Sim	Sim	Sim
Chrome 21	Sim	Sim	Sim
Chrome 22	Sim	Sim	Sim
Safari 4	Sim	Sim	Sim
Safari 5	Sim	Sim	Sim
Opera 11	Sim	Sim	Sim
Opera 12	Sim	Sim	Sim

Quadro 40 – Teste realizados de múltiplos navegadores

Observa-se, a partir das informações obtidas no teste, que os navegadores mais antigos não oferecem suporte às *media queries*, devido ao fato de serem recursos do CSS3, o qual também não é suportado por estes navegadores (W3C, 2012). O Quadro 41 apresenta a compatibilidade das *media queries* em navegadores *desktop* e móvel.

	IE	Firefox	Chrome	Safari	Opera	iOS Safari	Opera Mini	Android Browser	Blackberry Browser	Opera Mobile	Chrome for Android	Firefox for Android
18 versions back			4.0									
17 versions back			5.0									
16 versions back		2.0	6.0									
15 versions back		3.0	7.0									
14 versions back		3.5	8.0									
13 versions back		3.6	9.0									
12 versions back		4.0	10.0									
11 versions back		5.0	11.0									
10 versions back		6.0	12.0									
9 versions back		7.0	13.0		9.0							
8 versions back		8.0	14.0		9.5-9.6							
7 versions back		9.0	15.0		10.0-10.1							
6 versions back		10.0	16.0		10.5							
5 versions back		11.0	17.0	3.1	10.6			2.1				
4 versions back	5.5	12.0	18.0	3.2	11.0	3.2		2.2		10.0		
3 versions back	6.0	13.0	19.0	4.0	11.1	4.0-4.1		2.3		11.0		
2 versions back	7.0	14.0	20.0	5.0	11.5	4.2-4.3		3.0		11.1		
Previous version	8.0	15.0	21.0	5.1	11.6	5.0-5.1		4.0		11.5		
Current	9.0	16.0	22.0	6.0	12.0	6.0	5.0-7.0	4.1	7.0	12.0	18.0	15.0
Near future	10.0	17.0	23.0		12.1				10.0	12.1		
Farther future		18.0	24.0		12.5							

■ = Supported   
■ = Not supported   
■ = Partially supported   
■ = Support unknown

Fonte: adaptado de Caniuse (2012).

Quadro 41 – Compatibilidade das *media queries* nos navegadores

Com isso, percebe-se que cerca de 82% dos navegadores dão suporte às *media queries*. No entanto, caso algum navegador sem suporte às *media queries* for utilizado, não será possível seletivamente fornecer CSS com base em características diferentes.

### 3.5.2 Testes realizados em múltiplas resoluções de tela

Os testes realizados em múltiplas resoluções de tela consistiram em verificar se o leiaute do sistema web adapta-se as diferentes resoluções de tela. Sendo assim, o teste focou em avaliar os principais itens que compõem o conceito do RWD em conjunto com os aspectos de navegabilidade e legibilidade.

O simulador utilizado foi o Screenfly, da QuirkTools. Basta informar a URL do sistema web e escolher o nome do dispositivo que deseja ser testado.

O Quadro 42 apresenta os dados obtidos nos testes realizados em múltiplas resoluções de tela de *desktops* e indica se foram aceitáveis.

<b>DESKTOP</b>	<b>RESOLUÇÃO</b>	<b>ACEITÁVEL</b>
10" <i>Netbook</i>	1024x600	Sim
12" <i>Netbook</i>	1024x768	Sim
13" <i>Notebook</i>	1280x800	Sim
15" <i>Notebook</i>	1366x768	Sim
19" <i>Desktop</i>	1440x900	Sim
20" <i>Desktop</i>	1600x900	Sim
23" <i>Desktop</i>	1920x1080	Sim
24" <i>Desktop</i>	1920x1200	Sim

Quadro 42 – Testes realizados em múltiplas resoluções de tela de *desktops*



O Quadro 43 apresenta os dados obtidos nos testes realizados em múltiplas resoluções de tela de *tablets* e indica se foram aceitáveis.

<b><i>TABLET</i></b>	<b>RESOLUÇÃO</b>	<b>ORIENTAÇÃO</b>	<b>ACEITÁVEL</b>
ViewSonic ViewPad	480x800	Retrato	Sim
ViewSonic ViewPad	800x480	Paisagem	Sim
Velocity Cruz	600x800	Retrato	Sim
Velocity Cruz	800x600	Paisagem	Sim
Samsung Galaxy	600x1024	Retrato	Sim
Samsung Galaxy	1024x600	Paisagem	Sim
Apple iPad 1&2	768x1024	Retrato	Sim
Apple iPad 1&2	1024x768	Paisagem	Sim
Motorola Xoom	800x1280	Retrato	Sim
Motorola Xoom	1280x800	Paisagem	Sim

Quadro 43 – Testes realizados em múltiplas resoluções de tela de *tablets*

O Quadro 44 apresenta os dados obtidos nos testes realizados em múltiplas resoluções de tela de *mobiles* e indica se foram aceitáveis.

<b><i>MOBILE</i></b>	<b>RESOLUÇÃO</b>	<b>ORIENTAÇÃO</b>	<b>ACEITÁVEL</b>
Motorola RAZR V3m	176x220	Retrato	Não
Motorola RAZR V3m	220x176	Paisagem	Não
Motorola RAZR V8	240x320	Retrato	Sim
Motorola RAZR V8	320x240	Paisagem	Sim
BlackBerry 8300	320x240	Retrato	Sim
BlackBerry 8300	240x320	Paisagem	Sim
Apple iPhone 3&4	320x480	Retrato	Sim
Apple iPhone 3&4	480x320	Paisagem	Sim
LG Optimus S	320x480	Retrato	Sim
LG Optimus S	480x320	Paisagem	Sim
Google Nexus S	480x800	Retrato	Sim
Google Nexus S	800x480	Paisagem	Sim
ASUS Galaxy 7	480x800	Retrato	Sim
ASUS Galaxy 7	800x480	Paisagem	Sim

Quadro 44 – Testes realizados em múltiplas resoluções de tela de *mobiles*

O Quadro 45 apresenta os dados obtidos nos testes realizados em múltiplas resoluções de tela de televisões e indica se foram aceitáveis.

<b>TELEVISÃO</b>	<b>RESOLUÇÃO</b>	<b>ACEITÁVEL</b>
480p	640x480	Sim
720p	1280x720	Sim
1080p	1920x1080	Sim

Quadro 45 – Testes realizados em múltiplas resoluções de tela de televisões

Todos os testes apresentaram com sucesso os itens em avaliação, porém no terceiro teste observou-se que os dispositivos com resolução de tela menor que 240px, torna-se impraticável a questão de legibilidade e a navegabilidade é comprometida.

Há aspectos cruciais de uma experiência web, os quais não são possíveis de serem testados apenas em simuladores. Neste caso, deve-se testar em dispositivos reais para obter-se o desempenho verdadeiro do sistema web, bem como, fator de forma, densidade de pixels e

do impacto da rede, entre outros.

O Quadro 46 apresenta os dados obtidos nos testes realizados em múltiplas resoluções de tela de dispositivos reais e indica se foram aceitáveis.

<b>MOBILE</b>	<b>RESOLUÇÃO</b>	<b>ORIENTAÇÃO</b>	<b>ACEITÁVEL</b>
Motorola Spice	240x320	Retrato	Sim
Motorola Spice	320x240	Paisagem	Sim
Apple iPhone 4	320x480	Retrato	Sim
Apple iPhone 4	480x320	Paisagem	Sim
Samsung Galaxy Y	240x320	Retrato	Sim
Samsung Galaxy Y	320x240	Paisagem	Sim

Quadro 46 – Testes realizados em múltiplas resoluções de tela de *mobiles*

Observou-se que embora o teste tenha sido feito em dispositivos reais, apresentou-se o mesmo resultado obtido no simulador.

### 3.5.3 Testes realizados de emissões de NF-e

Os testes realizados de emissões de NF-e consistiram em verificar a emissão da NF-e com arquivos válidos e inválidos. Sendo assim, o teste focou em avaliar o comportamento do sistema web para execução desta tarefa.

Foram enviados dois arquivos de NF-e: o primeiro não atende ao leiaute da NF-e enquanto que o segundo atende, conforme o manual de integração do contribuinte da SECRETARIA DA FAZENDA (2012). Após isto, foi verificado no portal do SEFAZ se a NF-e foi realmente emitida.

O Quadro 47 apresenta os testes de emissão de NF-e.

<b>NF-E</b>	<b>SISTEMA WEB</b>	<b>SEFAZ</b>
Inválida	Erro	Não emitida
Válida	Sucesso	Emitida

Quadro 47 – Teste de emissão de NF-e

Observou-se que se houver qualquer irregularidade no leiaute da NF-e será apresentado erro no sistema web, caso contrário será emitida no SEFAZ.

### 3.5.4 Vantagens e desvantagens do RWD

Com o desenvolvimento deste trabalho, percebe-se que as vantagens e desvantagens da utilização do RWD estão de acordo com as que foram citadas pelos autores Marcotte (2010),

Gustafson (2012), Knight et al. (2011) e Webdesignshock (2011). O número crescente de navegadores e a diversidade de dispositivos nas mais variadas resoluções de tela, layouts estáticos não são mais suficientes para acomodar a quantidade crescente destes usuários. Desde os *desktops* até os dispositivos móveis, há uma necessidade crescente de que um layout seja flexível e que atenda à toda esta diversidade de resoluções. Sendo assim, apontam-se as vantagens do RWD:

- a) permite que o sistema web adapte-se perfeitamente para qualquer resolução de tela;
- b) como o sistema web é desenvolvido uma única vez, reduz significativamente o custo;
- c) a capacidade de projetar e construir o sistema web em um único projeto economiza tempo;
- d) por ser um conceito novo, há uma grande probabilidade de que a concorrência ainda não o implementou;
- e) o sistema web irá resistir mais tempo, pois estará preparado para a diversidade de dispositivos conectados que surgirão;
- f) fornece uma grande experiência aos usuários e torna-se mais amigável, pois adapta-se a qualquer dispositivo;
- g) melhora o ranking de pesquisa do sistema web, pois cada página terá uma única URL, preservando sua relevância para os buscadores;
- h) melhora as estatísticas do sistema web nas redes sociais, pois cada página terá uma única URL;
- i) o sistema web é mais fácil de manter, porque não envolve quaisquer componentes do lado do servidor. Apenas é preciso modificar o CSS de uma página para alterar seu layout em um dispositivo específico;
- j) não há redirecionamento entre versão *desktop* e móvel, conseqüentemente não adiciona carga desnecessária para o servidor;
- k) atinge um número maior de usuários, pois adapta-se a qualquer dispositivo;
- l) menos manutenção, pois o sistema web é único para qualquer dispositivo;
- m) reduz custo de hospedagem, pois será apenas um sistema web para ser hospedado;
- n) menos chance de erro, pois apenas um sistema web deverá ser atualizado;
- o) evita problemas canônicos, como conteúdo duplicado.

Pontos de vista negativos são tão importantes quanto os positivos, porque a existência de uma contrapartida oferece uma visão para equilibrar as cargas. Cria-se um debate e,

finalmente, ajuda a considerar se o conceito do RWD é bom ou ruim. Sendo assim, apontam-se as desvantagens do RWD:

- a) elaborar um layout totalmente funcional, bem desenhado e suportado por múltiplos navegadores envolve uma quantidade considerável de trabalho, especialmente durante a criação do CSS;
- b) apesar das *media queries* facilitarem muito o trabalho do desenvolvedor, é preciso criar um CSS completo e todos os adicionais para as diferentes resoluções;
- c) como requer uma quantidade maior de trabalho, também pode arcar com custos extras;
- d) a imagem é completamente baixada, independentemente do tamanho da resolução da tela, em seguida é feito o redimensionamento e a otimização da imagem para caber na tela. Em outras palavras, baixa-se um arquivo grande desnecessariamente. Tratando-se de dispositivos móveis o problema pode se agravar com a latência da conexão com a Internet;
- e) se a imagem for redimensionada e otimizada, conseqüentemente consome *Central Processing Unit* (CPU) e memória intensiva. Muitos dispositivos têm memória disponível limitada para páginas e imagens, e superar suas limitações de memória resulta em incompleta exposição e pode causar outros problemas;
- f) ocultar imagens com *media queries* de maneira equivocada não impede o navegador de baixá-las. Entende-se como maneira equivocada utilizar propriedade *display none* a fim de ocultar a imagem, ou alterar através de *media query* a resolução da imagem;
- g) as *media queries* não otimizam o HTML ou JavaScript, pois não removem o que não é executado, ou apresentado, apenas escondem;
- h) as *media queries* ainda não são suportadas por alguns navegadores de dispositivos móveis;
- i) não reconhece o contexto móvel, porque só adequa um sistema web para caber numa resolução, mas omite os principais recursos de dispositivos móveis, como a localização do usuário, a conectividade, o tempo que o usuário tem e as informações que o usuário precisa, limitações de hardware, software de desempenho, potencial total do dispositivo, aspectos como iluminação, som, visibilidade, acessibilidade, privacidade, entre outras.

### 3.5.5 Comparativo dos trabalhos correlatos

Nesta seção, alguns trabalhos com características semelhantes ao do sistema desenvolvido serão apresentados e comparados com o mesmo.

Comparado ao trabalho de Marcotte (2010), o presente trabalho contempla o uso dos conceitos do RWD em sua totalidade. Foi utilizado *grid* flexível de nove colunas, ao invés de duas colunas. As imagens além de serem flexíveis, possuem três resoluções diferentes, 1300px, 810px e 480px que são carregadas de acordo com a resolução da tela. Marcotte (2010) utiliza apenas imagens flexíveis em uma única resolução, não otimizando o tempo para carregá-las para as resoluções diferentes. Por último, as *media queries* utilizadas pelo presente trabalho são para as resoluções com largura menores que 480px, 710px, 810px, 960px e 1040px, enquanto Marcotte (2010) utiliza as *media queries* para as resoluções com largura menores que 400px e 600px, ou com no mínimo 1300px.

Um dos principais itens que compõe o RWD foi elucidado por Gustafson (2012), as *media queries*. O presente trabalho fez toda a concepção de um sistema web para que utilize os conceitos do RWD, ao invés de aplicá-los em um sistema web já existente, como fez Gustafson (2012). Além disto, o sistema web desenvolvido teve como objetivo ser adaptado para um número muito maior de navegadores e dispositivos. Ao contrário disto, Gustafson apenas preocupou-se com duas resoluções do navegador e para os dispositivos iPhone e iPad.

Com base no trabalho de Linhares (2011), pode-se definir as características necessárias para prover disponibilidade e escalabilidade através da plataforma em nuvem WA. Ambos os trabalhos utilizaram a escalabilidade de instâncias de máquinas virtuais, a fim de obter-se maior disponibilidade e desempenho. No entanto, as instâncias utilizadas pelo presente trabalho consumiam os mesmos recursos, isto de forma competitiva, enquanto Linhares (2011) alocava uma instância por recurso, logo não havia a competitividade por recursos e conseqüentemente o aumento de disponibilidade e desempenho.

## 4 CONCLUSÕES

O desenvolvimento do trabalho atendeu a todos os objetivos propostos. O primeiro objetivo atendido foi o de identificar vantagens e desvantagens na utilização do RWD, que tornaram-se cada vez mais claros durante o ciclo de desenvolvimento do sistema web.

O segundo objetivo atendido foi o de adaptar o sistema web desenvolvido para a diversidade de dispositivos e navegadores. Sendo assim, o próprio RWD já direciona para que isto seja possível. Embora os testes de múltiplos navegadores e múltiplas resoluções de tela tenham sido satisfatórios, percebe-se que o RWD não é uma solução final, mas serve como um passo para esta direção. O RWD está em fase inicial e muitas das técnicas estão sendo melhoradas, apesar dos benefícios oferecidos, ainda há a necessidade de capturar as vantagens e a gama de recursos de cada dispositivo.

O terceiro objetivo atendido foi de desenvolver um sistema web para emissão de NF-e, aplicando os conceitos do RWD. A partir de um XML de NF-e válido, ele é assinado digitalmente e emitido para o *webservice* do SEFAZ. Em seguida é salvo o status de autorização de uso. Resulta-se num processo de emissão de NF-e completo, onde há o envio e a consequente obtenção de retorno do envio, com exceção da digitação da NF-e.

O quarto objetivo atendido foi o de disponibilizar o sistema web na plataforma em nuvem WA. Tornou-se necessário apenas criar uma conta do WA, em seguida acessar o portal, por conseguinte criar e configurar um serviço hospedado para disponibilização do sistema web.

Quanto às tecnologias utilizadas, a IDE de desenvolvimento Visual Studio 2012, mostrou-se suficiente e eficiente para o desenvolvimento. Todos os recursos utilizados tiveram total suporte do *framework* .NET e da plataforma ASP.NET. Além disto, o WA já está integrado e possui SDK, facilitando o desenvolvimento de aplicações em nuvem.

Como limitação, o sistema web não permite a gerência dos certificados digitais e empresas emitentes já cadastradas, como edição e exclusão. Assim como não há nenhuma forma de cadastro de usuários, muito menos de gerência. Além disto, não é possível fazer a digitação da NF-e. O sistema web apenas assina digitalmente e envia ao *webservice*. O trabalho de gerar os arquivos XML das NF-es ficam a cargo de outros sistemas.

Como vantagem, o sistema web une o conceito de RWD (introduzido em 2010) com o de computação em nuvem, a fim de tornar as aplicações mais independentes em termos de uso e acesso, além de abrir novas possibilidades ainda não exploradas.

Observou-se que o conceito do RWD é muito extenso e ao mesmo tempo interessante, pois pensa no futuro de uma forma diferente. Trata-se de uma possível solução, uma maneira de projetar de forma mais completa para a flexibilidade inerente da web (MARCOTTE, 2011, p. 139).

Observou-se também que o sistema web que utiliza o RWD com a plataforma em nuvem WA pode obter um comportamento adaptativo e elástico, tornando-o mais independente de limitações de uso e intensidade. Além disto, segundo Taurion (2009, p. 33), “dispor de ambientes dinâmicos, autoajustáveis, de alta disponibilidade e escalabilidade, com custos de propriedade adequados torna-se o grande desafio dos gestores de tecnologia das organizações e governos”.

Por fim, o sistema web desenvolvido neste trabalho visou possibilitar que as informações sejam entregues da melhor maneira possível, de forma eficaz e inteligente.

#### 4.1 EXTENSÕES

Durante a especificação e implementação do sistema web surgiram algumas sugestões que podem ser agregadas ou melhoradas ao desenvolvido deste trabalho.

Inicialmente, sugere-se carregar primeiro o conteúdo essencial. Podem haver elementos no sistema web que sirvam como conteúdo auxiliar. Deve-se carregar estes elementos depois do conteúdo principal, recorrendo ao JavaScript. Além disto, todas as imagens quando redimensionadas poderiam ter sua resolução reduzida, o que resultaria em uma otimização do tempo de carregamento.

Outro ponto importante seria aumentar o tamanho do *grid* flexível para as resoluções de tela maiores e conseqüentemente aproveitar melhor o espaço ocupado pelo conteúdo das páginas. Já para as resoluções de tela menores, as tabelas poderiam inverter a orientação do cabeçalho, alterando de horizontal para vertical, apresentando somente duas colunas no formato descrição e valor.

Por fim, a plataforma do WA fornece serviços para auxílio ao desenvolvimento de aplicações que utilizam cache de memória. Seria interessante o desenvolvimento de um sistema web que explore serviços onde se faz necessária a utilização de cache, a fim de melhorar o desempenho do sistema. Além disto, também poderia ser utilizada a nuvem para o processamento paralelo, ou distribuído das principais rotinas do sistema.

## REFERÊNCIAS BIBLIOGRÁFICAS

ANTONOPOULOS, Nick; GILLAM, Lee. **Cloud computing: principles, systems and applications**. New York: Springer, 2010.

CANIUSE. **Can I use CSS3 Media Queries**. [S.l.], 2012. Disponível em: <<http://caniuse.com/css-mediaqueries>>. Acesso em: 07 nov. 2012.

CASTELLS, Manuel. **A galáxia da internet: reflexões sobre a internet, os negócios e a sociedade**. Rio de Janeiro: Zahar, 2003.

DUARTE, Roberto D. **Big brother fiscal - III: o Brasil na era do conhecimento**. 3. ed. Belo Horizonte: Ideas At Work, 2010.

GUSTAFSON, Aaron. Adaptive layouts with media queries. **Net Magazine**, Bath, n. 205, July 2012. Disponível em: <<http://www.netmagazine.com/tutorials/adaptive-layouts-media-queries>>. Acesso em: 07 nov. 2012.

KNIGHT, Kayla et al. **Professional web design**. 2nd. ed. Freiburg: Smashing, 2011.

KOMMALAPATI, Hanu. Windows Azure platform for enterprises. **MSDN Magazine**, [S.l.], v. 25, n. 2, 2010. Disponível em: <<http://msdn.microsoft.com/pt-br/magazine/ee309870.aspx>>. Acesso em: 07 nov. 2012.

LINHARES, Ricardo. **Integração de aplicações e serviços utilizando computação na nuvem com a plataforma Microsoft Windows Azure**. 2011. 74 f. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) – Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau. Disponível em: <[http://www.bc.furb.br/docs/MO/2011/348520\\_1\\_1.pdf](http://www.bc.furb.br/docs/MO/2011/348520_1_1.pdf)>. Acesso em: 07 nov. 2012.

MARCOTTE, Ethan. Responsive web design. **A List Apart**, New York, n. 306, May 2010. Disponível em: <<http://www.alistapart.com/articles/responsive-web-design/>>. Acesso em: 07 nov. 2012.

\_\_\_\_\_. **Responsive web design**. New York: A Book Apart, 2011.

PRIMAK, Fábio V. **Infortabilidade: a contabilidade na era da informática**. Rio de Janeiro: Ciência Moderna, 2009.

SECRETARIA DA FAZENDA. **Portal de nota fiscal eletrônica**. [S.l.], 2012. Disponível em: <<http://www.nfe.fazenda.gov.br/PORTAL/principal.aspx>>. Acesso em: 07 nov. 2012.

REDKAR, Tejaswi. **Windows Azure platform: unlock the power of cloud computing**. New York: Apress, 2009.



TAURION, Cezar. **Cloud computing – computação em nuvem:** transformando o mundo da tecnologia da informação. Rio de Janeiro: Brasport, 2009.

VERAS, Manoel. **Virtualização:** componente central do datacenter. Rio de Janeiro: Brasport, 2011.

W3C. [S.l.], 2012. Disponível em: <<http://www.w3.org/>>. Acesso em: 07 nov. 2012.

WEBDESIGNSHOCK. **Responsive web design, most complete guide.** [S.l.], 2011. Disponível em: <<http://www.webdesignshock.com/responsive-web-design/>> . Acesso em: 07 nov. 2012.

## APÊNDICE A – Descrição dos casos de uso

Neste apêndice são apresentados em detalhes os casos de uso do sistema web previstos no diagrama da seção 3.2.1.

O Quadro 48 detalha o caso de uso UC01, apresentado na Figura 8.

<b>UC01 – Emitir NF-e</b>	
<b>Resumo</b>	Realizar o envio da NF-e ao SEFAZ via <i>upload</i> .
<b>Requisito atendido</b>	RF01.
<b>Pré-condição</b>	A NF-e deve ser previamente gerada e estar em formato XML.
<b>Cenário principal</b>	1 – O Usuário deve selecionar a empresa emitente; 2 – O Usuário deve selecionar a NF-e desejada; 3 – O Usuário deve selecionar a ação de emissão de NF-e; 4 – O Usuário deve confirmar a ação de emissão de NF-e; 5 – O sistema recebe a NF-e; 6 – O sistema assina digitalmente a NF-e com o certificado digital da empresa emitente selecionada; 7 – O sistema envia a NF-e em um lote para o SEFAZ; 8 – O sistema obtém o retorno do lote da NF-e; 9 – O sistema obtém o status de autorização de uso da NF-e.
<b>Exceção 01</b>	No passo 5 do cenário principal, caso ocorra alguma inconsistência na validação como campo obrigatório, ou dados informados incorretamente, o sistema web apresenta mensagem referente ao problema encontrado.
<b>Exceção 02</b>	No passo 6 do cenário principal, caso a NF-e selecionada não atende o leiaute da NF-e, conforme o manual de integração do contribuinte da SECRETARIA DA FAZENDA (2012), a NF-e não será emitida.
<b>Pós-condição</b>	A NF-e emitida estará disponível no sistema web para visualização dos dados, ou para consultar o status de autorização de uso.

Quadro 48 – Detalhamento do caso de uso UC01

O Quadro 49 detalha o caso de uso UC02, apresentado na Figura 8.

<b>UC02 – Cadastrar empresas emitentes de NF-e</b>	
<b>Resumo</b>	Realizar o cadastro de uma nova empresa emitente de NF-e no sistema web.
<b>Requisito atendido</b>	RF02.
<b>Pré-condição</b>	O sistema deve ter algum certificado digital cadastrado.
<b>Cenário principal</b>	1 – O Usuário deve informar a razão social da empresa emitente; 2 – O Usuário deve informar o CNPJ da empresa emitente; 3 – O Usuário deve selecionar o certificado digital da empresa emitente; 4 – O Usuário deve confirmar a ação de cadastro de empresa emitente; 5 – O sistema grava as informações da empresa emitente em arquivo XML.
<b>Exceção 01</b>	No passo 4 do cenário principal, caso ocorra alguma inconsistência na validação como campo obrigatório, ou dados informados incorretamente, o sistema web apresenta mensagem referente ao problema encontrado.
<b>Pós-condição</b>	O CNPJ da empresa emitente estará cadastrado no sistema web.

Quadro 49 – Detalhamento do caso de uso UC02

O Quadro 50 detalha o caso de uso UC03, apresentado na Figura 8.

<b>UC03 – Cadastrar certificado digital</b>	
<b>Resumo</b>	Realizar o cadastro de um novo certificado digital no sistema web.
<b>Requisito atendido</b>	RF03.
<b>Pré-condição</b>	O certificado digital deve ser do tipo A1 ou A3 conforme determina o SEFAZ.
<b>Cenário principal</b>	1 – O <b>Usuário</b> deve informar o nome do certificado digital; 2 – O <b>Usuário</b> deve selecionar o certificado digital desejado; 3 – O <b>Usuário</b> deve informar a senha do certificado digital; 4 – O <b>Usuário</b> deve confirmar a ação de cadastro de certificado digital; 5 – O sistema grava as informações do certificado digital em arquivo XML.
<b>Exceção 01</b>	No passo 4 do cenário principal, caso ocorra alguma inconsistência na validação como campo obrigatório, ou dados informados incorretamente, o sistema web apresenta mensagem referente ao problema encontrado.
<b>Pós-condição</b>	O certificado digital estará cadastrado no sistema web.

Quadro 50 – Detalhamento do caso de uso UC03

O Quadro 51 detalha o caso de uso UC04, apresentado na Figura 8.

<b>UC04 – Receber NF-e</b>	
<b>Resumo</b>	Realizar o recebimento da NF-e no sistema web via <i>upload</i> .
<b>Requisito atendido</b>	RF04.
<b>Pré-condição</b>	A NF-e deve ser previamente gerada e estar em formato XML.
<b>Cenário principal</b>	1 – O <b>Usuário</b> deve selecionar a empresa emitente; 2 – O <b>Usuário</b> deve selecionar a NF-e desejada; 3 – O <b>Usuário</b> deve selecionar a ação de recebimento de NF-e; 4 – O <b>Usuário</b> deve confirmar a ação de recebimento de NF-e; 5 – O sistema recebe a NF-e.
<b>Exceção 01</b>	No passo 4 do cenário principal, caso ocorra alguma inconsistência na validação como campo obrigatório, ou dados informados incorretamente, o sistema web apresenta mensagem referente ao problema encontrado.
<b>Pós-condição</b>	A NF-e estará integrada no sistema web e disponível para consulta.

Quadro 51 – Detalhamento do caso de uso UC04

O Quadro 52 detalha o caso de uso UC05, apresentado na Figura 8.

<b>UC05 – Visualizar NF-e</b>	
<b>Resumo</b>	Visualizar os dados da NF-e.
<b>Requisito atendido</b>	RF05.
<b>Pré-condição</b>	A NF-e deve ter sido enviada ou recebida pelo sistema web.
<b>Cenário principal</b>	1 – O <b>Usuário</b> deve selecionar a empresa emitente; 2 – O <b>Usuário</b> deve selecionar o filtro por NF-es emitidas ou recebidas; 3 – O <b>Usuário</b> deve confirmar a ação de listar as NF-es; 4 – O sistema filtra e apresenta as NF-es; 5 – O <b>Usuário</b> deve selecionar a NF-e da listagem; 6 – O <b>Usuário</b> deve confirmar a ação de visualizar a NF-e; 7 – O sistema apresenta os dados da NF-e tabulados.
<b>Exceção 01</b>	No passo 3 do cenário principal, caso ocorra alguma inconsistência na validação como campo obrigatório, ou dados informados incorretamente, o sistema web apresenta mensagem referente ao problema encontrado.
<b>Pós-condição</b>	O <b>Usuário</b> visualiza os dados da NF-e.

Quadro 52 – Detalhamento do caso de uso UC05

O Quadro 53 detalha o caso de uso UC06, apresentado na Figura 8.

<b>UC06 – Consultar a autorização de uso da NF-e</b>	
<b>Resumo</b>	Permitir consultar o status de autorização de uso da NF-e no SEFAZ.
<b>Requisito atendido</b>	RF06.
<b>Pré-condição</b>	A NF-e deve ter sido enviada ou recebida pelo sistema web.
<b>Cenário principal</b>	<p>1 – O Usuário deve selecionar a empresa emitente;</p> <p>2 – O Usuário deve selecionar o filtro por NF-es emitidas ou recebidas;</p> <p>3 – O Usuário deve confirmar a ação de listar as NF-es;</p> <p>4 – O sistema filtra e apresenta as NF-es;</p> <p>5 – O Usuário deve selecionar a NF-e da listagem;</p> <p>6 – O Usuário deve confirmar a ação de consultar o status de autorização de uso da NF-e;</p> <p>7 – O sistema apresenta o status de autorização de uso da NF-e.</p>
<b>Exceção 01</b>	No passo 3 do cenário principal, caso ocorra alguma inconsistência na validação como campo obrigatório, ou dados informados incorretamente, o sistema web apresenta mensagem referente ao problema encontrado.
<b>Pós-condição</b>	É exibido o status de autorização de uso da NF-e de acordo com o SEFAZ.

Quadro 53 – Detalhamento do caso de uso UC06