

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ**  
**COORDENAÇÃO DE LICENCIATURA EM INFORMÁTICA**  
**DESENVOLVIMENTO DE SISTEMAS PARA INTERNET DE DISPOSITIVOS**  
**MÓVEIS**

**LUIZ FELIPE BARTZ**

**COMBUSTÍVEL AGORA: BUSCA INTEGRADA E INTELIGENTE DE VALORES**

**TRABALHO DE CONCLUSÃO DE CURSO**

**FRANCISO BELTRÃO - PR**

**2014**

**LUIZ FELIPE BARTZ**

**COMBUSTÍVEL AGORA: BUSCA INTEGRADA E INTELIGENTE DE VALORES**

Trabalho de conclusão de curso apresentada a Coordenação de Licenciatura em Informática da Universidade Tecnológica Federal do Paraná como requisito parcial para obtenção do título de “Especialista em Desenvolvimento de Sistemas para Internet e Dispositivos Móveis”.

Orientador: Prof. M. Sc. Wellton Costa

**FRANCISO BELTRÃO - PR**

**2014**



Ministério da Educação  
Universidade Tecnológica Federal do Paraná  
Campus Francisco Beltrão

Diretoria de Francisco Beltrão  
<Nome da Coordenação>  
<Nome do Curso>



---

## TERMO DE APROVAÇÃO

COMBUSTÍVEL AGORA: BUSCA INTEGRADA E INTELIGENTE DE VALORES

por

LUIZ FELIPE BARTZ

Este Trabalho de Conclusão de Curso foi apresentada em \_\_\_\_\_ de \_\_\_\_\_  
de \_\_\_\_\_ como requisito parcial para a obtenção do título de Especialista em  
Desenvolvimento de Sistemas para Internet e Dispositivos Móveis. O(a)  
candidato(a) foi arguido pela Banca Examinadora composta pelos professores  
abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho

\_\_\_\_\_.

( Aprovado / Aprovado com Restrições / Reprovado )

\_\_\_\_\_  
Wellton Costa  
Prof. Orientador

\_\_\_\_\_  
Paulo Júnior Varela  
Membro titular

\_\_\_\_\_  
Renato Hallal  
Membro titular

Dedico este trabalho a minha família, pois  
ela é minha base, meu alicerce.

## **AGRADECIMENTOS**

Agradeço a todos os amigos que incentivaram e apoiaram a conclusão de mais esta etapa e também aos que colaboraram no desenvolvimento deste trabalho.

Expresso minha gratidão aos meus colegas e companheiros de estrada Henrique Vendrame e Oneide Luiz Schneider, pois juntos superamos o cansaço, as dificuldades e concluímos mais essa fase.

Deixo registrado ainda o reconhecimento a minha família, que sem seu apoio não seria possível vencer este desafio.

## RESUMO

Considerando o grande consumo e dependência dos combustíveis pela população faz-se necessário a aquisição no melhor preço pelo consumidor e também o aumento da gama de consumidores por parte do fornecedor através de uma solução simples e confiável. O trabalho teve como objetivo a criação de um ambiente em que seja possível a troca de informação através de web service, proporcionando a integração entre as partes interessadas e gerando indiretamente o marketing digital. Assim, este ambiente é composto por três aplicações onde uma persiste e fornece os dados na plataforma web e as demais consomem as informações em plataformas distintas, sendo estas web e móvel. Para compreender e embasar a necessidade deste trabalho, na fundamentação teórica são explanados tópicos justificando as escolhas para o desenvolvimento, assim como na metodologia são apresentadas as etapas e os meios adotados. Por fim, nos resultados é evidenciado o ambiente criado.

**Palavras-chave:** Marketing. Informação. Web service. Integração. Web. Móvel.

## ABSTRACT

Seeing the large consumption and dependence on fuels by the population, it is necessary that the consumer get it at the best price and the supplier provide the increase of consumers by a simple and reliable solution. The work aimed making an environment where it is possible the exchange of information by web service, affording the integration between stakeholders and doing indirectly the digital marketing. Therefore, this environment is made of three applications wherein one persists and gives the data in the web platform and the others spend the information in different web and mobile platforms. To understand and support the need for this work, in theoretical foundation are explained topics to justify the choices for the development, as well in methodology are shown the steps and means adopted. Finally, in the results is demonstrated the environment created.

**Keywords:** Marketing. Information. Web service. Integration. Web. Mobile.

## LISTA DE ILUSTRAÇÕES

Diagrama 1: Diagrama de classes do sistema (aplicação principal) .....	31
Esquema 1: Fluxo do modelo MVC .....	22
Esquema 2: Arquitetura das aplicações .....	29
Figura 1: Exemplo JSON.....	20
Listagem 1: Estrutura MVC da aplicação - NetBeans (aplicação principal).....	32
Listagem 2: Web service da aplicação (aplicação principal) .....	33
Quadro 1: Levantamento de requisitos (aplicação principal).....	30
Tela 1: Tela de acesso (aplicação principal) .....	35
Tela 2: Tela de acesso (aplicação principal) .....	36
Tela 3: Tela de consulta das entidades (aplicação principal) .....	36
Tela 4: Tela de cadastro da entidade (aplicação principal) .....	37
Tela 5: Tela principal e menu da entidade (aplicação principal).....	37
Tela 6: Tela para manutenção de preços (aplicação principal) .....	38
Tela 7: Tela principal sincronizado informações (aplicação web).....	39
Tela 8: Tela principal com informações populadas (aplicação web) .....	40
Tela 9: Tela com resultado da pesquisa (aplicação web).....	40
Tela 10: Tela com dados da entidade (aplicação web) .....	41
Tela 11: Telas de busca (aplicação móvel) .....	42
Tela 12: Telas de resultado (aplicação móvel) .....	43

## LISTA DE SIGLAS

HTML	HyperText Markup Language
HTTP	Hyper Text Transport Protocol
IDE	Integrated Development Environment
JEE	Java Enterprise Edition
JPA	Java Persistence API
JSF	JavaServer Faces
JSON	JavaScript Object Notation
JVM	Java Virtual Machine
MVC	Model View Controller
NOSQL	Not Only SQL
ORM	Mapeamento Objeto-Relacional
REST	Representational State Transfer
SERPRO	Serviço Federal de Processamento de Dados
SGBDR	Sistema Gerenciador de Banco de Dados Relacional
SOA	Arquitetura Orientada a Serviços
SQL	Structured Query Language
UML	Unified Modeling Language
WWW	World Wide Web
XML	Extensible Markup Language

## SUMÁRIO

<b>1 INTRODUÇÃO</b> .....	<b>12</b>
1.1 PROBLEMA DE PESQUISA .....	13
1.2 HIPÓTESE .....	13
1.3 JUSTIFICATIVA .....	13
1.4 OBJETIVOS .....	14
1.4.1 Objetivo Geral .....	14
1.4.2 Objetivo Específico.....	14
1.5 METODOLOGIA.....	15
1.6 ESTRUTURA DO TABALHO .....	15
<b>2 FUNDAMENTAÇÃO TEÓRICA</b> .....	<b>16</b>
2.1 COMBUSTÍVEIS .....	16
2.2 INFORMAÇÃO .....	16
2.3 MARKETING .....	17
2.4 WEB .....	17
2.5 ARQUITERUA ORIENTADA A SERVIÇO (SOA).....	18
2.6 WEB SERVICE .....	18
2.7 REPRESENTATIONAL STATE TRANSFER (REST).....	19
2.8 JAVASCRIPT OBJECT NOTATION (JSON).....	19
2.9 JAVA .....	21
2.10 JAVA PERSISTENCE API (JPA) .....	21
2.11 MODEL VIEW CONTROLLER (MVC).....	22
2.12 FRAMEWORKS .....	23
2.13 JAVASERVER FACES (JSF).....	23
2.14 PRIMA FECES .....	24
2.15 JAVASCRIPT .....	24
2.16 JQUERY.....	25
2.17 NOT ONLY SQL (NOSQL).....	25
2.18 MONGODB .....	26
2.19 SPRING DATA .....	26
2.20 GLASSFISH .....	27
2.21 ANDROID.....	27
2.22 DEMOISELLE .....	28
<b>3 METODOLOGIA</b> .....	<b>29</b>
<b>4 APRESENTAÇÃO E DISCUSSÃO DOS RESULTADOS</b> .....	<b>35</b>

4.1 APLICAÇÃO PRINCIPAL .....	35
4.2 APLICAÇÃO WEB.....	39
4.3 APLICAÇÃO MÓVEL .....	42
<b>5 CONSIDERAÇÕES FINAIS .....</b>	<b>44</b>
<b>REFERÊNCIAS.....</b>	<b>45</b>

## 1 INTRODUÇÃO

A informação assume um papel fundamental e uma expressiva importância para o funcionamento coerente de qualquer organização. As mudanças e o avanço tecnológico criaram a necessidade de uma disponibilidade maior da informação em que fornecedores e consumidores buscam integrar suas informações de qualquer lugar.

Assim, as aplicações devem facilitar a comunicação entre si e seu funcionamento deve ser independente de dispositivos ou arquiteturas. Com isso, surge o web service (serviços web) como solução eficiente para esta situação.

A atual necessidade e importância dos combustíveis juntamente com as soluções existentes objetivaram o presente trabalho. Seu foco é a criação de um ambiente onde seja possível persistir, disponibilizar e consumir informações relativas a preços de combustíveis através de uma forma simples e confiável, garantindo o melhor preço ao consumidor e aumento nas vendas do fornecedor.

No decorrer do trabalho serão abordados princípios relativos aos combustíveis, informação e marketing. Além disso, serão tratadas as tecnologias, arquiteturas, frameworks, entre outros, optados para o desenvolvimento do proposto. Em seguida, é exposto o ambiente e suas aplicações necessárias e por fim apresentado os resultados obtidos através da aplicação destes.

## 1.1 PROBLEMA DE PESQUISA

Proporcionar um ambiente simples e confiável para troca de informações relativas a preços de combustíveis visando a aquisição do melhor preço pelo consumidor e aumento no faturamento do fornecedor.

## 1.2 HIPÓTESE

Criar uma aplicação capaz de persistir e disponibilizar informações pertinentes visando a integração entre diversas aplicações de arquiteturas distintas.

## 1.3 JUSTIFICATIVA

O ambiente proporcionado pela aplicação apresentou ser cabível devido a análise efetuada no cenário atual juntamente com as soluções disponíveis no mercado, pois o grande problema encontrado é a confiabilidade e validade das informações.

## 1.4 OBJETIVOS

A seguir são descritos os objetivos que direcionam o presente trabalho.

### 1.4.1 Objetivo Geral

Desenvolver uma ferramenta completa no módulo web responsável por controlar, persistir e disponibilizar as informações referente a preços de combustíveis através de web services, assim como implementar duas aplicações cliente consumindo as informações pertinentes.

### 1.4.2 Objetivo Específico

- Construir um ambiente integrado capaz de controlar, persistir e disponibilizar informações referentes a preços de combustíveis.
- Desenvolver uma aplicação web com auxílio de frameworks.
- Armazenar as informações relativas no banco de dados não-relacional.
- Implementar nesta aplicação um web service para comunicação com outras aplicações.
- Criar aplicações web e móvel consumindo o web service criado.

## 1.5 METODOLOGIA

Inicialmente foram identificadas as dificuldades encontradas pelas soluções disponíveis e juntamente com fundamentação teórica foi possível definir as diretrizes básicas para a criação do modelo desejado. Em seguida, iniciou-se a modelagem da aplicação principal e sua implementação. Após a conclusão da aplicação principal foi desenvolvido as demais aplicações.

## 1.6 ESTRUTURA DO TABALHO

O presente trabalho está estruturado da seguinte maneira:

No capítulo 1 apresentam-se os tópicos de introdução ao assunto, o problema, a hipótese, a justificativa, os objetivos e a metodologia utilizada.

No capítulo 2 abordam-se os conceitos, definições, arquiteturas, frameworks, entre outros através da fundamentação teórica servindo de base para a execução do trabalho.

No capítulo 3 trata-se dos meios e processos adotados para implementação das aplicações necessárias.

No capítulo 4 são demonstrados os resultados obtidos através das escolhas adotadas.

Ao final, o capítulo 5 relata as considerações finais, análise dos resultados e as conclusões.

## 2 FUNDAMENTAÇÃO TEÓRICA

A fundamentação teórica a seguir irá inicialmente apresentar a atual importância dos combustíveis, a necessidade da informação, assim como a relevância do marketing para as organizações.

Serão abordados também tópicos com conceitos, padrões, arquiteturas, modelos, especificações, entre outros essenciais para o decorrer do projeto.

### 2.1 COMBUSTÍVEIS

A necessidade do ser humano de dominar a energia trouxe consigo o avanço tecnológico. O homem primitivo iniciou utilizando a energia térmica e luminosa do fogo, escolhendo o que o mantinha aceso. Este processo chamamos hoje de combustível sendo a principal fonte de energia e que foi substituído com o passar do tempo, evoluindo de madeira à derivados do petróleo.(DIONYSIO; MEIRELLES, 2014)

Segundo Brasil (2014), a comercialização de combustíveis no país em 2013 totalizou 136.242 bilhões de litros, apresentando um aumento de 5,0% em relação ao ano anterior. Com isso, observa-se a grande demanda e importância juntamente com a real necessidade dos mesmos.

### 2.2 INFORMAÇÃO

Na tomada de decisões a informação é essencial, trazendo consigo o conceito de fazer conhecer e tornar público. Assim, Carvalho (2000) define a informação como um conjunto de dados, que se fornecido no tempo e na forma apropriada, aperfeiçoa o conhecimento e torna possível a execução de determinada tarefa ou auxilia na melhor decisão.

Assim sendo, os sistemas de informação possuem informações relevantes para organizações ou seu ambiente. Agem como mecanismos de apoio à gestão, baseados em tecnologia da informação, atuando como condutores das informações para facilitar, otimizar e agilizar o processo decisório.

## 2.3 MARKETING

Para Kotler (1998), o marketing vem sendo indispensável no mundo cada dia mais globalizado e competitivo onde empresas disputam e exploram pequenos nichos de mercado. Destacou também a transformação radical na economia nos últimos 30 anos, que com o aparecimento da internet e outros fatores permitiu as empresas ampliar seus mercados.

Marques (2008) ressalta que marketing é a arte de projetar e operacionalizar o conhecimento, estabelecer preços, promoção e multiplicação de idéias, produtos e serviços, objetivando a realização de trocas, que consistem no alcance de metas individuais e organizacionais.

## 2.4 WEB

Popularizada como Web, a WWW (World Wide Web), rede de alcance mundial, conforme Comitê (2010) é o sistema de maior sucesso na internet que possui uma tecnologia aberta e gratuita, o qual proporciona oportunidades para as empresas criarem um canal de comunicação com seus clientes, assim como pessoas interagirem entre si de qualquer lugar do mundo.

O autor ainda explana que a Web trata-se de um grande conjunto universal de aplicações, dados, documentos, páginas e serviços interligados através da rede mundial de computadores, que está disponível às pessoas de qualquer lugar do globo.

Já Conellem (2003) explica que o termo Web surgiu da visão do sistema como um bloco de nós com interconexões através de links. Assim, um sistema web é também um sistema de hipermídia devido a seus recursos estarem vinculados.

## 2.5 ARQUITETURA ORIENTADA A SERVIÇO (SOA)

A Arquitetura Orientada a Serviços (SOA) segundo Fraser, Rankine e Woodcock (2007) é um grupo de utilidades bem determinadas em forma de serviço disponibilizadas na rede. Também, determina a implementação de componentes como serviços modulares permitindo a descoberta e utilização dos mesmos, bem como fornece um repositório para publicação e descoberta dos serviços.

A utilização de uma Arquitetura Orientada a Serviços proporciona a adaptação entre sistemas de um modo fácil, comportando a criação de sistemas altamente dinâmicos conforme os serviços forem substituídos ou melhorado. Assim como promove o reuso dos serviços permitindo que estes sejam compostos, criando uma lógica de processo bem elaborada evitando o desperdício de recursos. Portanto, devido ao baixo acoplamento da arquitetura surgem diversas vantagens como: escalabilidade, flexibilidade e tolerância a falhas. (Josuttis, 2007).

## 2.6 WEB SERVICE

Web service para Kalin (2009) é um modelo de aplicação web distribuída caracteristicamente oferecida através de HTTP (Hyper Text Transport Protocol), onde os componentes podem ser aplicados e executados em dispositivos distintos.

Segundo Treinamentos (2013) web service é um serviço disponibilizado por um sistema que permite ser acessado diretamente por outros sistemas desenvolvidos em qualquer tecnologia por meio de uma rede como a internet.

Ademais, os web services e seus clientes podem operar mesmo que escritos em linguagens de programação distintas, como C/C++, Java, Perl, Python, Ruby e outras que oferecem bibliotecas, frameworks ou utilitários com suporte a esta solução. (KALIN, 2009).

## 2.7 REPRESENTATIONAL STATE TRANSFER (REST)

A arquitetura Representational State Transfer (REST) para web services surgiu como uma forma de facilitar o desenvolvimento destes serviços, não se tratando de um padrão, mas sim de um estilo arquitetural para criação de sistemas distribuídos, definido por Roy Thomas Fielding (FIELDING, 2000).

Silveira et al. (2011) explanam que grandes empresas vem adotando o REST devido a facilidade para integrar sistemas, assim como a praticidade para evoluir o mesmo. Pois atualmente os sistemas que não disponibilizam pontos para integração estão predestinados a grande esforço e manutenção.

Portanto, Richardson (2007) conclui que na medida que o REST vem se popularizando surgem novos frameworks para facilitar esta implementação. Estes por sua vez adquirem formas e características REST causando assim mais interesse nesta arquitetura.

## 2.8 JAVASCRIPT OBJECT NOTATION (JSON)

JavaScript Object Notation (JSON) trata-se de um formato leve para troca de dados sendo fácil para as pessoas lerem e escreverem, assim como máquinas analisarem e gerarem. É baseado em um subconjunto da linguagem de programação JavaScript, porém seu formato é independente da linguagem. Assim, estas características tornam o formato JSON ideal para troca de dados. (JSON, 2014).

Richardson (2007) observou uma crescente utilização em estruturas simples de dados como JSON, fugindo do padrão de documentos no formato XML (Extensible Markup Language). Tendo em vista que a maioria dos navegadores suportam estruturas de dados baseados em JavaScript, não estando amarrados a qualquer linguagem ou plataforma, tratando-se apenas de uma alternativa.

```
1 {
2   "nome": "Luiz Felipe",
3   "sobrenome": "Bartz",
4   "age": 23,
5   "endereco": {
6     "logradouro": "Av. Sul Brasil",
7     "cidade": "Maravilha",
8     "estado": "SC",
9     "cep": "89874-000"
10  },
11  "telefone": [
12    {
13      "tipo": "celular",
14      "numero": "049 9999-8888"
15    },
16    {
17      "tipo": "casa",
18      "numero": "049 3333-4444"
19    }
20  ]
21 }
```

Figura 1: Exemplo JSON  
Fonte: O autor (2014).

No JSON os valores são representados por nome (string) e valor, sendo válidos quatro tipos primitivos entre eles: string, número, booleano e nulo; e dois tipos estruturados: objeto e array. Para string utiliza-se uma seqüência de zero ou muitos caracteres Unicode, já o objeto é uma coleção não ordenada de nome/valor e o array trata-se de uma seqüência ordenada de zero ou muitos valores. (CROCKFORD, 2006).

## 2.9 JAVA

A linguagem de programação Java surgiu em maio de 1995, quando supostamente iria trabalhar com sites para web. A aceitação e o crescimento rápido da linguagem deram-se devido à ascensão da web na época, assim como a possibilidade de utilizar recursos que não eram possíveis até então. Atualmente, devido ao amadurecimento e as mudanças ocorridas no mundo, a utilização da linguagem é vista em aplicativos desktop, páginas da internet ou em celulares permitindo a portabilidade entre sistemas operacionais. (GONÇALVES, 2007).

Um dos principais benefícios do Java segundo Flanagan (2005), é a possibilidade de que o código escrito uma vez seja executado em qualquer lugar, seguindo assim sua promessa. Pois está integrado na maioria dos sistemas operacionais, incorporado nos mais populares navegadores web e em vários dispositivos eletrônicos como televisão e celulares.

Ainda, o autor explica que a plataforma é formada por um grupo predefinido de classes existentes em cada instalação do Java em que existe um interpretador chamado de JVM (Java Virtual Machine), que é responsável por interpretar e executar os bytecodes. Também, trata-se de uma linguagem de programação orientada a objetos, lançada pela Sun Microsystems e com sintaxe semelhante ao C que despertou o interesse de vários programadores.

## 2.10 JAVA PERSISTENCE API (JPA)

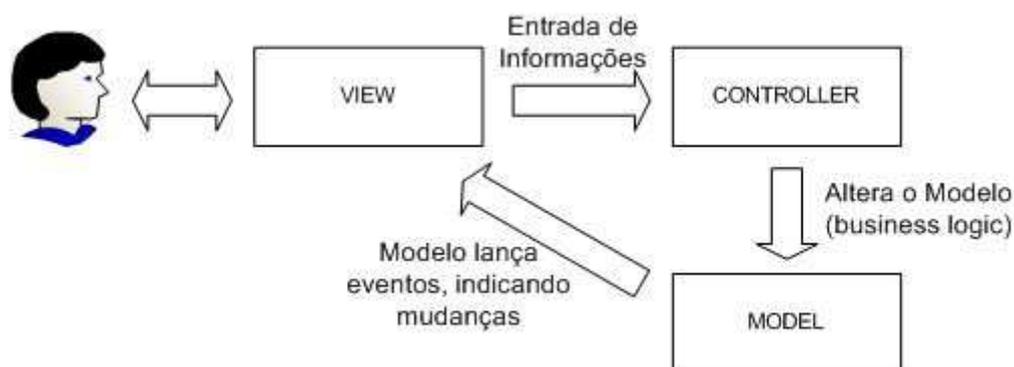
O Java Persistence API (JPA) é uma especificação elaborada pela comunidade Java, que segundo Keith e Schincariol (2006) oferece um padrão simples e poderoso para o Mapeamento Objeto-Relacional (ORM) .

Trata-se de uma interface padrão para anotar, recuperar e persistir objetos Java em vários bancos de dados relacionais sem a necessidade de códigos específicos para os mesmos, facilitando assim a portabilidade entre bancos de dados. (DEVELOPERS, 2014).

Em consonância, Keith e Schincariol (2009) explicam que trabalhando desta forma é possível o emprego de qualquer framework de persistência, que siga os padrões JPA de um meio desacoplado, facilitando se necessário a troca sem grandes danos na aplicação.

## 2.11 MODEL VIEW CONTROLLER (MVC)

A arquitetura Model View Controller (MVC) é um padrão de projeto, que objetiva a criação e separação de três partes distintas sendo elas: modelo, visão e controlador. O modelo é o meio de acesso a persistência de dados, a visão está relacionada à exibição e entrada dos dados, já o controlador se responsabiliza por tratar os dados recebidos da visão e aplicar as regras de negócio. Estes dados poderão ser persistidos, processados e após isso podem retornar ao usuário. (GONÇALVES, 2007).



Esquema 1: Fluxo do modelo MVC  
Fonte: Almeida (2011).

Conforme Marcoratti (2002), a proposta de divisão em camadas lógicas permite que aplicação seja mais flexível, onde cada camada pode sofrer alterações de forma independente. Da mesma forma, permite a separação em pacotes da lógica de negócio, reduzindo a dependência entre as partes distintas, fornecendo uma forma de trabalho autônoma e facilitando o reuso dos pacotes em diversos sistemas. Indicando assim um modelo padrão em sistemas web.

## 2.12 FRAMEWORKS

Framework é um conjunto de artefatos de software utilizado por várias aplicações distintas. Sendo geralmente classes, que juntamente ao software tornam obrigatório o uso das mesmas. (BRAUDE, 2005).

Os frameworks são criados para sanar a necessidade de reuso do código e o desenvolvimento ágil. Trazem consigo códigos, classes e padrões de projeto, que proporcionam grande reutilização e disponibilizam mecanismos para que o desenvolvedor estenda ou conecte-se em diversas funções. (LARMAN, 2004).

Portanto, Braude (2005) conclui que o maior objetivo no desenvolvimento de software é a reuso, pois caso não seja possível uma organização se beneficiar dos investimentos e habilidades dos colaboradores, necessitando várias vezes o trabalho dos mesmos, seus concorrentes se antecipam e tornam isso possível em seus produtos, tendo resultados superiores.

## 2.13 JAVASERVER FACES (JSF)

O JavaServer Faces (JSF) é um framework MVC baseado em Java que, segundo Bruns e Schalk (2010), auxilia, facilita e simplifica o desenvolvimento da interface do usuário, sendo uma das partes mais custosas para o desenvolvimento de aplicações web. Além disso, é o framework oficial da Sun Microsystems criado para simplificar o desenvolvimento de aplicação web. (NETO, 2011).

Este por sua vez foi elaborado para tornar simples o desenvolvimento de aplicações web que separam os componentes de interface do usuário com os componentes da lógica de negócio e que indiretamente estão interligados. A utilização deste é recomendada pelo seu fabricante para desenvolvimento web em Java. (GONÇALVES, 2008).

## 2.14 PRIMAFECES

O PrimeFaces é uma biblioteca com um grande conjunto de componentes responsáveis pela interface com o usuário para aplicações web baseadas em JSF. Sua utilização é simples, pois em um único arquivo disponibiliza todos os recursos sem necessidade de configurações e dependências a outras bibliotecas. (JONNA, 2014).

Assim, para Hlavats (2013), o PrimeFaces é a melhor tecnologia atualmente no desenvolvimento de aplicações web baseadas no JSF, tornando a aplicação funcional, poderosa e rica.

## 2.15 JAVASCRIPT

JavaScript é uma linguagem de programação web que assume o papel de especificar o comportamento das páginas, a qual é utilizada pela grande maioria dos sites modernos. Sendo que todos os navegadores atuais presentes em computadores, tablets e smartphones apresentam interpretador da linguagem, tornando ela a mais onipresente da história. (FLANAGAN, 2011).

Também, Crockford (2008) elege o JavaScript como uma importante linguagem de programação, pois é a linguagem dos navegadores web. Essa associação a torna a mais popular linguagem de programação.

## 2.16 JQUERY

O jQuery é uma biblioteca JavaScript focada em consultas eficientes nos elementos de documentos para assim poder manipulá-los. Estas consultas permitem retornar objetos com uma diversidade de métodos úteis para operar sobre os respectivos elementos. (FLANAGAN, 2010)

Segundo Rutter (2011), a biblioteca foi criada para auxiliar desenvolvedores a escrever e estender interações JavaScript de uma forma concisa e rápida, utilizando um conjunto de métodos em torno das funções nativas do JavaScript.

## 2.17 NOT ONLY SQL (NOSQL)

O termo "Not only SQL" (NoSQL) é uma combinação entre duas palavras: Não e SQL (Structured Query Language), sendo uma tecnologia ou produto que combate o SQL que no início era chamado de "Não SGBDR" (Sistema Gerenciador de Banco de Dados Relacional) ou "Não relacional", mas assumiu o "NoSQL" devido a sonoridade agradável. Atualmente é utilizado para especificar os bancos de dados que não seguem os princípios do SGBDR, trabalham com grande volume de dados ou são acessados e manipulados na escala da web. Assim, NoSQL não é um único produto ou tecnologia, mas sim uma classe de produtos e uma coleção de vários conceitos sobre armazenamento e manipulação de dados, onde muitos deles estão relacionados. (TIWARI, 2011).

Dayley (2014a) conclui que a motivação por trás do NoSQL é o design simplificado, escala horizontal e um melhor controle sobre a disponibilidade dos dados. Esse modelo de banco de dados é especialista em tipos de dados que necessitam de um melhor desempenho dos que os tradicionais SGBDR disponibilizam. Seu intuito é romper a tradicional estrutura dos bancos relacionais, onde os programadores podem implementar modelos e formas que se enquadram com sua necessidade, o que até então não era possível.

## 2.18 MONGODB

O MongoDB é um banco de dados orientado a documentos, conceituado assim como NoSQL, que possui seu código aberto e uma grande performance. Segue as premissas de ser ágil e altamente escalável, diferenciado dos demais por não possuir esquemas e armazenar os dados em coleções idênticas ao JSON usando pares de chave/valor. (MONGODB, 2014).

Para Dayley (2014b) o MongoDB é muito ágil, escalável e fácil de implementar. Esse banco de dados permite o armazenamento de um grande volume de dados para aplicações web e serviços que necessitam de alto tráfego de informações. Ainda, Dayley (2014a) conclui que o MongoDB é o mais popular e com maior suporte ao NoSQL disponível atualmente.

## 2.19 SPRING DATA

O Spring Data é um framework para acessar dados que dissocia o código através de mecanismo de acesso. Sua forma de trabalhar permite que o desenvolvedor concentre-se exclusivamente na lógica de negócios tornando seu trabalho mais fácil. (KONDA, 2012).

Desde seu surgimento, Brisbin et al. (2012) destacam que o Spring Data tem prestado suporte sofisticado para as tecnologias tradicionais de acesso a dados. Este suporte teve como alvo os bancos de dados relacionais, sendo marcado como a ferramenta predominante no mercado quando se tratava de persistência de dados. Com o surgimento do NoSQL, a tecnologia evoluiu fornecendo também aos desenvolvedores uma solução para este tipo de estrutura.

Portanto, Kainulainen (2012) conclui que o objetivo do Spring Data é fornecer uma maneira fácil de criar aplicações que utilizam bancos de dados relacionais e não relacionais, assim como outras diversas tecnologias de acesso a dados.

## 2.20 GLASSFISH

O GlassFish segundo Pelegri-Ilopart, Yoshida e Moussine-pouchkine (2007), é um servidor de aplicações baseado na Plataforma Java para aplicações e serviços web. A constante utilização deste é caracterizada pela confiabilidade, desempenho, facilidade e produtividade. Também, possuiu seu código aberto (open source) e está disponível gratuitamente.

Logo, Heffelfinger (2014) explana que o GlassFish disponibiliza as bibliotecas necessárias para desenvolver e implementar aplicativos Java seguindo as especificações do JEE (Java Enterprise Edition), que consiste em praticamente uma receita detalhada para a implementação do software.

## 2.21 ANDROID

O Android é uma plataforma totalmente aberta e completa para tecnologia móvel, pois possui um pacote de programas, middleware, aplicativos e interface do usuário juntamente com um sistema operacional. Foi construído para permitir que os desenvolvedores criem aplicações tirando o maior proveito que o hardware possa fornecer. (ABLESON et al., 2011)

Pereira e Silva (2012) ressaltam que o Android é um ambiente de software para dispositivos móveis e não uma plataforma de hardware que possui um sistema operacional baseado em um kernel Linux. Esta plataforma de software vem revolucionando o mercado global de aplicativos para celulares, pois seu código fonte aberto despertou o interesse do mercado de telefonia mundial.

## 2.22 DEMOISELLE

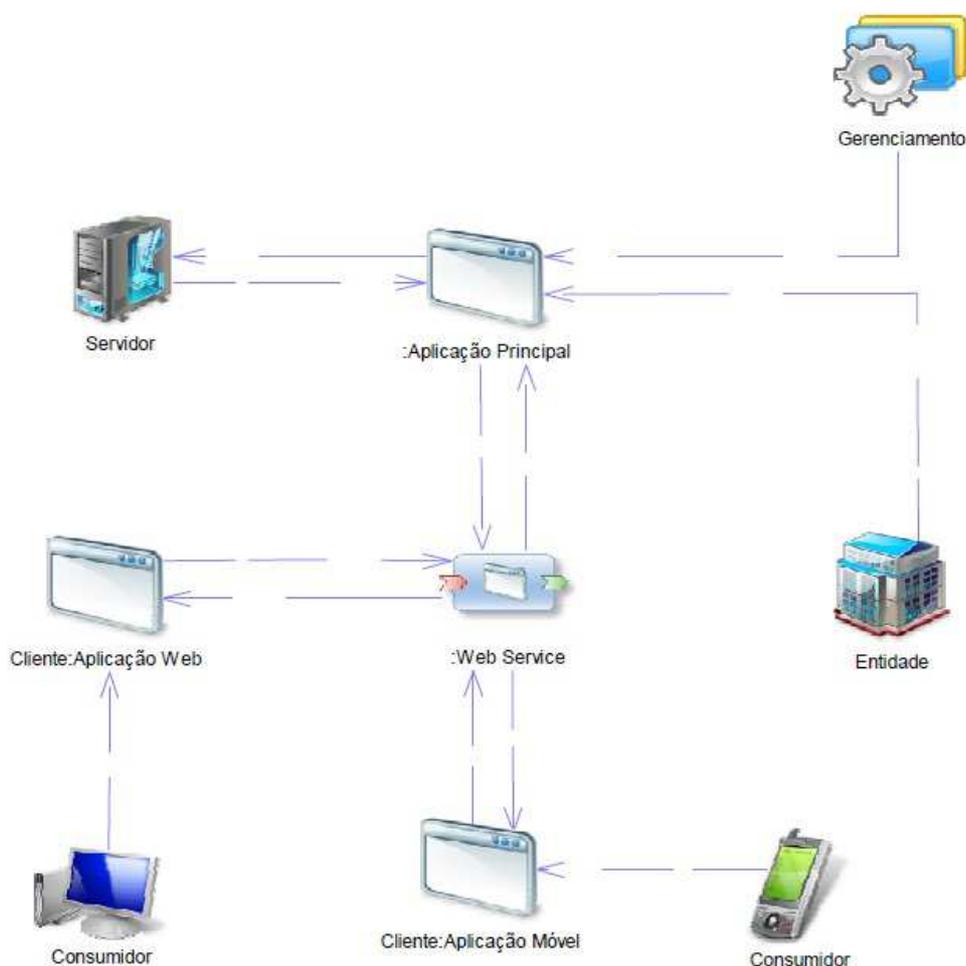
O Governo Federal através da SERPRO (Serviço Federal de Processamento de Dados) criou o framework Demoiselle que segundo Sacramento et al. (2013), objetiva facilitar a criação de aplicações, adota e integra vários frameworks especialistas que minimiza assim o tempo de escolha/integração entre eles e também fornece uma série de mecanismos reusáveis para as mais comuns funcionalidades de uma aplicação. Este, por sua vez, está disponível desde 2009 como software livre.

Além disso, a implementação do framework baseia-se em especificações definidas pela comunidade Java, sendo estes aceitos mundialmente. O que possibilita ao desenvolvedor estar desacoplado de produtos específicos. (LISBOA, 2010).

Ainda, Lisboa (2009) afirma que o Demoiselle é basicamente uma biblioteca de módulos que atende as necessidades de infra-estrutura básica para uma aplicação web não distribuída. Assim, a adoção deste framework pretende automatizar e acelerar a integração de sistemas, eliminar o retrabalho e aumentar a produtividade.

### 3 METODOLOGIA

O trabalho teve como objetivo criar três aplicações distintas para consulta de preços de combustíveis. Inicialmente, realizou-se um estudo sobre os problemas enfrentados pelas atuais soluções do mercado. Em seguida, estudaram-se os meios necessários para resolver o problema juntamente com imagens de programação, arquitetura, frameworks e padrões envolvidos. Após isso, foi elaborado o Esquema 1, objetivando a compreensão da arquitetura necessária.



Esquema 2: Arquitetura das aplicações  
Fonte: O autor (2014).

Para criação das aplicações propostas e integrá-las foi necessário que uma das partes efetuasse o controle, a persistência e a disponibilidade dos dados . Esta por sua vez foi denominada "Aplicação Principal" e o seu funcionamento de forma correta é essencial. Assim, para auxiliar posteriormente no desenvolvimento desta foi elaborado o levantamento de requisitos garantindo sua funcionalidade. Segue abaixo parcial do documento.

	Title ID	Full Description	Code	Priority	Workl	Risk	Status
1	1.	<b>Município</b> Cadastro de Município.	REQ_0001	Undefined	0	Undefined	Draft
9	2.	<b>Produto</b> Cadastro de Produto, informação sobre quais tipos de Produto (Gasolina, Gasolina Comun,...) estão disponíveis para informação/consulta de Preço.	REQ_0027	Undefined	0	Undefined	Draft
18	3.	<b>Preço</b> Cadastro/Atualização dos Preços dos Produtos disponíveis/ utilizados.	REQ_0033	Undefined	0	Undefined	Draft
19	3.1	<b>Informação</b>	REQ_0060	Undefined	0	Undefined	Draft
20	3.1.1	<b>Produto</b> Referência ao Produto a ser utilizado.	REQ_0062	Undefined		Undefined	Draft
21	3.1.2	<b>Valor</b> Valor do Produto.	REQ_0063	Undefined		Undefined	Draft
22	3.1.3	<b>Atualização</b> Data da última atualização do valor.	REQ_0064	Undefined		Undefined	Draft
23	3.2	<b>Validação</b>	REQ_0061	Undefined	0	Undefined	Draft
27	3.3	<b>Funcionalidade</b>	REQ_0067	Undefined	0	Undefined	Draft
	3.3.1	<b>Consulta de Preço</b> Consulta Web Service dos preços disponíveis conforme os parametros da busca (Produto/Município/Lugradou Bairro).	REQ_0068	Undefined		Undefined	Draft
29	4.	<b>Bandeira</b> Cadastro de Bandeira do Combustivel	REQ_0047	Undefined	0	Undefined	Draft
37	5.	<b>Usuario</b> Cadastro de Usuário para efetuar Login no Sistema Web.	REQ_0042	Undefined	0	Undefined	Draft
52	6.	<b>Configuração</b> Cadastro das configurações do Sistema.	REQ_0043	Undefined	0	Undefined	Draft
61	7.	<b>Entidade</b> Cadatro de Entidade.	REQ_0044	Undefined	0	Undefined	Draft

Quadro 1: Levantamento de requisitos (aplicação principal)

Fonte: O autor (2014).

Através dos requisitos observados foi possível criar o diagrama de classes (Diagrama 1), sendo este um padrão da UML (Unified Modeling Language) que é bastante utilizado no desenvolvimento de sistemas orientado a objetos. Através dele é possível criar as classes que compõe o sistema sem maior esforço no desenvolvimento.

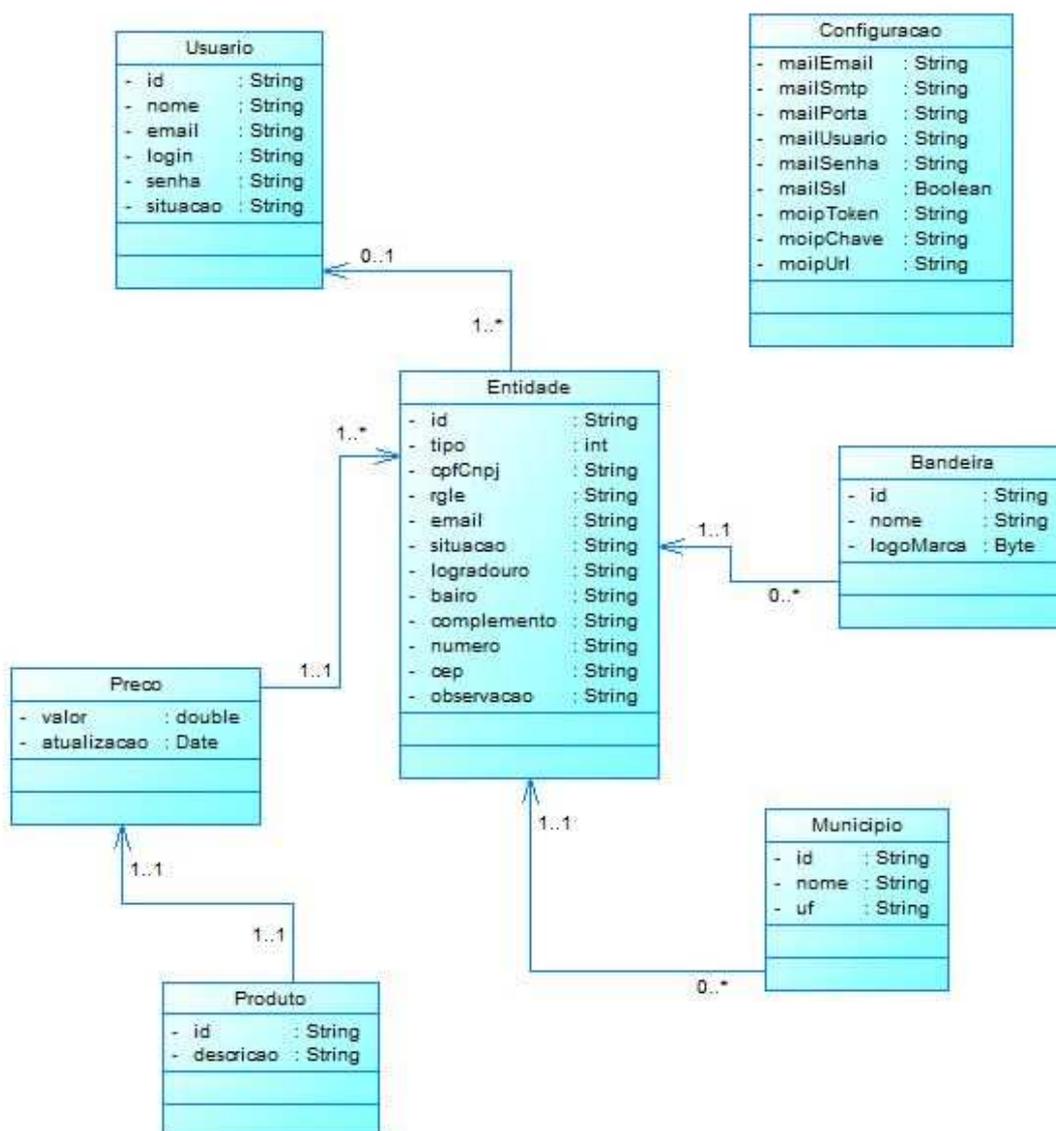
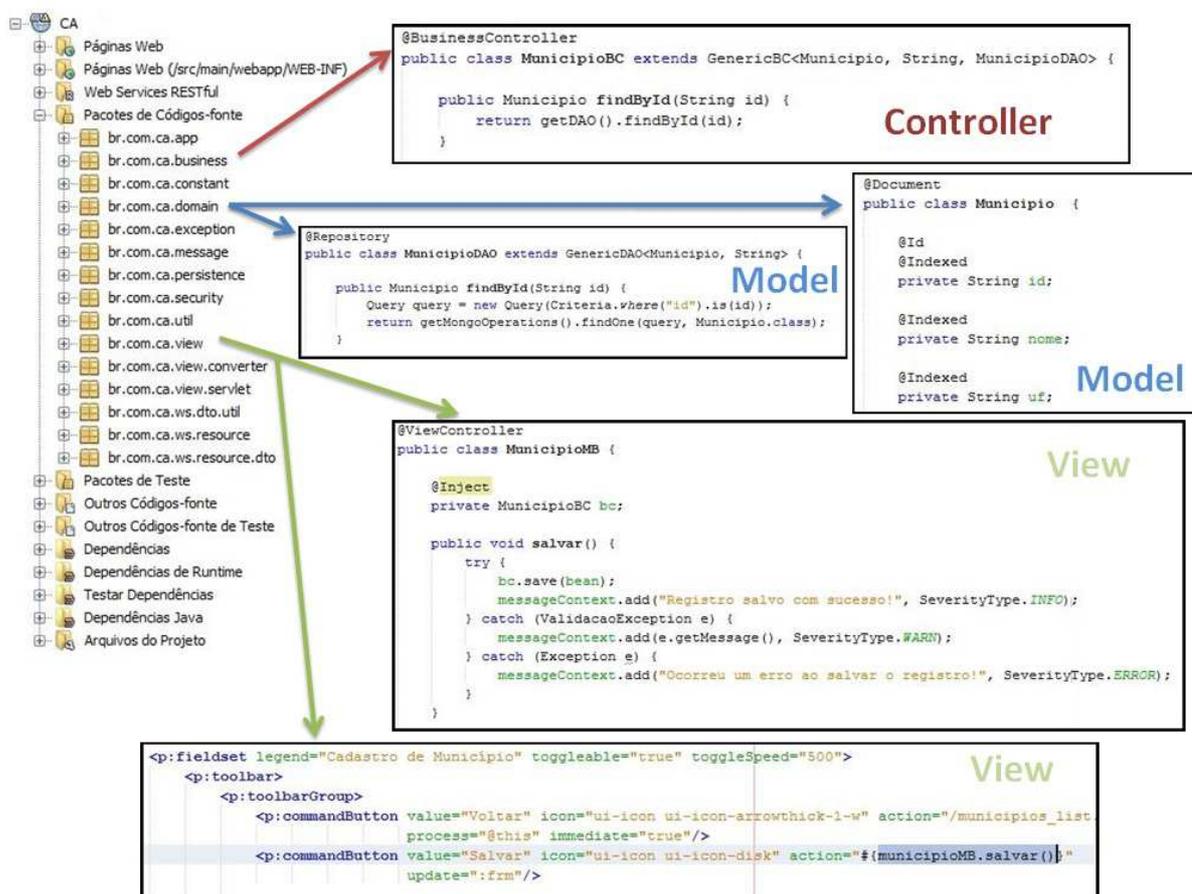


Diagrama 1: Diagrama de classes do sistema (aplicação principal)

Fonte: O autor (2014).

Esta aplicação foi desenvolvida na linguagem de programação Java utilizando a IDE (Integrated Development Environment) NetBeans através do framework Demoiselle 2.3 e seus frameworks acoplados junto a ele como o caso do JSF. Este foi responsável pela criação da interface visual com o usuário juntamente com a biblioteca de componente Primefaces .

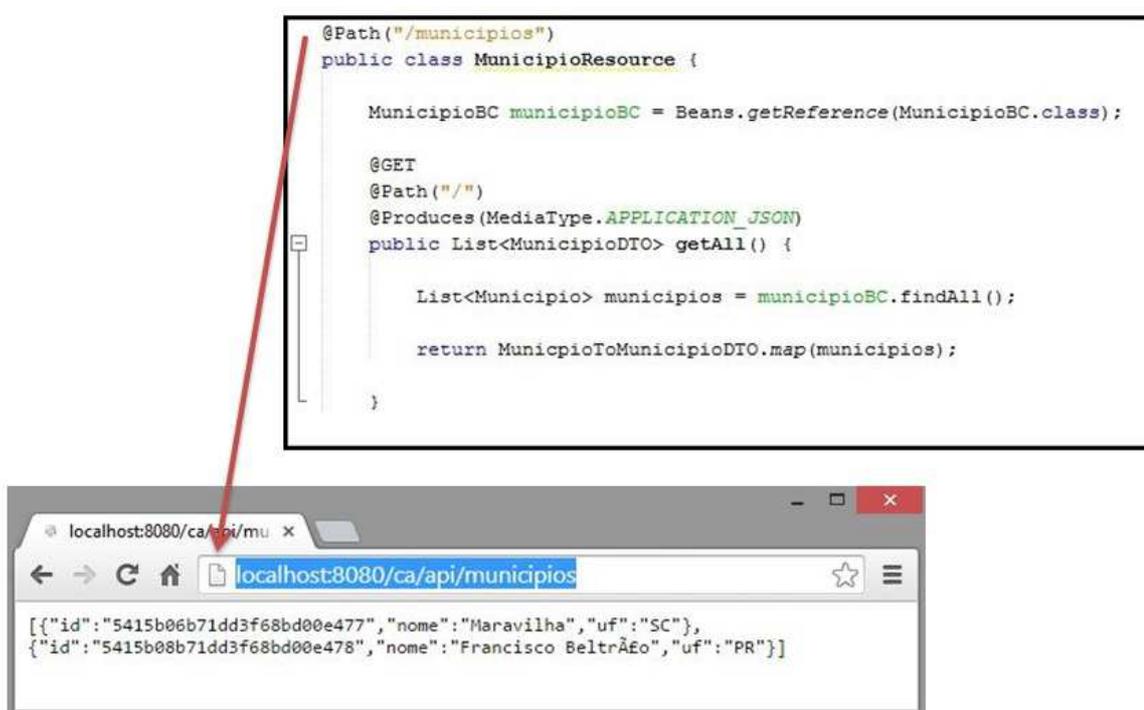
A organização do projeto na estrutura de arquivos e códigos fontes foi possível através da estrutura MVC pré-definida no framework Demoiselle. Com isso, o desenvolvimento foi facilitado, assim como em manutenções futuras. A Listagem 1 apresenta a estrutura e organização da aplicação exemplificando o MVC juntamente com trechos de códigos fonte.



Listagem 1: Estrutura MVC da aplicação - NetBeans (aplicação principal)  
Fonte: O autor (2014).

Para a persistência dos dados foi utilizado o padrão JPA, também embutido ao Demoiselle juntamente ao SpringData onde as informações foram armazenadas no MongoDB, sendo este definido como NoSQL.

A disponibilidade dos dados e a integração das demais aplicações foi possível através de web services com arquitetura REST adotando o formato JSON para troca de informações. Abaixo (Listagem 2) é apresentado um dos serviços disponibilizados pela aplicação com a codificação e os dados resultantes.



Listagem 2: Web service da aplicação (aplicação principal)  
Fonte: O autor (2014)

Por fim, para disponibilizar a aplicação foi necessário que a mesma estivesse implantada em um servidor Java. Então, foi adotado o GlassFish devido a facilidade, desempenho e a crescente utilização do mesmo.

O desenvolvimento das aplicações restantes somente foi possível após a conclusão da primeira parte descrita, pois ela disponibiliza as informações pertinentes e determina a forma de trabalho de uma maneira única através de um só canal de comunicação, também conhecido como API (Interface de Programação de Aplicações). Estas aplicações clientes foram desenvolvidas em arquiteturas diferentes.

A Aplicação Web trata-se de uma página web simples, ou seja, escrito apenas com HTML (HyperText Markup Language), JavaScript juntamente da biblioteca jQuery e com auxílio da IDE NetBens. Já a segunda, a aplicação móvel, visa um aplicativo inteligível para os dispositivos móveis com sistema operacional Android a partir da versão 1.5 (Cupcake) codificado através da IDE Eclipse também na linguagem de programação Java.

## 4 APRESENTAÇÃO E DISCUSSÃO DOS RESULTADOS

Abaixo será demonstrado os resultados obtidos através da implementação das aplicações propostas juntamente com a revisão literária apresentada.

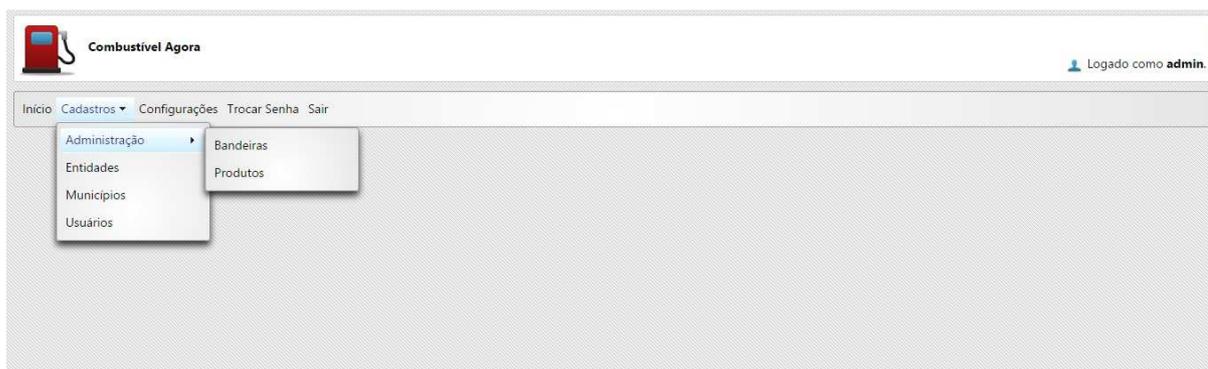
### 4.1 APLICAÇÃO PRINCIPAL

A aplicação principal foi assim denominada devido ao papel que a mesma exerce perante às demais, pois além de persistir e disponibilizar os dados também é responsável pela separação e controle de acesso em dois ambientes, o administrativo e da entidade. Para acessar a aplicação em ambos os ambientes é necessário estar autenticado por meio da tela de acesso ao sistema conforme Tela 1, onde após efetuá-la é automaticamente disponibilizado as funcionalidades pertinentes.



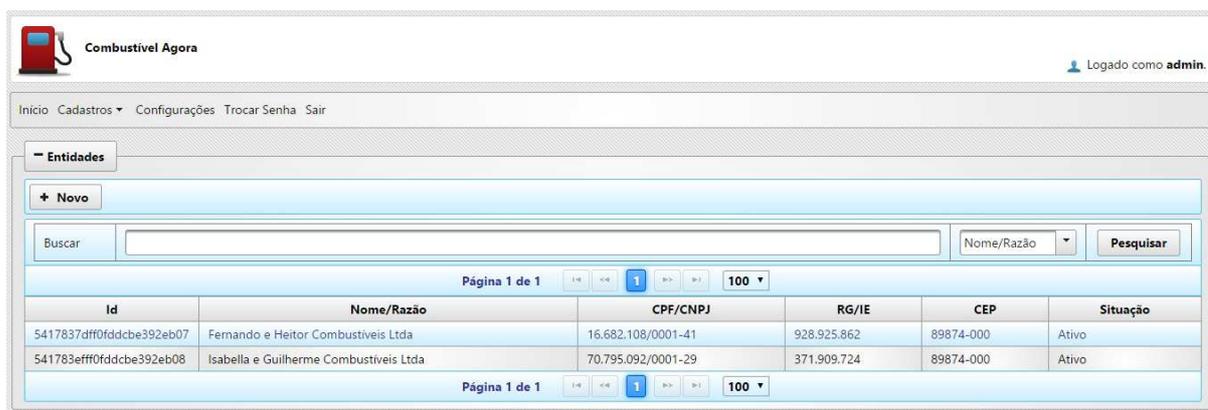
Tela 1: Tela de acesso (aplicação principal)  
Fonte: O Autor (2014).

No ambiente administrativo é possível gerenciar os cadastros básicos para o ideal funcionamento. Nestes estão inclusos conforme Tela 2, o cadastro de produtos (combustíveis) e municípios que são disponibilizados para integração da consulta nas demais aplicações. Também, é possível efetuar o cadastro de bandeiras (marca da distribuidora), entidades e demais usuários administradores.



Tela 2: Tela de acesso (aplicação principal)  
Fonte: O Autor (2014).

Ao acessar o módulo de entidades é disponibilizado uma listagem dos cadastros existentes, sendo possível efetuar consulta através de um campo específico para uma manutenção do mesmo ou efetuar um novo cadastro conforme Tela 3.



Tela 3: Tela de consulta das entidades (aplicação principal)  
Fonte: O Autor (2014).

O cadastro e manutenção das informações pertinentes a entidade é efetuado através da Tela 4. Nesta, é efetuado a validação e obrigatoriedade de campos, manutenção da lista de usuário vinculados a entidade para acesso ao sistema, assim como a referência a cadastros previamente efetuados como bandeira e município, sendo essenciais para o desenvolver da busca futura dos dados.

The screenshot shows the 'Cadastro de Entidades' form in the 'Combustível Agora' application. The form is titled 'Cadastro de Entidades' and has a navigation bar with buttons for 'Voltar', 'Novo', 'Salvar', and 'Deletar'. The form fields are as follows:

Id	5417837dff0fddcbe392eb07
Situação	Ativo
Tipo	Juridica
* Razão Social	Fernando e Heitor Combustíveis Ltda
Nome Fantasia	FH Combustíveis
* CNPJ	16.682.108/0001-41
* IE	928.925.862
* E-mail	fh@ww.com.br
* Telefone	(99) 9999-9999
Celular	
Observação	

The user is logged in as 'admin'.

Tela 4: Tela de cadastro da entidade (aplicação principal)  
Fonte: O Autor (2014).

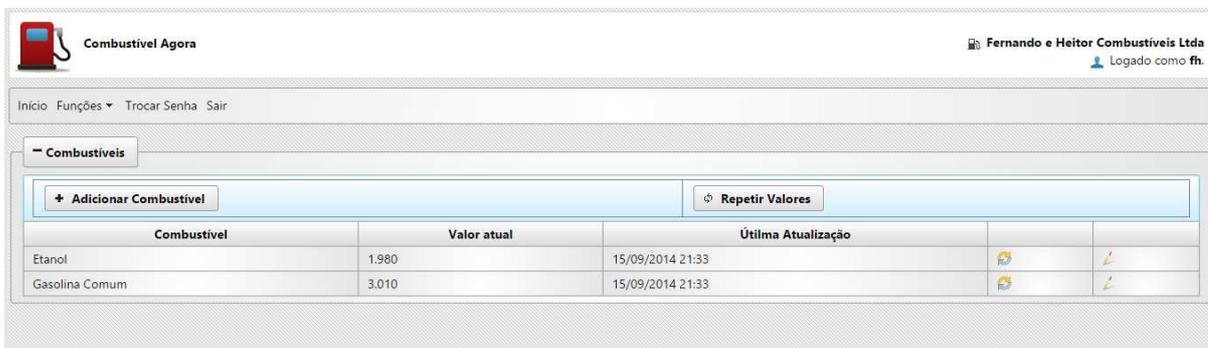
Abaixo (Tela 5), é visualizado o ambiente principal disponibilizado para entidade juntamente com suas respectivas funcionalidades. Assim, é possível efetuar a manutenção nos usuários associados a seu cadastro para acesso ao sistema. Também, é possível efetuar a manutenção dos dados cadastrais efetuado antecipadamente pelo administrador e a manutenção nos preços de combustíveis, onde futuramente estas informações serão concedidas para a consulta entre as outras aplicações.

The screenshot shows the main interface of the 'Combustível Agora' application. The user is logged in as 'Fernando e Heitor Combustíveis Ltda' with the role 'fh'. The navigation bar includes 'Início', 'Funções', 'Trocar Senha', and 'Sair'. A dropdown menu is open under 'Funções', showing the following options:

- Dados Cadastro
- Combustíveis
- Usuarios

Tela 5: Tela principal e menu da entidade (aplicação principal)  
Fonte: O Autor (2014).

A manutenção no preço dos combustíveis é visualizado na Tela 6, sendo permitido a inclusão de preços somente para combustíveis (produtos) anteriormente cadastrados pelo administrador, assim como a atualização no preço dos mesmos após adicionado.



Combustível	Valor atual	Última Atualização		
Etanol	1.980	15/09/2014 21:33		
Gasolina Comum	3.010	15/09/2014 21:33		

Tela 6: Tela para manutenção de preços (aplicação principal)  
Fonte: O Autor (2014).

Desta forma, com as informações necessárias é possível que os métodos do web service forneçam os dados fundamentais para realização da busca, assim como o resultado desejado.

## 4.2 APLICAÇÃO WEB

Com a aplicação web é possível efetuar a consulta dos preços de um determinado combustível em uma determinada cidade, sendo estas informações obrigatórias para a consulta e retornadas pelo web service, onde foram previamente cadastradas na aplicação principal através do administrador. Abaixo (Tela 7) é visualizado a sincronização destes dados ao acessar inicialmente a página.



Combustível Agora

Carregando...

**Busca**

Selecione o combustível

Selecione o município

Digite o logradouro

Digite o bairro

**Pesquisar**

Tela 7: Tela principal sincronizado informações (aplicação web)

Fonte: O Autor (2014).

Após sincronizados os dados, estes são populados nos seus respectivos campos, sendo possível a seleção conveniente conforme Tela 8. Também, é possível efetuar uma busca aproximada ao local desejado, informando o bairro ou logradouro, não sendo estes obrigatórios.

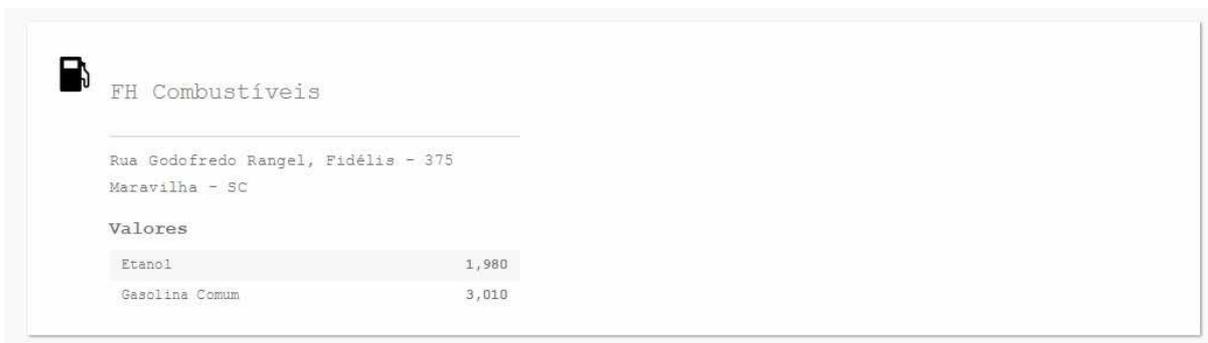
Tela 8: Tela principal com informações populadas (aplicação web)  
Fonte: O Autor (2014).

Ao efetuar a pesquisa, clicando no botão “Pesquisar”, os parâmetros definidos são repassados para o web service, onde este processa e retorna uma lista de informações que se enquadram a elas. Esta lista é processada, populada e apresentada ao usuário conforme Tela 9.

RESULTADOS - Etanol em Maravilha/SC		
	FH Combustíveis Rua Godofredo Rangel, Fidélis - 375	R\$ 1,980 Última atualização em 15/09/2014 21:33
	IG Combustíveis Rua José Bonfante, Vila Macarini - 353	R\$ 2,450 Última atualização em 15/09/2014 21:32

Tela 9: Tela com resultado da pesquisa (aplicação web)  
Fonte: O Autor (2014).

Ao selecionar, clicando em um dos registros apresentados, é direcionado à outra tela (Tela 10), em que através de outro método do web service é feita uma busca e retornado os dados e preços respectivos a entidade. O retorno é tratado e apresentado ao usuário.



The screenshot displays a web interface for 'FH Combustíveis'. It features a fuel pump icon, the company name, and the address: 'Rua Godofredo Rangel, Fidélis - 375, Maravilha - SC'. Below this, a section titled 'Valores' contains a table with two rows: 'Etanol' at 1,980 and 'Gasolina Comum' at 3,010.

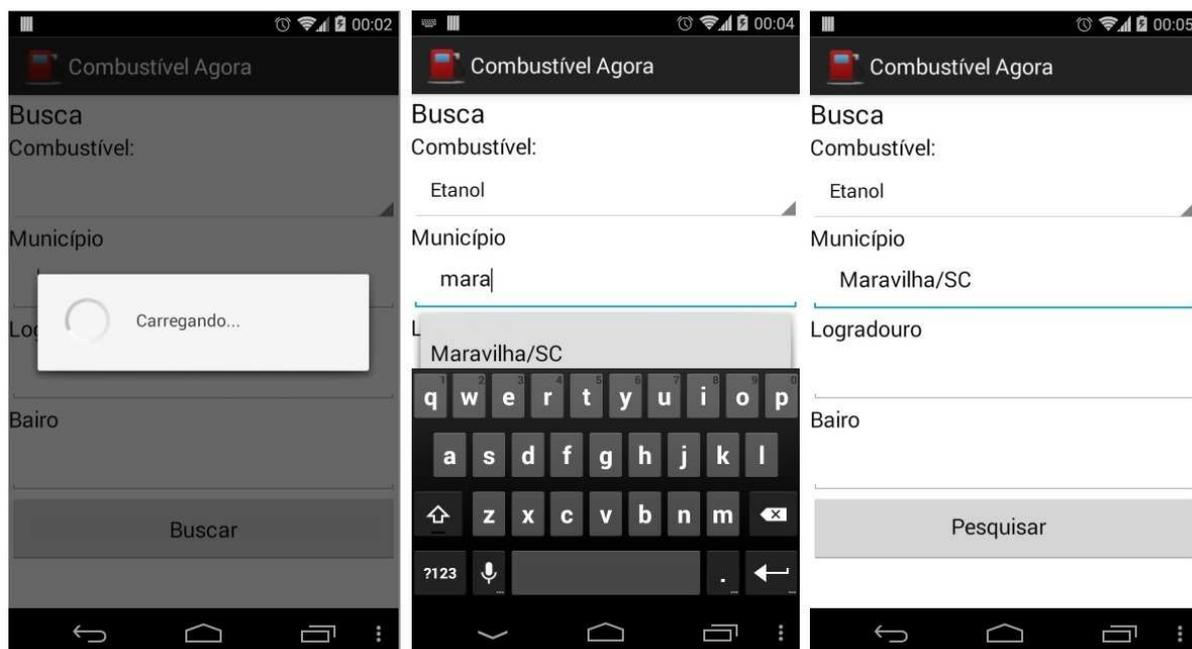
Valores	
Etanol	1,980
Gasolina Comum	3,010

Tela 10: Tela com dados da entidade (aplicação web)  
Fonte: O Autor (2014).

As buscas explanadas acima não ficam explícitas para o usuário, pois estas são tratadas internamente pela aplicação. Qualquer cadastro, sendo ele combustível, cidade ou entidade, necessário para a integração com o web service, é controlado através de um identificador único facilitando assim a busca das informações quando necessário.

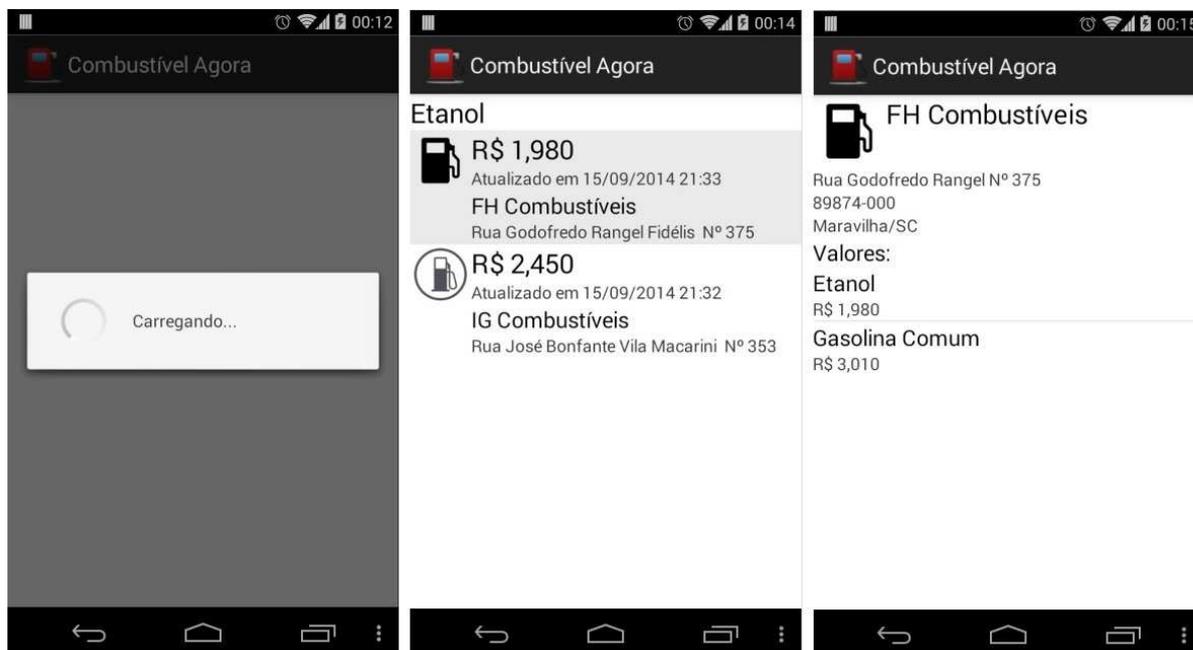
### 4.3 APLICAÇÃO MÓVEL

A aplicação móvel tem o funcionamento igual a aplicação web demonstrada acima, porém em outro ambiente e arquitetura, mas utilizando o mesmo canal de comunicação (web service). Na Tela 11 é observado o processo de sincronização e busca de informações.



Tela 11: Telas de busca (aplicação móvel)  
Fonte: O Autor (2014).

Também, com a Tela 12 nota-se o resultado da pesquisa efetuada, assim como os dados específicos após selecionar um dos registros retornados.



Tela 12: Telas de resultado (aplicação móvel)  
Fonte: O Autor (2014).

Mesmo tratando-se de outro ambiente, foi possível com suas próprias características específicas estabelecer o padrão de funcionamento entre as aplicações clientes.

## 5 CONSIDERAÇÕES FINAIS

A criação do ambiente proposto através do desenvolvimento das aplicações necessárias garantiu o controle e disponibilidade dos dados relativos aos preços dos combustíveis. Este, por sua vez, demonstrou que é possível integrar aplicações através de um único canal de comunicação, não se atendo a arquitetura ou linguagem de programação.

Este ambiente foi denominado "Combustível Agora", em que através de uma aplicação web é possível armazenar e disponibilizar as informações para as aplicações clientes, sendo elas, móvel e uma página web.

A escolha das tecnologias foi crucial para o desenvolvimento em um todo, pois influenciou diretamente no sucesso geral e garantiu ganho de tempo na implementação, assim como facilitou a manutenção quando necessária.

De maneira geral, através dos resultados obtidos é possível concluir que o presente trabalho atende os requisitos propostos e pode ser considerada uma rica solução para esta atividade.

## REFERÊNCIAS

ABLESON, W. Frank et al. **Android in Action**. 3. ed. [s. L.]: Manning, 2011.

ALMEIDA, Rodrigo Rebouças de. **Model-View-Controller: MVC**. Disponível em: <<http://www.dsc.ufcg.edu.br/~jacques/cursos/map/html/arqu/mvc/mvc.htm>>. Acesso em: 28 ago. 2014.

BRASIL. Agência Nacional do Petróleo, Gás Natural e Biocombustíveis. **IX Seminário de Avaliação do Mercado**. Disponível em: <<http://www.anp.gov.br/?pg=70174&m=&t1=&t2=&t3=&t4=&ar=&ps=&cachebust=1408971955025>> Acesso em: 25 ago. 2014.

BRAUDE, Eric. **Projeto de Software: Da programação à arquitetura: uma abordagem baseada em Java**. Porto Alegre: Bookman, 2005. 619 p.

BRISBIN, Jon et al. **Spring Data: Modern Data Access for Enterprise Java**. [s. L.]: O'reilly, 2012. 316 p.

BURNS, Ed; SCHALK, Chris. **JavaServer Faces: The Complete Reference**. Estados Unidos da América: Mcgraw-hill, 2010. 722 p.

CARVALHO, M. S. **Informação: da produção à utilização**. In: ROZENFELD, S. (Org.) **Fundamentos da Vigilância Sanitária**. Rio de Janeiro: Editora Fiocruz, 2000, p. 233-256.

Comitê Gestor da Internet no Brasil. **Dimensões e características da Web brasileira: um estudo do .gov.br**. 2010. Disponível em: <<http://www.cgi.br/media/docs/publicacoes/2/cgibr-nicbr-censoweb-govbr-2010.pdf>>. Acesso em: 25 ago. 2014.

CONALLEN, Jim. **Desenvolvendo aplicações Web com UML**. Rio de Janeiro: Campus, 2003. 476 p.

CROCKFORD, Douglas. **JavaScript: The Good Parts**. [s. L.]: O'reilly, 2008. 172 p.

CROCKFORD, Douglas. **The application/json Media Type for JavaScript Object Notation (JSON)**. 2006. Disponível em: <<http://www.ietf.org/rfc/rfc4627.txt>>. Acesso em: 27 ago. 2014.

DAYLEY, Brad. **NoSQL with MongoDB in 24 Hours**. [s. L.]: Sams, 2014a. 544 p.  
DEVELOPERS, Google. **Using JPA with App Engine**. 2014. Disponível em: <<https://developers.google.com/appengine/docs/java/datastore/jpa/overview?hl=pt-br>>. Acesso em: 28 ago. 2014.

DAYLEY, Brad. **Node.js, MongoDB, and AngularJS: Web Development**. [s. L.]: Addison Wesley, 2014b. 800 p.

DIONYSIO, Renata Barbosa; MEIRELLES, Fatima Ventura Pereira. **Combustíveis: A química que move o mundo**. Disponível em: <[http://web.ccead.puc-rio.br/condigital/mvsl/Sala de Leitura/conteudos/SL\\_combustiveis.pdf](http://web.ccead.puc-rio.br/condigital/mvsl/Sala%20de%20Leitura/conteudos/SL_combustiveis.pdf)>. Acesso em: 22 ago. 2014.

FIELDING, Roy Thomas. **Architectural Styles and the Design of Network-based Software Architectures**. Tese (Doutorado) — UNIVERSITY OF CALIFORNIA, IRVINE, 2000.

FLANAGAN, David. **Java: o guia essencial**. São Paulo: Bookman, 2005. 1097 p.

FLANAGAN, David. **JavaScript: The Definitive Guide**. 6. ed. [s. L.]: O'reilly, 2011. 1078 p.

FLANAGAN, David. **JQuery Pocket Reference**. [s. L.]: O'reilly, 2010. 160 p.

FRASER, Ryan; RANKINE, Terry; WOODCOCK, Robert. **Service Oriented grid Architecture for Geosciences Community**. 2007. Disponível em: <<http://crpit.com/confpapers/CRPITV68Fraser.pdf>>. Acesso em: 26 ago. 2014.

GONÇALVES, Edson. **Desenvolvendo Aplicações Web com JSP, Servlets, Javasever Faces, Hibernate, EJB 3 Persistence e Ajax**. Rio de Janeiro: Editora Ciência Moderna, 2007. 736 p.

GONÇALVES, Edson. **Desenvolvendo aplicações web com NetBeans IDE 6**. Rio de Janeiro: Editora Ciência Moderna, 2008.

HEFFELFINGER, David R.. **Java EE 7 with GlassFish 4 Application Server**. 3. ed. Birmingham: Packt, 2014. 348 p.

HLAVATS, Ian. **Instant PrimeFaces Starter**. Birmingham: Packt Publishing, 2013. 90 p.

JONNA, Sudheer. **Learning PrimeFaces Extensions Development**. Birmingham: Packt Publishing, 2014. 192 p.

JSON (2014). Disponível em: <<http://www.json.org>>. Acesso em: 27 ago. 2014.

KAINULAINEN, Petri. **Spring Data**. Birmingham: Packt, 2012. 160 p.

KALIN, Martin. **Java Web Services: Implementando**. [s. L.]: Atla Books, 2009. 312 p.

KEITH, Mike; SCHINCARIOL, Merrick. **Pro EJB 3: Java Persistence API**. [s. L.]: Apress, 2006. 480 p.

KEITH, Mike; SCHNICARIOL, Merrick. **Pro JPA 2: Mastering the JavaTM Persistence API**. Estados Unidos da América: Apress, 2009. 481 p.

KONDA, Madhusudhan. **Just Spring Data Access**. [s. L.]: O'reilly, 2012. 78 p.  
KOTLER, Philips, AMSTRONG, Gary. **Princípios de Marketing**. 7 ed. Rio de janeiro. LTC Ed S/A, 1998.

LARMAN, Craig. **Utilizando UML e padrões: uma introdução à análise e ao projeto orientados a objetos e ao processo unificado**. 2. ed. Porto Alegre: Bookman, 2004. 607 p.

LISBOA, Flavio Gomes da Silva. **Introdução ao Demoiselle Framework: Uma abordagem comparativa de Utilização do padrão MVC para o desenvolvimento de aplicações Web em Java orientada ao reuso**. 2010. Disponível em: <<http://www.youblisher.com/p/98445-Introducao-ao-Demoiselle-Framework/>>. Acesso em: 01 set. 2014.

LISBOA, Flávio Gomes da Silva. Made in Brazil: Padronização e Reuso de Aplicações Web com Demoiselle Framework. **Revista Mundo Java: jrubby da Web ao Desktop**, Curitiba, n. 36, p.8-17, 01 jul. 2009. Bimestral.

MACORATTI, José Carlos. **Padrões de Projeto: O modelo MVC - Model View Controller**. Disponível em: <[http://www.macoratti.net/vbn\\_mvc.htm](http://www.macoratti.net/vbn_mvc.htm)>. Acesso em: 28 ago. 2014.

MARQUES, Cícero Fernandes. **Marketing para negócios empresariais**. Curitiba: Gráfica Fotolaser, 2008.

MONGODB. **MongoDB Documentation Project**. 2014. Disponível em: <<http://docs.mongodb.org/manual/MongoDB-manual.pdf>>. Acesso em: 29 ago. 2014.

NETO, Camilo Lopes de Medeiros. **Guia Prático: Construindo Aplicações JEE com Frameworks**. Rio de Janeiro: Editora Ciência Moderna Ltda., 2011.

PELEGRI-LLOPART, Eduardo; YOSHIDA, Yutaka; MOUSSINE-POUCHKINE, Alexis. **The GlassFish Community: Delivering a Java EE Application Server**. 2007. Disponível em: <<https://glassfish.java.net/faq/v2/GlassFishOverview.pdf>>. Acesso em: 01 set. 2014.

PEREIRA, Lúcio Camilo de Oliveira; SILVA, Michel Lourenço da. **Android Para Desenvolvedores**. 2. ed. [s. L.]: Brasport, 2012. 240 p.

PEREIRA, Maria José Lara de Bretãs; FONSECA, João Gabriel Marques. **Faces da Decisão: as mudanças de paradigmas e o poder da decisão**. São Paulo: Makron Books, 1997.

RICHARDSON, Leonard; RUBY, Sam. **RESTful Serviços Web**. Rio de Janeiro: Alta Books, 2007, 336 p.

RUTTER, Jake. **Smashing jQuery**. [s. L.]: John Wiley & Sons, 2011. 336 p.

SACRAMENTO, Cleverson et al. **Framework Demoiselle 2.3.0: Guia de Referência**. 2013. Disponível em: <<http://demoiselle.sourceforge.net/docs/framework/reference/2.3.4/html/>>. Acesso em: 01 set. 2014.

SILVEIRA, Paulo et al. **Introdução À Arquitetura e Design de Software: Uma Visão Sobre a Plataforma Java**. [s. L.]: Campus, 2011. 280 p.

TIWARI, Shashank. **Professional NoSQL**. Indianapolis: John Wiley & Sons, 2011.

TREINAMENTOS, K-19. **Integração de Sistemas com Webservices, JMS e EJB.** 2013. Disponível em: <<http://www.k19.com.br/downloads/apostilas/java/k19-k23-integracao-de-sistemas-com-webservices-jms-e-ejb>>. Acesso em: 26 ago. 2014.