

CENTRO PAULA SOUZA



FACULDADE DE TECNOLOGIA DE AMERICANA
Curso Superior de Bacharelado em Análise de Sistemas e Tecnologia da
Informação

Aline Crippa Borges

DESENVOLVENDO UM SISTEMA WEB COM O EXTJS

Americana, SP

2013

FACULDADE DE TECNOLOGIA DE AMERICANA
Curso Superior de Bacharelado em Análise de Sistemas e Tecnologia da
Informação

Aline Crippa Borges

DESENVOLVENDO UM SISTEMA WEB COM O EXTJS

Trabalho monográfico, desenvolvido em cumprimento à exigência curricular do Curso Superior de Bacharelado em Análise de Sistemas e Tecnologia da Informação da Fatec Americana, sob orientação do Prof. Esp. Antônio Alfredo Lacerda.
Área de concentração: Programação.

Americana, SP

2013

Aline Crippa Borges

DESENVOLVENDO UM SISTEMA WEB COM O EXTJS

Trabalho de conclusão de curso apresentado à Faculdade de Tecnologia de Americana como parte dos requisitos para obtenção do Título de Bacharel em Análise de Sistemas e Tecnologia da Informação.
Área de concentração: Programação.

Americana, 02 de Dezembro de 2013.

Banca Examinadora:

Antônio Alfredo Lacerda (Presidente)
Especialista
FATEC Americana

Nome completo do membro da banca (Membro)
Maior titulação
FATEC Americana

Nome completo do membro da banca (Membro)
Maior titulação
FATEC Americana

AGRADECIMENTOS

À Deus, em primeiro lugar, pelas bênçãos derramadas em minha vida e pela fé e capacitação de estar concluindo mais esta etapa.

Ao professor Especialista Antônio Alfredo Lacerda, pela dedicação e paciência ao me orientar e auxiliar no desenvolvimento deste trabalho.

Ao meu namorado Willian Albane, pelo apoio, carinho e motivação para o desenvolvimento deste projeto.

Aos meus pais Sandra M. Crippa e Robinson T. Bragotto pela ajuda, compreensão, motivação durante toda minha graduação e pela educação durante toda minha vida.

À minha irmã Ariane Crippa Borges pela paciência nos momentos difíceis e ao meu cunhado Afonso M. Silva pela compreensão.

À minha tia Roseline Crippa pela verificação e correção dos conteúdos em inglês e aos meus familiares pelo carinho e dedicação ao longo dos anos.

Aos meus amigos e companheiros do curso Amanda H. M. M. dos Santos, Matheus A. Silveira e Raquel Alves pela amizade conquistada e cultivada, pelo apoio e companheirismo no cotidiano e aos meus colegas de curso pela amizade e compartilhamento de conhecimento.

Aos meus pastores Ciro, Ozaide, Moacir e Betinha que oram e intercedem pela minha vida e à minha amiga Sílvia H. Stefano Goes que ora por mim e me dá forças para lutar e conquistar meus objetivos através de sua intercessão.

À FATEC - Americana, pelo apoio e amizade demonstrada pelos profissionais.

DEDICATÓRIA

Aos meus pais, ao meu namorado, ao professor Esp. Antônio Lacerda, e a todos que me ajudaram durante essa etapa da minha vida.

RESUMO

O trabalho aborda os conteúdos do *framework* ExtJS e de um sistema *web*, juntamente com um estudo de caso do sistema *web* e um projeto prático do sistema utilizando a interação com o *framework* para um sistema que pode ser aplicado no cotidiano. Neste projeto foi estudado e descrito sobre sistema de informação, focando o *framework* ExtJS e sua definição, vantagens e desvantagens e sua diversidade de versões. O trabalho foi descrito baseado em pesquisas bibliográficas e projeto experimental de integração, desenvolvimento e aplicação. O *framework* estudado permitiu elucidar um sistema *web* simples, comum no cotidiano. Foi utilizada no projeto prático basicamente com ênfase na programação da linguagem PHP com códigos simples, visando facilitar o entendimento e melhorar o tempo do desenvolvimento.

Palavras-chave: *Framework*. EXTJS. Sistema *Web*.

ABSTRACT

The work approaches the contents of the ExtJS framework and a web system, along with a case study of web system and a practical project using the system interaction with the framework for it, that can be applied in everyday life. This project has been studied and described on information systems, focusing on framework, highlighting the ExtJS and its definition, advantages and disadvantages, versions, etc. The work was described based on bibliographical researches and experimental project integration and development, applying the framework studied in an idea simple system web common everyday. It was used in the practical design basically programming in PHP language with simple codes to facilitate the understanding and improve the time of development.

Keywords: Framework. EXTJS. Web System.

LISTA DE ILUSTRAÇÕES

Figura 1 - Usuário <i>Web</i> - Solicitação e Resposta para uma Página <i>Web</i>	19
Figura 2 - Fluxo de Eventos e Informações em uma Arquitetura MVC.....	25
Figura 3 - Endereço do site Sencha	26
Figura 4 - Aba <i>Products</i> da Sencha.....	27
Figura 5 - Página para <i>download</i> do Sencha ExtJS	27
Figura 6 - <i>Download</i> do ExtJS	28
Figura 7 - Página de Conclusão do <i>download</i> do ExtJS	28
Figura 8 - Página inicial da documentação do ExtJS 4.2	29
Figura 9 - MER Banco de Dados Sistema de Agendamento	33
Figura 10 - Caso de Uso do Sistema de Agendamento para Salão de Beleza	34
Figura 11 - Diagrama de Atividade Cadastrar Novo Cliente no Sistema	35
Figura 12 - Diagrama de Atividade Agendar Horário pelo Sistema	36
Figura 13 - Tela de Login do Sistema	37
Figura 14 - Tela de Menu do Sistema	38
Figura 15 - Tela de Cadastro de Usuários - Atendente	38
Figura 16 - Tela de Cadastro de Usuários - Cliente	39
Figura 17 - Tela de Agenda Calendário – Atendente	39
Figura 18 - Tela de Agenda Calendário - Cliente	40

LISTA DE TABELAS

Tabela 1 - Vantagens e Desvantagens do ExtJS	31
--	----

LISTA DE ABREVIATURAS E SIGLAS

- CSS:** Cascading Style Sheets (Linguagem de Estilo para HTML)
- DDS:** *Decision Support System* (SAD – Sistema de Apoio a Decisão)
- EIS:** *Executive Information Systems* (Sistemas de Informação Executiva)
- ERP:** *Enterprise Resource Planning* (Planejamento de Recursos Empresariais)
- HTML:** *HyperText Markup Language* (Linguagem de Marcação de Hipertexto)
- HTTP:** *HyperText Transfer Protocol* (Protocolo de Transferência de Hipertexto)
- IDE:** *Integrated Development Environment* (Ambiente Integrado de Desenv.)
- MER:** Modelo Entidade-Relacionamento
- MIT:** *Massachusetts Institute Technology* (Instituto Tecnologia Massachusetts)
- MVC:** *Model View Controller* (Modelo Visão Controlador)
- PDF:** *Portable Document Format* (Formato de Documento Portátil)
- PHP:** *Personal Home Page*
- RIA:** *Rich Internet Application* (Aplicação Rica em Web)
- URL:** *Uniform Resource Locator* (Localizador Padrão de Recursos)
- XML:** *Extensible Markup Language*
- YUI:** *Yahoo! User Interface*
- WEB:** *World Wide Web* (Conhecido como WWW ou W3)

SUMÁRIO

1. INTRODUÇÃO	11
2. SISTEMA DE INFORMAÇÃO	14
2.1. ENTENDENDO O QUE É UM SISTEMA DE INFORMAÇÃO	14
2.2. EVOLUÇÃO DO SISTEMA DE INFORMAÇÃO	14
2.3. VANTAGENS DE UM SISTEMA DE INFORMAÇÃO E SUA RELAÇÃO NAS ORGANIZAÇÕES..	16
2.4. INTRODUÇÃO AO SISTEMA DE INFORMAÇÃO WEB	17
2.4.1. Linguagem da Web	19
2.5. RECURSOS NECESSÁRIOS PARA DESENVOLVIMENTO DE SISTEMAS WEB.....	20
3. FRAMEWORK	22
3.1 PRINCIPAIS TIPOS DE FRAMEWORKS.....	22
3.2. FRAMEWORK EXTJS	23
3.2.1. História do Framework ExtJS	23
3.2.2. Arquitetura MVC no Framework ExtJS	24
3.2.3. Documentação do Framework ExtJS	26
3.2.4. Versões do Framework ExtJS	30
3.2.5. Vantagens e Desvantagens de Utilizar o ExtJS	31
4. DESENVOLVENDO APLICAÇÃO WEB COM FRAMEWORK EXTJS.....	32
4.1. REQUISITOS E FERRAMENTAS UTILIZADAS PARA O DESENVOLVIMENTO DO PROJETO .	32
4.2. DOCUMENTAÇÃO DO SISTEMA DESENVOLVIDO.....	33
4.2.1. MER (Modelo Entidade Relacionamento)	33
4.2.2. Caso de Uso do Sistema de Agendamento	33
4.2.3. Diagramas de Atividade das Principais Funcionalidades do Sistema	35
4.3. PROGRAMAÇÃO E DESENVOLVIMENTO	36
4.4. LAYOUT DO SISTEMA DE AGENDAMENTO.....	37
5. CONSIDERAÇÕES FINAIS	41
REFERÊNCIAS.....	42
ANEXO – CÓDIGO DE PARTE DA PROGRAMAÇÃO DO SISTEMA DE	
AGENDAMENTO	44

1. INTRODUÇÃO

Esse trabalho foi desenvolvido em cumprimento escolar para obtenção do Título de Bacharel em Análise de Sistemas e Tecnologia da Informação. Como o tema abordado não foi abordado como disciplina na grade curricular, sendo muito empregado atualmente a nível de desenvolvimento, foi estudado o *framework* ExtJS para ter melhor conhecimento dos recursos e usabilidades do mesmo, desenvolvendo um sistema básico para melhor entendimento do *framework* e aplicá-lo em minha rotina no serviço.

Para obter uma melhor visão deste *framework*, foi estabelecida a seguinte pergunta: Como o *framework* ExtJS pode auxiliar e facilitar um programador no desenvolvimento de um sistema?

Portanto, este projeto irá apresentar o *framework* ExtJS, que otimiza o desenvolvimento de sistemas, objetivando facilitar e agilizar a programação de alguns recursos requeridos no sistema.

Com esse *framework*, ExtJS, é possível agilizar e facilitar o desenvolvimento do sistema durante a criação de interface, podendo, também, padronizar as mesmas. Por ser considerado um facilitador de código muito utilizado por programadores na criação de recursos mais complexos do projeto, farei uma abordagem prática desenvolvendo um sistema simples que utilize o ExtJS para visualizar sua interface e seus recursos em um sistema e, também, uma abordagem teórica para entender melhor os conceitos e os procedimentos para se utilizar essa ferramenta junto de seus diversos recursos.

Foi realizado o levantamento bibliográfico sobre o *framework* em questão e seus recursos, buscando esclarecer para que serve, o que é, quando deve ser utilizado e alguns de seus recursos.

Neste trabalho será demonstrado o estudo e o funcionamento do *framework*, visando otimizar o tempo de desenvolvimento e facilitar a programação e criação da interface através do ExtJS. Será desenvolvido um sistema básico para exemplificar o uso do *framework* ExtJS em relação aos sistemas que possibilitam seu uso para auxiliar no desenvolvimento de um sistema com recursos mais complexos.

O *framework*, em estudo, possibilita padronizar um ambiente visual, ou seja, uma interface, reduzindo o tempo que o desenvolvedor necessita para desenvolver um sistema ou uma aplicação e facilitar o desenvolvimento.

O trabalho foi realizado utilizando pesquisa bibliográfica, que, segundo Silva (2001), é compreendida quando o trabalho é elaborado a partir de material já publicado, é constituído principalmente de livros, artigos de periódicos e atualmente com material disponibilizado na Internet.

O trabalho também utilizará de pesquisa exploratória para melhor compreensão do assunto, segundo Silva (2001), visa proporcionar uma visão geral de um determinado fato, do tipo aproximativo. Este tipo de pesquisa é realizado especialmente quando o tema escolhido é pouco explorado e torna-se difícil sobre ele formular hipóteses precisas e operacionalizáveis.

O princípio deste projeto é mostrar de forma prática como que este *framework* pode contribuir em um ambiente de programação, bem como suas praticidades e facilidades, passível de investigação mediante procedimentos sistematizados.

O projeto será útil para adquirir um conhecimento mais profundo sobre o que é um *framework* e como o mesmo atua em uma interface, como e quando utilizá-lo. Assim, com esse conhecimento, o projeto auxiliará na vida profissional, tornando possível um melhor desenvolvimento e maior otimização e, possível manutenção do *software*.

Os assuntos abordados no projeto estão dispostos conforme descrito a seguir.

Capítulo 1 – Introdução: Nesse tópico será abordada a introdução sobre o assunto do projeto, disponibilizando um entendimento geral sobre o que estará disponível no projeto para estudo e análises e a metodologia utilizada para seu desenvolvimento.

Capítulo 2 – Sistemas *Web*: O tópico abordará uma explicação sobre o que é um sistema e quais são os requisitos básicos para desenvolvê-lo.

Capítulo 3 – *Framework*: O tópico abordará as funções e os componentes de um *framework* e quando utilizar em alguns projetos a ser desenvolvido. Será utilizado um *framework*, chamado ExtJS, para auxiliar na criação de interfaces e recursos complexos.

Capítulo 4 – Desenvolvendo Aplicação *Web* com *Framework* ExtJS: Nesse tópico será desenvolvido um sistema básico de agendamento de horário para melhor entendimento e aplicação dos recursos do *framework* ExtJS 4.

Capítulo 5 – Considerações Finais: Nesse tópico será descrito a conclusão obtida em relação a toda a pesquisa realizada, buscando encontrar o momento da aplicação deste *framework* em questão e um conhecimento não apenas teórico, mas sim prático do assunto abordado.

Referências: Nesse tópico serão especificados os livros e pesquisas utilizadas para busca de conhecimento no assunto abordado e base para fundamentação teórica de todo o projeto desenvolvido.

2. SISTEMA DE INFORMAÇÃO

2.1. Entendendo o que é um Sistema de Informação

Um sistema de informação pode ser definido como um conjunto de componentes inter-relacionados trabalhando juntos para coletar, recuperar, processar, armazenar e distribuir informações com a finalidade de facilitar o planejamento, o controle, a coordenação, a análise e o processo decisório em organizações (LAUDON E LAUDON, 1999).

Um sistema de informação é uma combinação de fatores que se relacionam para coletar, manipular e propagar informações para obter um mecanismo de realimentação para alcançar um objetivo (STAIR e REYNOLDS, 2006).

Um sistema de informação também pode ser definido como um sistema que manipula e controla arquivos e dados utilizando ou não de tecnologia da informação (GONÇALVES, 2012).

2.2. Evolução do Sistema de Informação

Segundo Toni (2013), entre as décadas de 70 e 80 a tecnologia de informação era considerada um mal necessário, pois, além de um custo alto, o retorno não era imediato e os investimentos nessa área eram sempre discutidos. Porém, no final do século XX, a tecnologia da informação tornou-se uma ferramenta essencial para todas as organizações, pois, através do uso das tecnologias disponíveis, os processos eram observados e gerenciados com maior facilidade. Desse modo, a tecnologia da informação ganhou status e necessidade nas organizações.

Conforme a maturidade das empresas com relação às tecnologias, as empresas acabam por adquirir maior competitividade, nos diversos segmentos de desenvolvimento dos produtos, serviços e custos. Para atingir esses resultados, a qualidade e a reengenharia foram ferramentas muito utilizadas. Porém, ainda haviam muitas dificuldades localizadas na gestão interna dos processos das organizações. Para acatar essa demanda foram implementados os primeiros sistemas integrados de gestão (ERP) que forneciam um controle automatizado dos processos, inter-relacionando os setores da empresa (TONI, 2013).

Diz ainda o autor que a principal função do ERP (*Enterprise Resource Planning*) é integrar as informações. Por meio de um *software* e uma base de dados central é realizada a troca de informações entre todos os setores de uma empresa, permitindo acompanhar um processo qualquer de forma ágil e precisa.

Com o aumento da competição do mercado e a necessidade que as organizações possuem em estar sempre à frente dos concorrentes, foi desenvolvida a tecnologia EIS (*Executive Information Systems*) ou Sistemas de Informação Executiva (TONI, 2013).

Segundo Furlan, Ivo, Amaral (1994), os EIS são considerados sistemas computacionais designados a atender necessidades de informação dos executivos, tendendo acabar com a necessidade de intermediários entre os executivos e a tecnologia. O termo EIS surgiu no final da década de 70 como consequência de tarefas desenvolvidas por Rockart e Treacy, que eram pesquisadores do MIT (Instituto de Tecnologia de Massachusetts). O EIS busca unir todas as informações indispensáveis e disponíveis para que o executivo possa, através de gráfica ou relatórios, acompanhar e controlar os negócios da organização em um único sistema de informação.

O EIS trabalha com uma única base de dados, tanto física ou lógica, possibilitando ao executivo tomar decisões através de todas as informações disponíveis. Para possibilitar que as informações cheguem aos executivos, são instalados dispositivos computacionais que possibilitam coletar de dados em qualquer base de dados de qualquer setor da organização e, assim, chegar aos executivos rapidamente (TONI, 2013).

Segundo o autor citado anteriormente, o conceito de EIS é muito confundido com o de SAD (Sistemas de Apoio a Decisão) ou DSS (*Decision Support System*). Apesar de estarem relacionados, a diferença se dá pelo público-alvo que ambos atendem e o tipo de problemas que resolvem. Enquanto o EIS busca disponibilizar informações para os executivos sem manipular os dados, mas possibilitando a visualização em muitos níveis de detalhes, o DSS visa atender o nível intermediário de gerência, possibilitando, assim, a interação por meio de análises lógicas que

possibilitam fazer previsões. Deste modo o usuário pode saber de informações consultando na forma de questionamentos.

Segundo Toni (2013), o DSS surgiu através da grande crise do petróleo na década de 70, obrigando os executivos a utilizarem os recursos computacionais para obterem sucesso nos investimentos, através de análises de cenários e previsões para os investimentos e as organizações. Os modelos mais comuns eram os financeiros baseados em fórmulas predefinidas para trabalhar as variáveis informadas e mostrar resultados ou possíveis situações futuras. O DSS integra vários sistemas de processamento de transações, fazendo acesso à base de dados relacionando com outras bases para a tomada de decisões. O seu crescimento foi instantâneo nas organizações, pois a necessidade de auxílio para tomada de decisões em vários gerenciais era muito grande.

Diz o autor ainda que esta evolução persiste com *softwares* ligados ao relacionamento com clientes, fornecedores e que origem conhecimento e não somente informações para as empresas, uma vez que através do conhecimento é possível agir e tomar decisões concisas e não apenas por intuição.

2.3. Vantagens de um Sistema de Informação e sua Relação nas Organizações

Segundo Frazatto (2010), as vantagens de se utilizar um sistema de informação em uma empresa se dão pelos dos seguintes fatores: redução de custos, produtividade, integração, internet e atendimento ao cliente.

Diz a autora ainda que a redução de custos é causada pelo avanço tecnológico de computadores que passam a fazer atividades que antes demoravam dias e agora, com a informatização, demoram horas.

A produtividade está relacionada à automatização dos processos e a um cálculo que divide o salário pelo tempo gasto por um funcionário em realizar uma tarefa manual, sendo que o resultado desse cálculo é a produtividade que a empresa possui e, com a informatização do setor, a produtividade de algumas tarefas pode aumentar, gerando uma melhor possibilidade para a empresa na competição do mercado (FRAZATTO, 2010).

A integração é uma ótima oportunidade para que seja possível a relação da empresa com o mercado de vendas e compras eletrônicas e, também, com vendas para grandes clientes que estão atualizando seu método de compra para portais ou através de acordos eletrônicos, diz (FRAZATTO, 2010).

Segundo a autora citada anteriormente, a internet é uma vantagem enorme para que a empresa possa se comunicar com os clientes através da tecnologia da *web*. Além de integrar todos os fatores citados acima, é possível vender produtos e serviços da empresa pela internet, integrar negócios e parcerias de forma mais ágil e barata e, proporciona um melhor atendimento aos clientes.

O atendimento ao cliente é um fator que pode fidelizar um cliente em potencial, pois, através de um bom atendimento possibilitado por ferramentas tecnológicas simples ou complexas, dependendo da necessidade do negócio da empresa, o cliente considerará seu nível de satisfação depois de ser atendido para escolher os produtos de uma empresa e/ou indicá-la para outros (FRAZATTO, 2010).

Segundo o Portal da Educação (2013), há, também, outras vantagens em se utilizar um sistema de informação, conforme citadas abaixo.

- Maior integridade e veracidade das informações
- Maior segurança de acesso à informação
- Maior estabilidade
- Otimização do fluxo de informações para maior agilidade e organização

2.4. Introdução ao Sistema de Informação Web

Segundo Guanais (2008), aplicação ou sistema *web* é um termo usado para representar sistemas de informação cogitados para uso por meio de um navegador, tanto na internet quanto na Intranet, sendo um conjunto de programas executado em um servidor HTTP. O desenvolvimento *web* está ligado à necessidade de facilitar a atualização e manutenção conservando o código-fonte no mesmo local, de onde ele é acessado por diversos usuários.

Uma melhor definição para aplicação ou sistema *web* é dada por um *software* que faz uso da *web*, por meio de um *browser*, como uma aplicação de *software* que utiliza a *web*, através de um *browser*, como ambiente de execução e, também, pode ser considerado tudo que é realizado em um servidor qualquer (GUANAIS, 2008).

O autor ainda descreve que a diferença entre um *site web* e uma aplicação *web* é que uma aplicação antes de ser exibida na página acessada, é executada por um computador conectado à internet e, após a execução, são retornadas as regras de negócios e os dados necessários utilizados por essa aplicação e no *site* não é necessário o processamento em um computador, antes da exibição para o usuário.

Assim, em um sistema *web*, é necessário um servidor para efetuar a comunicação entre as solicitações enviadas e recebidas, possibilitando que o usuário requira um recurso, através de um navegador, que retorne para ele documentos em formato PDF, imagens e gráficos ou páginas em HTML após o servidor encontrar e responder a solicitação necessitada, conforme descrito por Guanais (2008).

O servidor possui alguns recursos, mas não consegue processar todos os recursos sozinho como, por exemplo, criação de páginas dinâmicas e armazenamento de dados em um banco de dados. Uma página dinâmica tem a função de auxiliar, através de uma aplicação passada pelo *servlet*, que ocorra a comunicação de uma aplicação estática disponibilizada e rodada pelo servidor. O armazenamento de dados no servidor ocorre através de uma aplicação de apoio (*servlet*) para o qual o servidor envia os parâmetros para armazenar os dados em um banco de dados. (PALMEIRA, 2012)

Segundo o autor acima, podem ocorrer falhas de segurança em diferentes etapas como, por exemplo, na análise de requisito, na especificação e implementação do *software*. Esses riscos de vulnerabilidade da aplicação em uma empresa podem gerar impactos.

Segundo Palmeira (2012), o HTTP utiliza um modelo de solicitações e respostas, no qual uma solicitação acontece quando o usuário faz um requerimento HTTP e o servidor *web* retorna uma resposta HTTP com o *browser* verificando como tratar esse conteúdo recebido. Caso a resposta que veio do servidor seja uma

página HTML, então é adicionado na resposta HTTP. A seguir há uma figura demonstrando como funciona uma solicitação *web* realizada por um usuário.

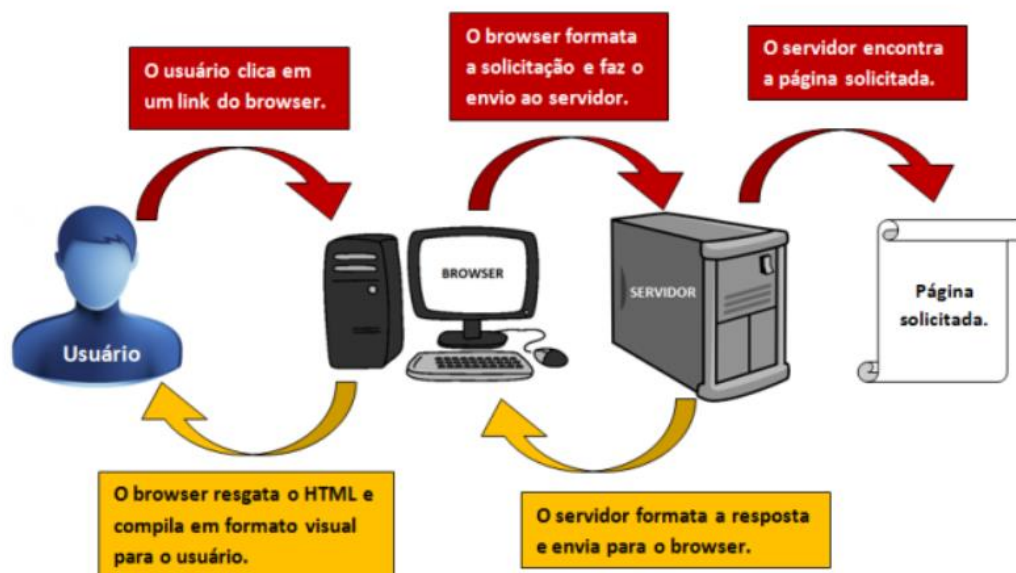


Figura 1 - Usuário Web - Solicitação e Resposta para uma Página Web

Fonte: PALMEIRA (2012)

Há duas solicitações bastante utilizadas pelo HTML, o *GET* e o *POST*. A diferença entre as duas solicitações é que enquanto o *GET* vincula dados do formulário no final da *URL*, o *POST* adiciona dados do formulário no corpo do requerimento (PALMEIRA, 2012).

2.4.1. Linguagem da Web

A maior parte do conteúdo na *Web* é construído a partir de páginas que utilizam a linguagem HTML (*HyperText Markup Language*), pois é a linguagem que deu início ao desenvolvimento *web* e continua presente e bem utilizada atualmente (WATERS, 1999).

O autor ainda diz que com a linguagem é possível desenvolver páginas simples e sofisticadas, onde pode ser escrita até mesmo em um bloco de notas, porém não é o mais cômodo. Atualmente há varias ferramentas que auxiliam a codificação do sistema ou da página. Podemos dividir essas ferramentas em três grupos, conforme a seguir.

- Editores HTML: São ferramentas que dão ênfase a liberdade para criação de páginas, sem trabalhar muito com o código, mas utilizando muito a programação visual.
- Macros para processadores de texto: São os editores de texto, onde é capaz utilizarmos o ambiente do processador de textos para familiarizar com o usuário, embora não seja possível utilizar recursos mais sofisticados visuais.
- Conversores: São ferramentas capazes de converter os mais variados formatos em HTML.

A linguagem HTML tem suas qualidades e suas deficiências, inclusive com a questão de páginas dinâmicas, que foi onde entraram as linguagem para evolução do HTML, uma delas é o XML (*eXtensible Markup Language*) na evolução do HTML seria XHTML. A linguagem XML permite que qualquer tipo de informação seja integrado a *web*, cobrindo uma variedade de aplicações e trabalhando como serviços de distribuição *Web* (WATERS, 1999).

2.5. Recursos Necessários para Desenvolvimento de Sistemas *Web*

Os recursos básicos necessários para um desenvolvimento de sistema *web* são praticamente os mesmos que para desenvolver um *software* ou sistema local, no entanto, utilizam-se alguns recursos adicionais para possibilitar que o sistema funcione pela *web*. A seguir, estão listados os itens para o desenvolvimento.

- Banco de Dados: é um recurso que deve ser escolhido de acordo com as necessidades e usabilidades que o cliente desejar. É possível citar alguns bancos de dados conhecidos como, por exemplo, o Postgres, o SQL e o Oracle (REZENDE, 2004).
- Servidor *Web*: é necessário para intermediar a relação entre envios e recebimentos dos dados em um sistema, um site ou qualquer informação solicitada/enviada por um usuário ou um computador, ou seja, oferecer serviços em uma rede através de um conjunto de *hardwares* e *softwares*. Alguns servidores *web* mais utilizados e/ou conhecidos atualmente são o Tomcat, o Apache, o Websphere e o Zeus Web Server (NET, 2010).

- Linguagem de Programação: para se definir um bom ambiente de programação, que deve ter integração com o banco de dados e o servidor, é muito importante escolher em qual linguagem será desenvolvido o sistema. Podem ser citadas algumas linguagens bastante conhecidas atualmente para desenvolvimento como, por exemplo, PHP, Java, C, C#, C++ e Delphi (MATTOS, 2012).
- Ferramentas: as ferramentas são escolhidas segundo as necessidades para desenvolvimento e implantação de cada sistema individualmente. Há inúmeros tipos de ferramentas. Relacionado à programação pode-se citar Sublime Text, Dreamweaver, Notepad ++, entre diversas outras. Há, também, ferramentas de suporte e auxílio ao programador, chamadas *frameworks*, utilizadas de acordo com o sistema que será desenvolvido. Para os *framework* podem ser citados o ExtJS e o JQuery, etc. No caso, para banco de dados, temos as ferramentas e/ou os sistemas gerenciadores pgAdmin, SQLServer e MySQL, Firebird e, entre outros, DBDesigner (REZENDE, 2004; NET, 2010 MATTOS, 2012).

3. FRAMEWORK

Framework é um conjunto de classes que colaboram para realizar uma responsabilidade para um domínio de um subsistema da aplicação (Fayad e Schmidt).

Um *framework* nada mais é do que um conjunto de códigos prontos que tem como objetivo facilitar o processo de desenvolvimento de sistemas e *softwares* (SAUVÉ, 2006).

Segundo Sauv  (2006), um *framework* necessariamente deve ser us vel e reus vel, com uma boa documenta o e facilidade de uso. Deve ter possibilidade de ser completado (expans vel), possuir um uso seguro, deve ser completo, e ser eficiente, pois a ideia de um *framework*   que ele seja aplic vel em v rias situa es.

Como diz o autor, um *framework* pode atender diversas necessidades ou uma espec fica de acordo com a configura o durante a programa o do sistema. Por tanto h  v rios tipos de *framework* para atender diferentes funcionalidades. Sendo assim, uma empresa pode utilizar v rios *frameworks*.

3.1 Principais Tipos de *Frameworks*

Os principais tipos de *frameworks* s o os seguintes: de suporte, de aplica o e de dom nio. Cada um destes *frameworks* visa atender um conjunto de necessidades espec ficas (SAUV , 2006).

Segundo Sauv  (2006), o *framework* de suporte possui caracter sticas como proporcionar servi os de sistemas operacionais com computa o distribu da e acesso a arquivos. O *framework* de aplica o   conhecido tamb m como *framework* horizontal e soluciona apenas uma parte do problema da aplica o. O *framework* de dom nio, conhecido como *framework* vertical, resolve boa parte dos problemas da aplica o.

Diz o autor ainda que a diferen a entre os *frameworks* vertical e horizontal   que no caso do vertical o conhecimento encapsulado   aplicado a sistemas e pertence a um dom nio espec fico de problemas enquanto o horizontal possui conhecimento encapsulado aplicado a uma grande gama de sistemas. H  ainda um

tipo de *framework* para aplicações *web*, que se baseia no *framework* de aplicação, mas está relacionado ao desenvolvimento de sistemas para ambiente *web*. Esse último tipo de *framework* tem como objetivo prestar suporte e oferecer serviços de desenvolvimento para aplicações *web*, serviços *web* e sites dinâmicos.

3.2. Framework ExtJS

Segundo Groner (2011), o *framework* ExtJS é orientado a objetos e todo escrito em JavaScript, com o foco para desenvolvimento RIA (*Rich Internet Application*) ou para aplicações na internet com interfaces ricas. A arquitetura utilizada pelo *framework* é a MVC (*Model View Controller*) e o mesmo é de fácil integração, além da variedade de componentes que possui e a integração com o Adobe AIR e com o GWT (*Google Web Toolkit*).

O *framework* possui dois tipos de licença, comercial (paga) e a *open source* (código aberto). O ExtJS se encontra atualmente na versão 4.2 e a distribuição é realizado pela empresa Senchalnc, com localização na Califórnia, Estados Unidos (GRONER, 2011).

Segundo Groner (2011), o ExtJS é um *framework* “*cross-browser*”, rodando, assim, até mesmo no Internet Explorer 6, fornecendo um desenvolvedor de ferramentas que cria interfaces para páginas e sistemas baseados em *web* com uma aparência elegante, alta performance e customização. Ele é especialmente utilizado para desenvolvimento lado cliente e lado servidor e, para o desenvolvimento, é possível utilizá-lo em qualquer linguagem de programação *web*, como PHP, Java, ASP.Net, entre outras.

3.2.1. História do *Framework* ExtJS

Criado por Jack Slocum, o ExtJS iniciou como extensão do site Yahoo (YUI), onde teve apoio da comunidade. Quando foi criado, o *framework* chamava-se yui-ext, trabalhando junto ao YUI, que era a base. Pelo fato do ExtJS ser uma extensão do YUI, o yui-ext não funcionava sem ele. (ROSA, 2012)

Como diz o autor acima, quando o yui-ext foi se desenvolvendo, foram sendo agrupados diversos adaptadores, que faziam o papel de ponte entre o yui-ext e outros *frameworks* JavaScript, como, por exemplo, o Prototype e o jQuery. Através

de seu crescimento e reconhecimento muito rápido em relação aos desenvolvedores, devido a uma extensão rica em funcionalidades e componentes de interface, adicionando importância aos mais conhecidos *frameworks* de Javascript (YUI, Prototype e jQuery), o seu desenvolvimento foi recebendo colaboradores e agilidade. Assim, sobrevieram várias versões 0.x do yui-ext, antes que fosse lançada a versão 1.0 final, na qual seu nome foi modificado para ExtJS, comprovando que o mesmo já não era mais apenas uma extensão para o YUI.

Segundo Rosa (2012), a seguinte release lançada foi o ExtJS 1.1 e a partir dessa versão ocorreu um salto transformando-o em um *framework* de JavaScript de verdade ou em *Ajax framework* e não mais uma limitada extensão para outros *frameworks*. No entanto ainda há a possibilidade de usá-lo assim.

A partir da versão 1.1.1 o ExtJS passou a ser cordialmente suportado pelos seguintes navegadores: Microsoft Internet Explorer 6+, Mozilla Firefox 1.5+, Apple Safari 2+ e Opera 9+ e após a série 1.1, começou a ser desenvolvido o ExtJS 2.0, apresentando várias funções e componentes básicos totalmente remodelados para trabalhar com mais eficácia e códigos reduzidos. (ROSA, 2012)

3.2.2. Arquitetura MVC no *Framework* ExtJS

Segundo Lamim (2012), a arquitetura MVC é baseada em três componentes, *model* (modelo), *view* (visão) e *controller* (controlador). O MVC, além de decompor a aplicação nos três tipos de componentes citados acima, determina, também, as interações entre eles, conforme é explicado a seguir.

Segundo o autor acima, um controlador pode mandar comandos para sua visão agregada para alterar a apresentação da visão do modelo, podendo, também, enviar comandos para o modelo para atualizar o estado do modelo. No caso de um modelo são informadas suas visões e seus controladores associados quando ocorre uma alteração em seu estado. Esta notificação possibilita que as visões gerem saídas atualizadas e que os controladores modifiquem o conjunto de comandos disponíveis. Uma implementação passiva do MVC monta estas notificações, pelo fato da aplicação não precisar delas ou, então, a plataforma de *software* não conseguir suportá-las. Já a visão requer da informação do modelo que ela precisa para gerar um perfil de saída.

A arquitetura MVC foi desenvolvida para computação pessoal, no entanto o MVC foi ajustado como uma arquitetura para as aplicações *web* em todas as linguagens de programação de maior utilização. Diversos *frameworks* de aplicação comerciais e não comerciais que conseguem utilizar esse modelo foram criados. Estes *frameworks* alteram em suas interpretações, principalmente no jeito que as responsabilidades MVC são divididas entre um cliente e servidor (LAMIM, 2012).

Diz o autor ainda que os *frameworks web* MVC mais recentes possuem uma abordagem de fina de cliente que disponibiliza, quase que totalmente, o modelo, a visão e a lógica do controlador no servidor. Nesta abordagem, o cliente envia requisições de *hiperlink* ou entrada de formulário ao controlador e, então, recebe uma página *web* completa e atualizada da visão. O modelo se localiza inteiro no servidor. Como as tecnologias de cliente amadureceram, os *frameworks* como JavaScript MVC e Backbone foram desenvolvidos permitindo que os componentes MVC executem parcialmente no cliente.

Segundo Lamim (2012), é possível descrever um caso prático de uma aplicação *web* no qual a *view* é um documento HTML definido pela aplicação. O controlador recebe uma entrada utilizando o método GET ou o método POST depois de um estímulo do utilizador e determina como processá-la, obtendo objetos do domínio para tratar a lógica de negócio e, por fim, obtendo uma visão para exibir a saída.

A seguir há uma figura para melhor compreensão do funcionamento da arquitetura MVC.

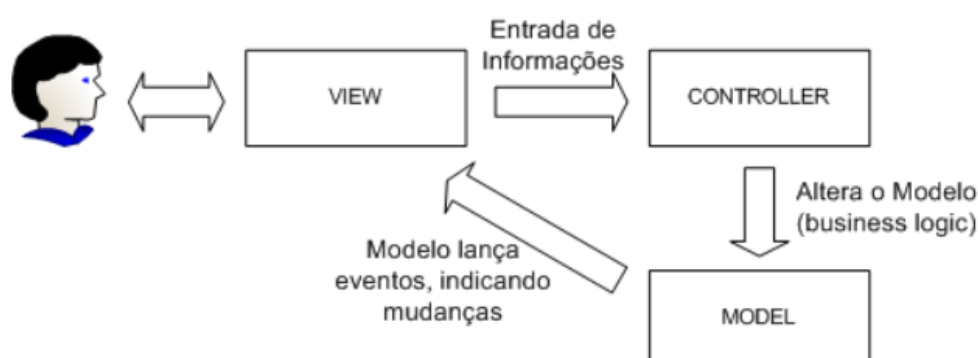


Figura 2 - Fluxo de Eventos e Informações em uma Arquitetura MVC

Fonte: ALMEIDA (2013)

3.2.3. Documentação do *Framework* ExtJS

Segundo Groner (2011), a documentação do ExtJS é considerada farta, existindo uma comunidade ativa que utiliza o ExtJS no Brasil, além da comunidade Norte-Americana. Mesmo com a documentação do *framework* em inglês, o entendimento é bastante fácil, além de ser bem dinâmica e apresentar diversos exemplos de suas funcionalidades. Há também algumas ferramentas de criação visual de componentes, como, por exemplo, formulários e plug-ins para algumas IDEs disponíveis gratuitamente.

Diz a autora ainda que é possível acessar a documentação tanto pelo site da Sencha como off-line, de uma máquina qualquer. Para conseguir acessar no modo off-line é necessário baixar a documentação em um computador, conforme passos exibidos abaixo.

- Entrar no site da Sencha através do link www.sencha.com



Figura 3 - Endereço do site Sencha

Fonte: Sencha (2013)

- Selecionar na aba *Products* o *framework* Sencha ExtJS

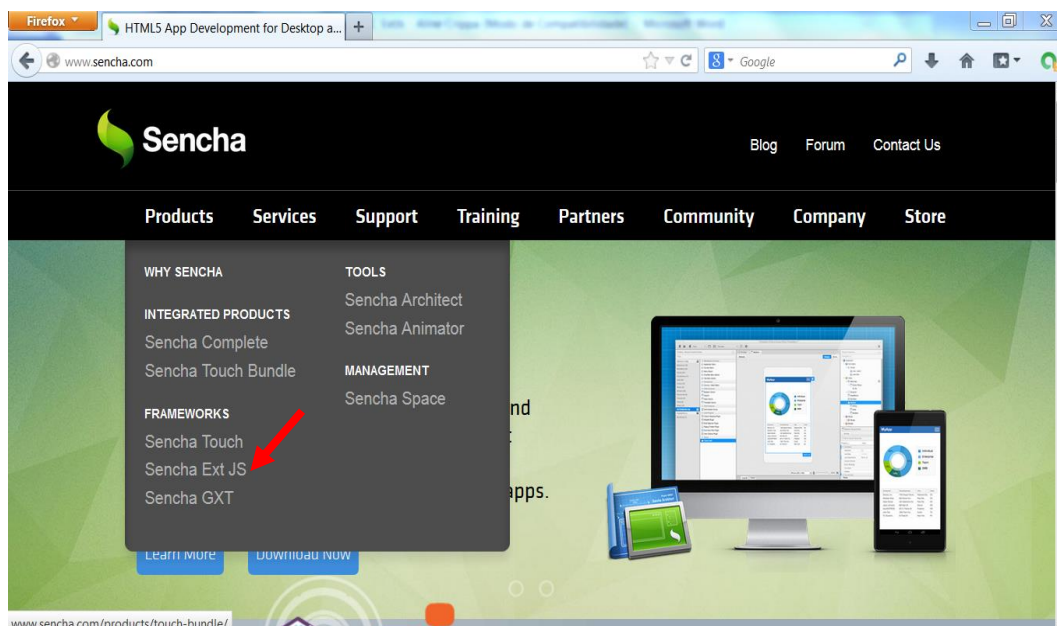


Figura 4 - Aba *Products* da Secha

Fonte: Sencha (2013)

- Clicar em *Download* para abrir a página com as opções de *download*

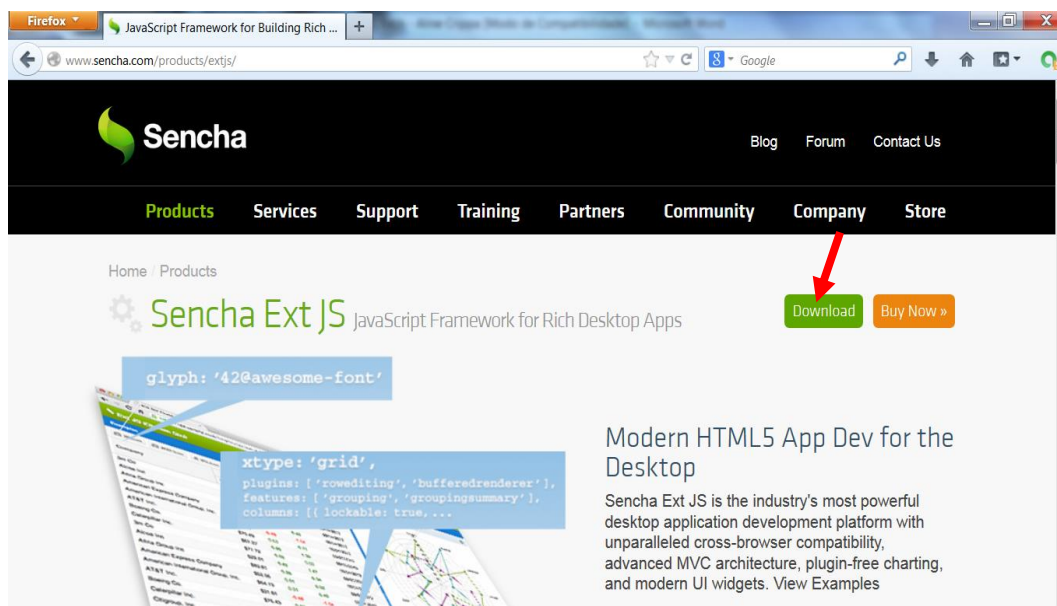


Figura 5 - Página para *download* do Sencha ExtJS

Fonte: Sencha (2013)

- Clicar em *Download*, no contexto *Working in Open Source?*, para baixar o ExtJS gratuito

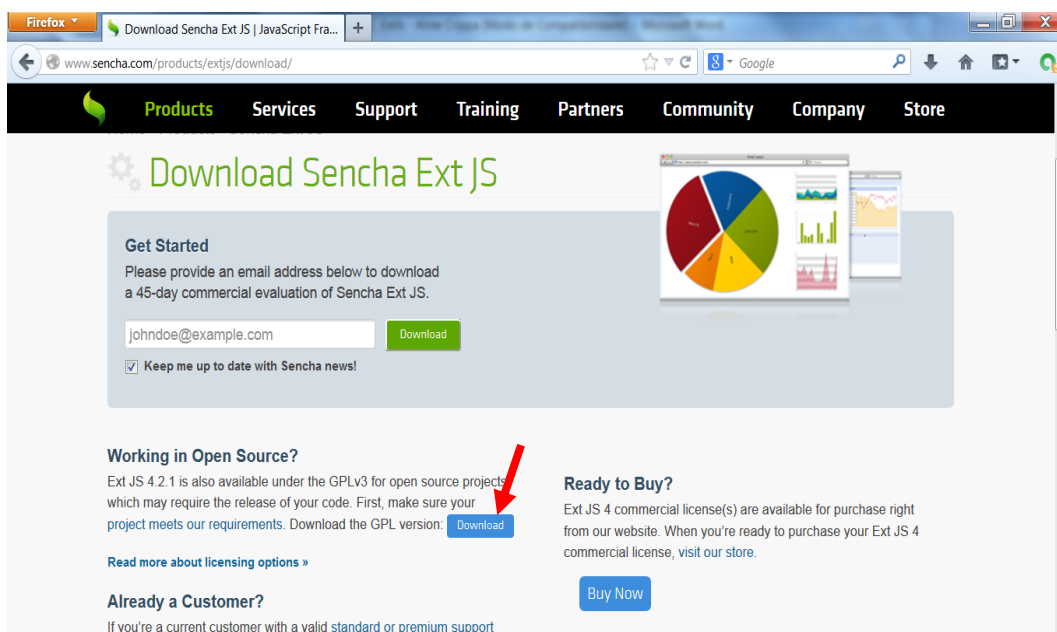


Figura 6 - *Download* do ExtJS

Fonte: Sencha (2013)

- Para concluir o *download*, clicar em *Download* e depois em OK.

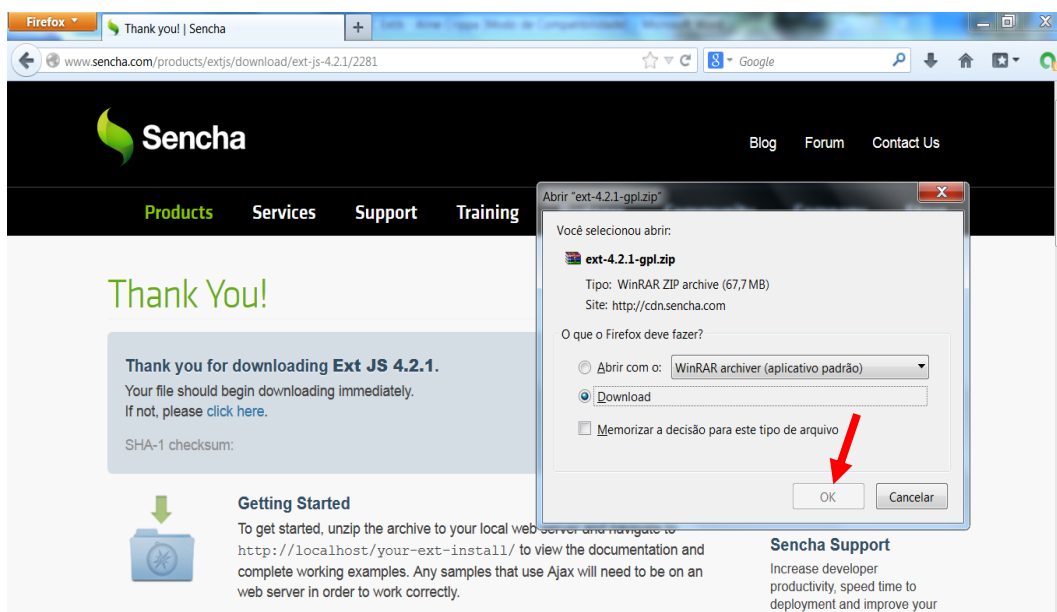


Figura 7 - Página de Conclusão do *download* do ExtJS

Fonte: Sencha (2013)

Após concluir o *download*, basta abrir o local em que foi salvo e descompactar a pasta do ExtJS. Depois disso já é possível visualizar todos os exemplos, itens e recursos oferecidos pela documentação do *framework* ExtJS (GRONER, 2011).

Diz a autora ainda que há, também, a possibilidade de utilizar a documentação diretamente no site, acessando o link <http://docs.sencha.com/extjs/4.2.2/>, como mostra figura a seguir.

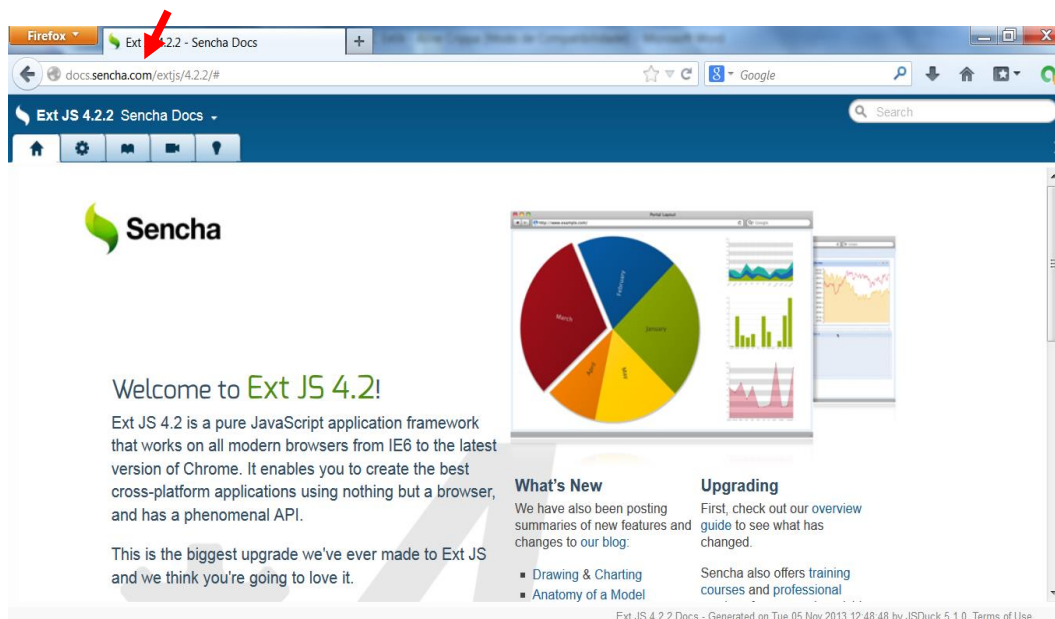


Figura 8 - Página inicial da documentação do ExtJS 4.2

Fonte: Sencha (2013)

Segundo Groner (2011), dentro da documentação do ExtJS pode-se encontrar exemplos de como utilizar os recursos e explicações sobre as funcionalidades desses recursos. As principais funcionalidades desta biblioteca são:

- **Dados:** fornece de maneira ágil, fácil e eficiente o intercâmbio de dados com os *scripts* php, asp, java, etc, através do JSON, mas também trabalha bem com arquivos XML.
- **Drag & Drop:** uma excelente funcionalidade do ExtJS é o arrastar e soltar (*drag & drop* ou DD). Através desse componente, o desenvolvedor pode criar instruções que permitam arrastar e soltar elementos nas páginas *web*, inclusive gerando alterações em dados ou outras ações, de modo fácil.
- **Formulários:** sendo considerado um ponto forte do ExtJS, representa facilidade e elegância nos formulários que podem ser criados. Possibilita

arrastar itens de uma tabela de dados e estes dados surgem em campos de um formulário, por exemplo.

- *Grid*: é possível criar diversos tipos de *grid* ou tabelas de dados, podendo editar dados diretamente, ordenar, mover colunas, e diversas outras funcionalidades interessantes de forma elegante e eficiente.
- *Layout*: é possível criar *layouts* elegantes, bonitos e práticos. A criação de painéis, janelas, abas, entre outros, pode ocorrer dinamicamente, além de ser fácil de implementar os temas, uma vez que o visual fica a cargo de arquivos CSS.
- Menus: além de bonitos e elegantes, os menus de diversos tipos podem ser criados e integrados em diversos elementos.
- *Tree*: uma das melhores funcionalidades do ExtJS é a criação de árvores, pois, além de servirem de menus hierárquicos, podem servir de classificadores de dados, exibição de arquivos e pastas e diversas outras funcionalidades.

Além dos componentes citados anteriormente, o ExtJS conta com componentes que permitem redimensionar os elementos, paginação de dados, abas, barras de progresso, *QuickTips*, barras de *status*, *splitbars*, *tooltips* e *toolbars*, *windows* entre outros (GRONER, 2011).

3.2.4. Versões do *Framework* ExtJS

Segundo Groner (2011) versão 2.0 do ExtJS foi lançada em 4 de dezembro de 2007, sendo originada como provimento de uma interface e recursos semelhantes aos tradicionalmente agregados com aplicações desktop. Como a versão 2.0 não foi compatível com a 1.1 foi desenvolvido um guia de migração para resolução desse problema.

Diz a autora acima que a versão 3.0 do ExtJS foi lançada em 6 de julho de 2009, adicionando suporte de comunicação ao REST e uma nova plataforma Ext.Direct do lado servidor. Novos componentes gráficos e elementos foram incluídos aos componentes de visualização padrão. O *framework* era compatível com o código de versão 2.0.

A versão 4.0 do *framework* ExtJS foi lançada no dia 26 de abril de 2011, incluindo uma estrutura de classe totalmente revista, um pacote de dados revisto, um pacote de animação e desenho que usa SVG e VML e revisto gráficos e temas. O ExtJS também inclui uma arquitetura opcional que fornece um modelo-visão-controlador de estilo (MVC) de organização de código (GRONER, 2011).

3.2.5. Vantagens e Desvantagens de Utilizar o ExtJS

Segundo MANTOAN (2010), o ExtJS possui algumas vantagens e desvantagens ao ser utilizado, por isso é necessário verificar quais os requisitos e funcionalidades que serão utilizados e validar se deve usá-lo no desenvolvimento de determinado sistema. Podem ser listadas algumas dessas vantagens e desvantagens, conforme exibido em tabela a seguir.

Tabela 1 - Vantagens e Desvantagens do ExtJS

Vantagens	Desvantagens
Melhor <i>Framework</i> para grid	Burocracia para estender ou usar os componentes
Documentação farta	Comunidade um pouco rude
Criação de interfaces funcionais, eficientes, elegantes	<i>Framework</i> complexo para desenvolvimento de sistemas simples
<i>IDE</i> Própria	Sintaxe mais extensa que do JQuery
Usabilidade Simples	

Fonte: Próprio Autor (2013)

4. DESENVOLVENDO APLICAÇÃO WEB COM *FRAMEWORK* EXTJS

Este capítulo é uma abordagem prática sobre o tema discutido neste trabalho, no qual será simulado um sistema *web* de agendamento simples de um salão de beleza utilizando o *framework* ExtJS para facilitar o desenvolvimento deste agendamento.

4.1. Requisitos e Ferramentas Utilizadas para o Desenvolvimento do Projeto

Os requisitos deste projeto são basicamente o agendamento de um horário via *web* e o cadastro de novos clientes. Para que sejam atendidos esses requisitos, será necessário o uso de um banco de dados que armazene as informações dos clientes e da agenda do salão e integre essas informações quando necessário.

O sistema funcionará do seguinte modo: o cliente marca um horário a primeira vez conforme o método tradicional, através de uma ligação ao salão, e, quando o cliente comparece no horário e na data marcados, é criado um cadastro com seus dados e disponibilizado para ele um login e uma senha que o permitam acessar o sistema *web* do salão. Com esse login e senha, o usuário pode acessar o sistema e agendar um horário e os serviços que desejar, nas datas e horários que estiverem disponíveis, pelo próprio sistema, sem ser necessário entrar em contato com o salão.

Para o desenvolvimento do projeto prático foram utilizadas as seguintes ferramentas:

- Sublime Text: um editor de textos utilizado para realizar a programação do sistema de agendamento. A programação utiliza basicamente a linguagem php para o desenvolvimento do sistema *web*.
- PgAdmin: um gerenciador de banco de dados que permite criar, gerenciar e excluir dados e informações necessárias para o desenvolvimento do sistema utilizando o Postgres como linguagem de banco de dados.
- XAMPP: um pacote com servidores de código aberto que possibilita executar sistemas localmente facilitando e agilizando o desenvolvimento do sistema.
- *Framework* ExtJS: um *framework* que auxilia e facilita o desenvolvimento dos códigos do sistema. Nesse projeto foi utilizada a versão 4.2.2.

- Mozilla Firefox: um navegador que possibilita visualizar e testar o sistema de agendamento desenvolvido.
- Firebug: um complemento do Mozilla Firefox que possibilita encontrar erros durante e após o desenvolvimento do sistema, enquanto estiver sendo executado e a versão utilizada no projeto foi a 1.12.4.
- Astah Community: uma ferramenta utilizada para desenvolver os diagramas e o caso de uso relacionados ao sistema de agendamento.

4.2. Documentação do Sistema Desenvolvido

4.2.1. MER (Modelo Entidade Relacionamento)

Na representação a seguir foi descrito apenas parte do MER, com algumas tabelas do banco de dados utilizadas, para demonstrar a possibilidade de se realizar uma boa documentação do sistema de Agendamento.

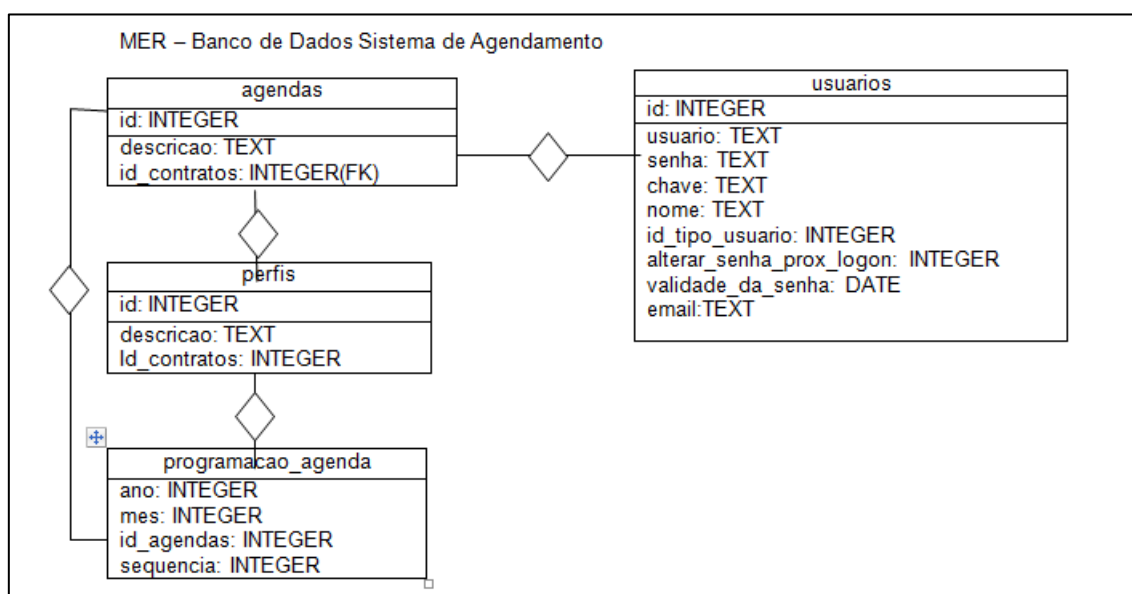


Figura 9 - MER Banco de Dados Sistema de Agendamento

Fonte: Próprio Autor (2013)

4.2.2. Caso de Uso do Sistema de Agendamento

O caso de uso do projeto desenvolvido mostra as principais funcionalidades e a relação entre os atores e o sistema. O Atendente possui um perfil administrativo com permissão de acesso ao sistema no qual pode administrar as contas dos clientes e a agenda, podendo alterar dias e serviços solicitados pelos clientes. Este ator também consegue realizar as mesmas funcionalidades disponibilizadas para os

clientes. O Cliente tem um perfil restrito, que lhe exibe apenas as informações de seu login. Este ator tem acesso à agenda apenas para marcar e desmarcar horário e escolher serviços disponíveis em sua visualização. Ambos os atores precisam ter cadastro para conseguirem acessar o sistema.

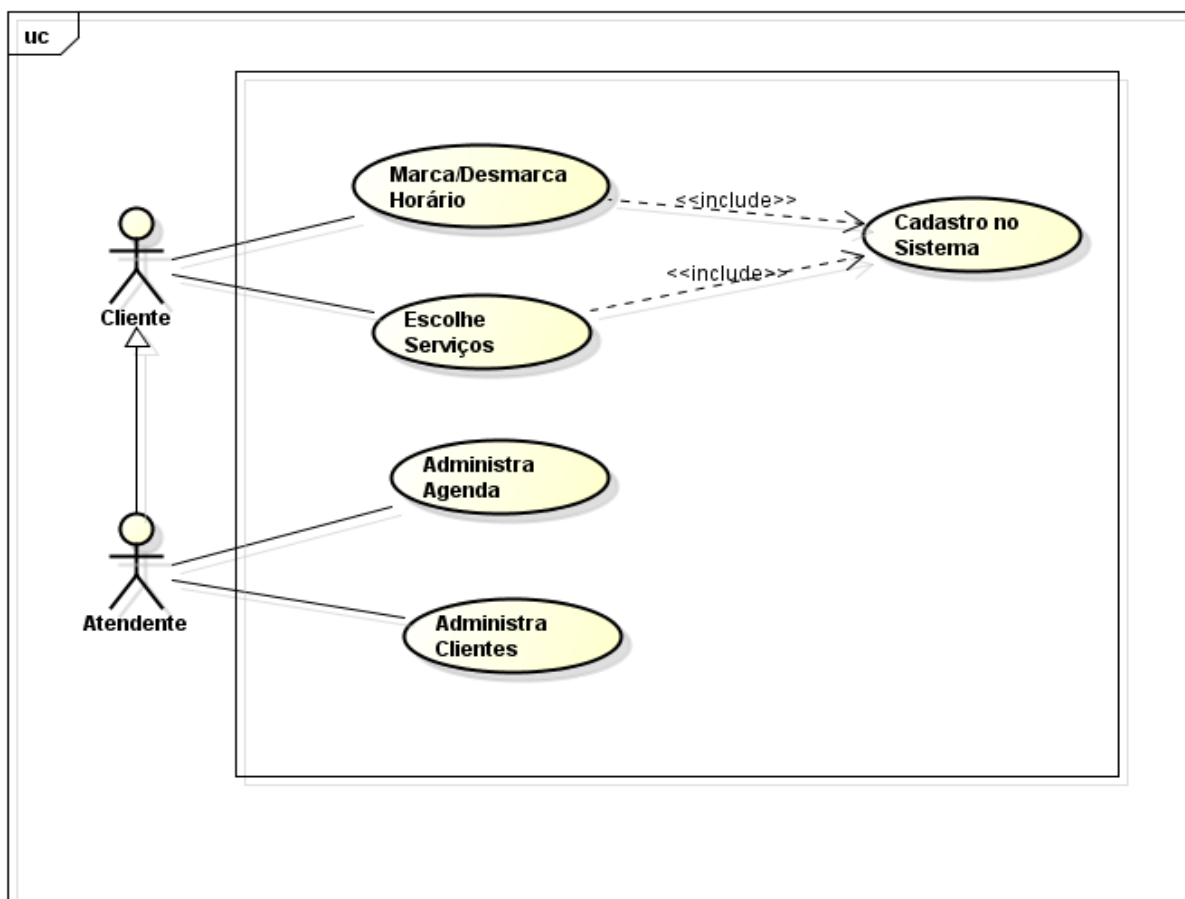


Figura 10 - Caso de Uso do Sistema de Agendamento para Salão de Beleza

Fonte: Próprio Autor (2013)

4.2.3. Diagramas de Atividade das Principais Funcionalidades do Sistema

Os diagramas de atividades têm como objetivo demonstrar como são utilizadas as principais funcionalidades do sistema de Agendamento.

O diagrama de atividade representado a seguir é utilizado para que, através do atendente, seja criada uma conta para que o cliente possa acessar o sistema de agendamento. Para que isso ocorra o atendente deve acessar o sistema, abrir o cadastrar clientes, criar um novo usuário, inserir as informações solicitadas para o usuário no sistema e, assim, o sistema registrará o novo cadastro. O cliente necessita apenas informar os dados necessários para o atendente e ao término do cadastro, receberá um login e senha para acessar o sistema quando necessitar.

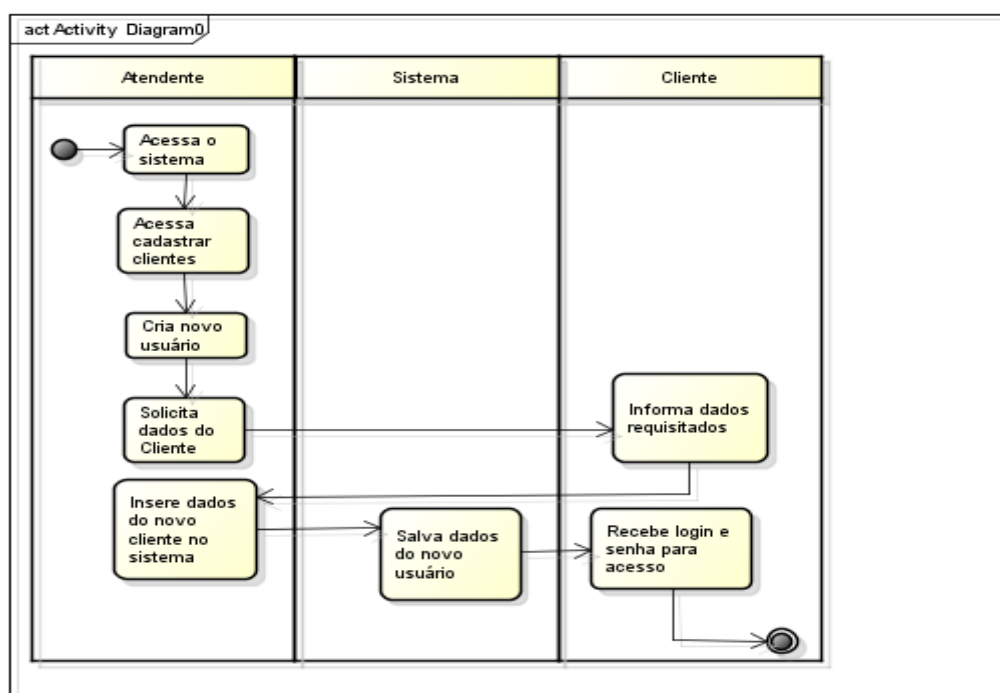


Figura 11 - Diagrama de Atividade Cadastrar Novo Cliente no Sistema

Fonte: Próprio Autor (2013)

O diagrama de atividade Marcar Horário está representado abaixo para melhor compreensão do funcionamento da agenda. Com o login e senha, o cliente acessa o sistema, entra em agenda e seleciona os serviços, horário e data desejados. É, então, reservada a data escolhida e enviada uma solicitação para que o atendente confirme a mesma com o cliente que reservou a data. Caso seja confirmado o agendamento, a data é reservada e caso contrário o atendente

desmarca a reserva da data e disponibilizada novamente na agenda para os clientes.

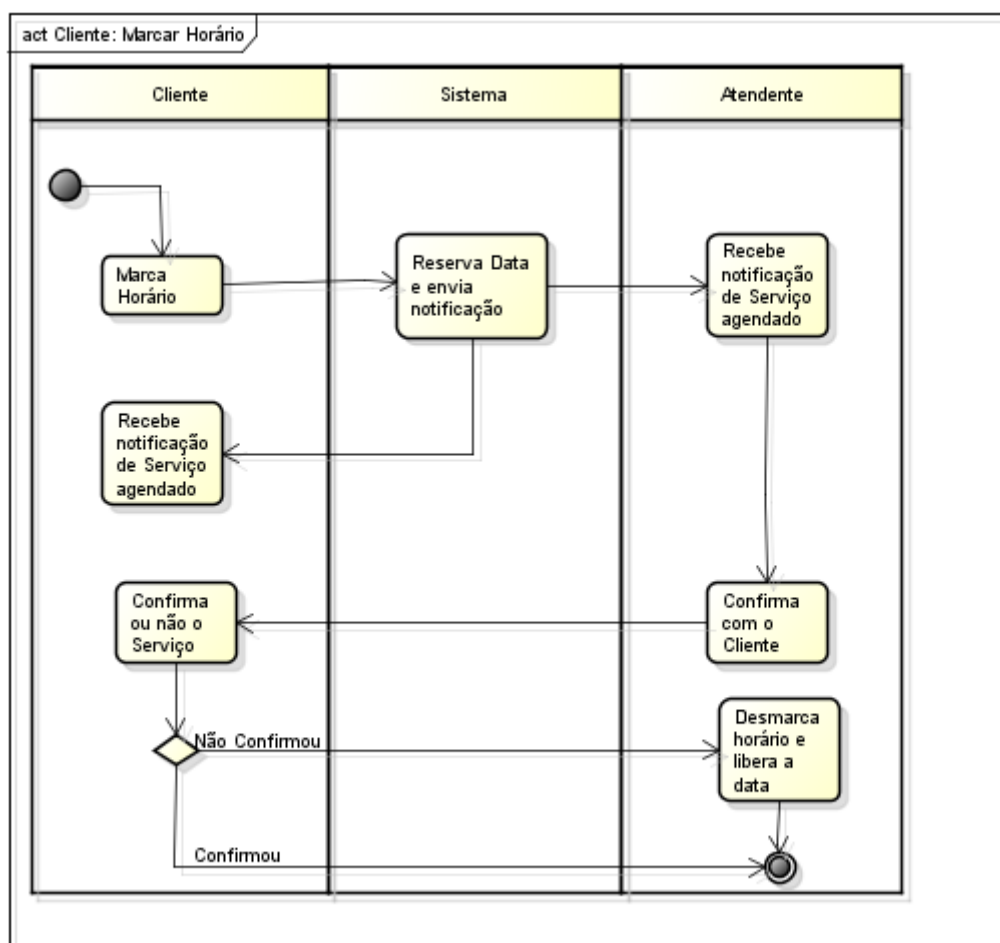


Figura 12 - Diagrama de Atividade Agendar Horário pelo Sistema

Fonte: Próprio Autor (2013)

4.3. Programação e Desenvolvimento

Os principais códigos de programação utilizados para desenvolvimento do sistema estão nomeados com suas funcionalidades, a seguir.

index.php: abre a tela de login do sistema, para que o usuário possa acessá-lo.

login.php: compara o usuário e senha digitados na tela de login, verificando com o banco de dados se existe esse usuário e essa senha, fazendo a validação e retornando esse resultado para o **index.php**.

sistema.php: funciona como uma interface que chama as principais funcionalidades do sistema, relacionando todos os códigos do mesmo.

funcoes.js: cria as principais funções do sistema utilizando o ExtJS e retornam os resultados para o **sistema.php**.

actions.php: armazena as ações que são utilizadas e chamadas em todo o sistema como, por exemplo, a ação de salvar um novo usuário ou a alteração de um usuário existente.

agenda.php: realiza a conexão entre o sistema e o banco de dados, para que seja possível a troca de informações.

Além desses códigos principais, foram utilizados diversos outros códigos para o funcionamento do sistema como, por exemplo, para inserir, remover e atualizar dados disponibilizados no sistema.

Para melhor entendimento da utilização do ExtJS em um sistema, será exibido um anexo com parte trecho do código de **funcoes.js**.

4.4. Layout do Sistema de Agendamento

As figuras exibidas a seguir mostram o layout desenvolvido na parte prática do projeto, sendo que algumas telas têm diferentes permissões e conteúdos de acordo com o usuário logado na agenda do sistema do salão, podendo ser atendente ou cliente.



Figura 13 - Tela de Login do Sistema

Fonte: Próprio Autor (2013)



Figura 14 - Tela de Menu do Sistema

Fonte: Próprio Autor (2013)

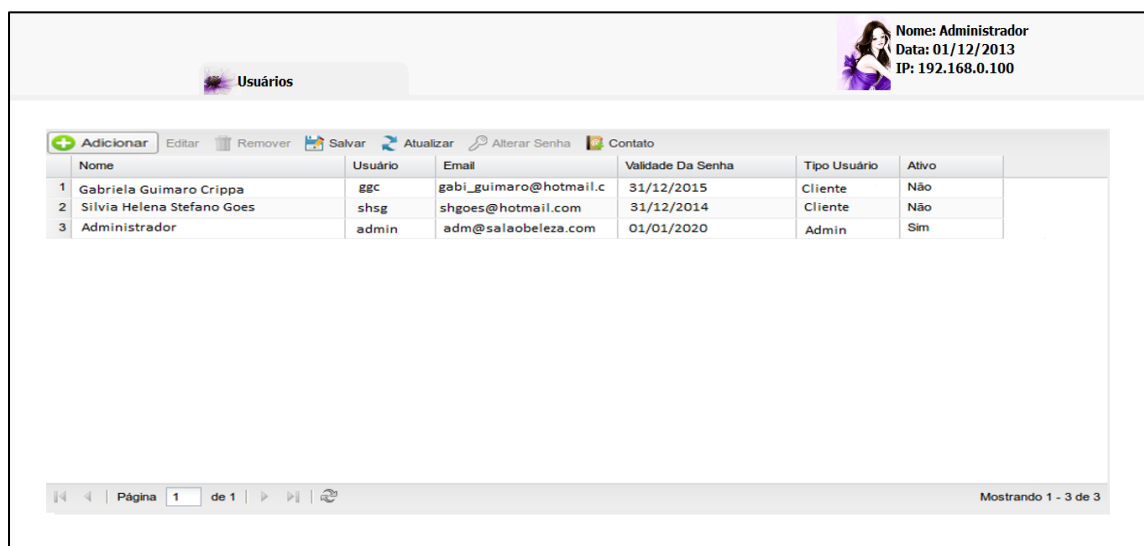


Figura 15 - Tela de Cadastro de Usuários - Atendente

Fonte: Próprio Autor (2013)

Usuários

Nome: Gabriela G. Crippa
Data: 01/12/2013
IP: 192.168.0.100

	Nome	Usuário	Email	Validade Da Senha
1	Gabriela Guimaro Crippa	ggc	gabi_guimaro@hotmail.c	31/12/2015

Página 1 de 1 | Mostrando 1 - 3 de 3

Figura 16 - Tela de Cadastro de Usuários - Cliente

Fonte: Próprio Autor (2013)

Agenda

Nome: Administrador
Data: 01/12/2013
IP: 192.168.0.100

1 Ext JS Calendar | Event Unhas - Mão e Pé - Sílvia H. S. Goes was updated

December 1, 2013 - January 4, 2014

Sun	Mon	Tue	Wed	Thu	Fri	Sat
Today 5:17 pm		12:00pm Almoço				12:00pm Almoço
Fechado		13:00pm Almoço				12:00pm Almoço
Fechado		12:00pm Almoço				12:00pm Almoço
Fechado		12:00pm Almoço				12:00pm Almoço
Fechado	7:00pm Tintura - Sandra ... 12:00pm Almoço 6:30pm Corte com Secag...	Fechado		12:00pm Almoço		12:00pm Almoço 12:45pm Tintura - Gabrie... 3:15pm Corte com Secag...
Fechado	12:00pm Almoço 12:45pm Limpeza de Pele. 7:00pm Unhas - Mão e Pé.	Fechado		12:00pm Almoço		

Figura 17 - Tela de Agenda Calendário – Atendente

Fonte: Próprio Autor (2013)

Nome: Gabriela G. Crippa
Data: 01/12/2013
IP: 192.168.0.100

Agenda

1 Ext JS Calendar

December 1, 2013 - January 4, 2014

S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	1	2	3	4
5	6	7	8	9	10	11

Today

Sun	Mon	Tue	Wed	Thu	Fri	Sat
Today 5:11 pm Fechado	2 12:00pm Almoço	3 12:00pm Almoço	4 12:00pm Almoço	5 12:00pm Almoço	6 12:00pm Almoço	7 12:00pm Almoço
8 Fechado	9 12:00pm Almoço	10 12:00pm Almoço	11 12:00pm Almoço	12 12:00pm Almoço	13 12:00pm Almoço	14 12:00pm Almoço
15 Fechado	16 12:00pm Almoço	17 12:00pm Almoço	18 12:00pm Almoço	19 12:00pm Almoço	20 12:00pm Almoço	21 12:00pm Almoço
22 Fechado	23 7:00am Horário Indisponi... 12:00pm Almoço 6:30pm Horário Indisponi...	24 Fechado	25 12:00pm Almoço	26 12:00pm Almoço	27 12:00pm Almoço	28 12:00pm Almoço 12:45pm Horário Agenda... 3:15pm Horário Agendad...
29 Fechado	30 12:00pm Almoço 12:45pm Horário Indispo... 7:00pm Horário Indisponi...	31 Fechado	Jan 1 12:00pm Almoço	2 12:00pm Almoço	3 12:00pm Almoço	4 12:00pm Almoço

Figura 18 - Tela de Agenda Calendário - Cliente

Fonte: Próprio Autor (2013)

5. CONSIDERAÇÕES FINAIS

Neste projeto foi possível visualizar a complexidade de um desenvolvimento de sistemas, através do auxílio do *framework* ExtJS para a criação de componentes e recursos que o sistema oferece.

Esse *framework* foi utilizado com o intuito de facilitar a criação de componentes mais complexos como uma agenda e que exigem maior tempo para serem construídos. Assim, através do ExtJS, consegui otimizar o tempo do desenvolvimento, além de padronizar o *layout* do sistema.

Com esse trabalho, pode-se perceber a possibilidade de integrar o desenvolvimento com vários *frameworks* de funções distintas.

Foram utilizadas, também, várias ferramentas como, por exemplo, a de programação e o gerenciador de banco de dados. Essas ferramentas utilizadas, citadas ao longo deste trabalho, auxiliaram no desenvolvimento do projeto prático.

As principais vantagens em se utilizar esse *framework* para a criação de interfaces são a fácil implementação, a excelente documentação para tirar dúvidas ou realizar consultas, além de disponibilizar exemplos de aplicações com o ExtJS, possuir uma IDE própria com diversos recursos visuais, facilitando o desenvolvimento, e seus recursos como *Grids* e *Tree* que são excelentes.

A arquitetura MVC do ExtJS facilita a sua interação com sistemas *Web*, uma vez que a arquitetura utilizada na *web* é a MVC.

No projeto prático foi desenvolvido um projeto de um sistema de agendamento simples, sem agregar muitos recursos e tecnologias, apenas para demonstrar a facilidade e usabilidade dos recursos do *framework* em um desenvolvimento mais complexo.

Portanto, a sugestão para continuação deste projeto é programar e aperfeiçoar o sistema, incluindo novas tecnologias disponíveis, novos recursos e funcionalidades, além da possibilidade de agendar e efetuar o pagamento dos serviços pelo próprio sistema.

REFERÊNCIAS

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **Citação:** NBR-10520/ago - 2002. Rio de Janeiro: ABNT, 2002.

_____. **Referências:** NBR-6023/ago. 2002. Rio de Janeiro: ABNT, 2002.

ALMEIDA, R. R. de. **Model View Controller (MVC)**. Disponível em: <<http://www.dsc.ufcg.edu.br/~jacques/cursos/map/html/arqu/mvc/mvc.htm>>. Acesso em: 12 Set 2013, 21h.

EDUCAÇÃO, P. **As Vantagens de um Sistema de Informação**. Disponível em: <<http://www.portaleducacao.com.br/informatica/artigos/43037/as-vantagens-de-um-sistema-de-informacao>>. Acesso em: 22 Set 2013, 15h37.

GONÇALVES, L. S. **Sistema de Informação**. Disponível em: <<http://arquivoscdn.portalava.com.br/videos/videolivriaria/pdfs/6519.pdf>>. Acesso em: 25 Set 2013, 20h.

GRONER, L. **Curso ExtJS 4**. Disponível em: <<http://www.loiane.com/2011/11/curso-de-ExtJS-4-gratuito/>>. Acesso em: 07 Mar 2013

GRONER, L. **Ext JS 4 First Look**. Birmingham, Uk: Packt Publishing, 2011.

GUANAIS, K. **Aplicações Ricas de Internet**. Disponível em: <<http://www3.iesampa.edu.br/ojs/index.php/sistemas/article/view/544>>. Acesso em: 20 Out 2013, 21h42.

FRAZATTO, J. **Vantagens do Sistema de Informação nas Empresas**. Disponível em: <<http://josyfazatto.blogspot.com.br/2010/11/vantagens-do-sistema-de-informacao-nas.html>>. Acesso em: 13 Out 2013, 22h.

FURLAN, J. D., IVO, I. da M., AMARAL, F. P. **Sistema de Informação Executiva EIS - Executive Information System: Como Integrar os Executivos ao Sistema Informacional das Empresas, Fornecendo Informações Úteis e Objetivas para suas Necessidades Estratégicas e Operacionais**. São Paulo: Makron Books, 1994.

LAMIM, J. **MVC – O Padrão de Arquitetura de Software**. Disponível em: <<http://www.oficinadanet.com.br/artigo/1687/>>. Acesso em: 15 Out 2013, 22h45.

LAUDON, K. C., LAUDON, J. P. **Sistema da Informação com Internet**. Rio de Janeiro; LTC, 1999.

MANTOAN, F. **Frameworks para Rich Internet Applications**. Disponível em: <<http://fernandomantoan.com/tag/extjs/>>. Acesso em: 20 Set 2013, 22h25.

MATTOS, R. **Linguagem de Programação**. Disponível em: <<http://www.linhadecodigo.com.br/artigo/489/linguagem-de-programacao.aspx>>. Acesso em: 20 Set 2013, 21h45.

NET, O. da. **O que é um Servidor Web?** Disponível em: <http://www.oficinadanet.com.br/artigo/servidores/o_que_e_um_servidor_web>. Acesso em: 20 Out 2013, 21h40.

PALMEIRA, T. V. V. **Como Funcionam as Aplicações Web.** Disponível em: <<http://www.devmedia.com.br/como-funcionam-as-aplicacoes-web/25888>>. Acesso em: 15 Out 2013, 10h45.

ROSA, E. da. **ExtJS: Um Excelente Framework de JavaScript.** Disponível em: <<http://codigofonte.uol.com.br/artigos/extjs-um-excelente-framework-de-javascript>>. Acesso em: 15 Out 2013, 17h15.

REZENDE, R. **Conceitos Fundamentais de Banco de dados.** Disponível em: <http://www.sqlmagazine.com.br/colunistas/ricardorezende/02_conceitosbd.asp>. Acesso em: 17 Set 2013, 15h10.

SAUVÉ, J. P. **Frameworks: O que é um Framework?** Disponível em: <<http://www.dsc.ufcg.edu.br/~jacques/cursos/map/html/frame/oque.htm>>. Acesso em: 24 Out 2013, 20h17.

SAUVÉ, J. P. **Tipos de Framework.** Disponível em: <<http://www.dsc.ufcg.edu.br/~jacques/cursos/map/html/frame/tipos.htm>>. Acesso em: 24 Out 2013, 20h45.

SENCHA. Inc. **Sencha.** Disponível em: <<http://www.sencha.com>>. Acesso em: 27 Out 2013, 15h.

SILVA, E. L. da **Metodologia da pesquisa e elaboração de dissertação.** 3ª ed. Florianópolis: Laboratório de Ensino à Distância da UFSC, 2001.

STAIR e REYNOLDS, R. M. **Princípios de Sistemas de Informação.** Tradução da 6ª ed., 2006.

TONI, J. A. **A Evolução dos Sistemas de Informação.** Disponível em: <<http://www.e-combr.com.br/artigos/19-a-evolucao-dos-sistemas-de-informacao.html>>. Acesso em: 12 Out 2013

WATERS, C. **Web Concepção & Design.** Editora Quark, 1999.

ANEXO – Código de Parte da Programação do Sistema de Agendamento

```
/*Usuários Perfis*/
Ext.define('public.usuarios_perfis', {
    extend: 'Ext.data.Model',
    fields: [
        { name: 'id', type:'int'},
        { name: 'descricao'},
        { name: 'selected'}
    ]
});
store['public.usuarios_perfis'] = Ext.create('Ext.data.Store', {
    autoDestroy: true,
    id: 'store-public.usuarios_perfis',
    autoLoad: false,
    autoScroll: true,
    remoteSort: true,
    storeId: 'id-public.usuarios_perfis',
    sorters: [{ property: 'descricao', direction: 'ASC'}],
    model: 'public.usuarios_perfis',
    proxy: {
        type: 'ajax',
        api: {
            read: 'public/comandos/read_perfis.php'
        },
        actionMethods: {read: 'POST', update: 'POST'},
        reader: {
            type: 'json',
            root: 'data',
            totalProperty: 'total',
            successProperty: 'success',
            messageProperty: 'msg'
        },
        writer: {
```

```

        type: 'json',
        root: 'data',
        writeAllFields: true,
        encode: true
    }
}
});
store['public.usuarios_perfis'].addListener('load',function(ds,regs,ops) { //regs =
ds.data.items; ops = ds.lastOptions
    if (ds.getTotalCount() > 0) {
        for(i=0;i<ds.getTotalCount();i++) {
            if(regs[i].data.selected == 1){

                grid['public.usuarios_perfis'].getSelectionModel().select(i,true);
            }
        }
    }
});
function crudPublicUsuariosPerfis(field,id) {
    grid['public.usuarios_perfis'] = Ext.create('Ext.grid.Panel', {
        store: store['public.usuarios_perfis'],
        selType: 'checkboxmodel',
        columns: [
            {
                header: 'Id',
                hidden: true,
                dataIndex: 'id'
            },
            {
                header: 'Perfis',
                xtype: 'gridcolumn',
                dataIndex: 'descricao',
                flex: 4,
                width: 'auto',

```

```

        minWidth: 150
    },
    {
        xtype: 'checkcolumn',
        header: 'Permissão',
        dataIndex: 'checked',
        hidden: true,
        editor: {
            xtype: 'checkbox',
            cls: 'x-grid-checkheader-editor'
        }
    }
],
renderTo: 'cadastro',
stripeRows : true ,
border: false,
width: '100%',
layout: 'fit',
multiSelect: true,
singleSelect: false,
columnLines: true,
viewConfig: { forceFit: true },
frame: true,
height: 400,
bbar: Ext.create('Ext.PagingToolbar', {
    store: store['public.usuarios_perfis'],
    displayInfo: true,
    displayMsg: 'Mostrando {0} - {1} de {2}',
    emptyMsg: 'Nenhum registro encontrado.'
}),

listeners: {
    'beforerender': function() {
        store['public.usuarios'].load();
    }
}

```



```

        id_usuarios:
Ext.getCmp('id_usuario').getValue(),
        id_perfis: Ext.JSON.encode(id_perfis)
    },
    success: function(response, opts){
        response =
Ext.decode(response.responseText);
        Ext.MessageBox.alert('Aviso',
response.msg);
    },
    failure: function (response) {
        response =
Ext.decode(response.responseText);
        Ext.MessageBox.alert('Aviso',
response.msg);
    }
});
}
}
});
}

```