

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
CÂMPUS PATO BRANCO
CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE
SISTEMAS**

**CICERO RALDI DA FONSECA
FAUSTO DANIEL ROSSI**

**SISTEMA WEB PARA GERENCIAMENTO DE EMPRESAS PRODUTORAS DE
EVENTOS**

TRABALHO DE CONCLUSÃO DE CURSO

**PATO BRANCO – PR
2013**

**CICERO RADI DA FONSECA
FAUSTO DANIEL ROSSI**

**SISTEMA PARA GERENCIAMENTO DE EMPRESAS PRODUTORAS DE
EVENTOS**

Trabalho de Conclusão de Curso de graduação, apresentado a disciplina de Trabalho de Diplomação, do curso superior de Análise e Desenvolvimento de Sistemas, da Universidade Tecnológica Federal do Paraná – Câmpus Pato Branco, como requisito para obtenção do título de Tecnólogo.

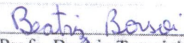
Orientador: Profa. Beatriz T. Borsoi

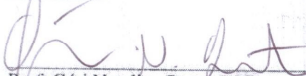
**PATO BRANCO – PR
2013**

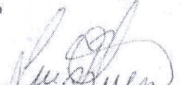
ATA Nº: 215


DEFESA PÚBLICA DO TRABALHO DE DIPLOMAÇÃO DO ALUNO CICERO RALDI DA FONSECA.

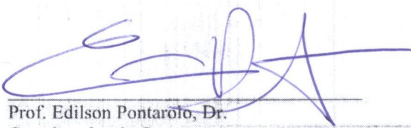
Às 16:08 hrs do dia 18 de setembro de 2013, Bloco V da UTFPR, Câmpus Pato Branco, reuniu-se a banca avaliadora composta pelos professores Beatriz Terezinha Borsoi (Orientadora), Géri Natalino Dutra (Convidado) e Luís Carlos Ferreira Bueno (Convidado), para avaliar o Trabalho de Diplomação do aluno Cicero Raldi da Fonseca, matrícula 1116630, sob o título **Desenvolvimento de Sistema para Empresas Produtoras de Eventos**, como requisito final para a conclusão da disciplina Trabalho de Diplomação do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, COADS. Após a apresentação o candidato foi entrevistado pela banca examinadora, e a palavra foi aberta ao público. Em seguida, a banca reuniu-se para deliberar considerando o trabalho **APROVADO**. Às 16:38 hrs foi encerrada a sessão.


Prof. Beatriz Terezinha Borsoi, Dr.
Orientadora


Prof. Géri Natalino Dutra, M.Sc.
Convidado


Prof. Luís Carlos Ferreira Bueno, M.Sc.
Convidado

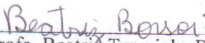

Prof. Eliane Maria De Bortoli Fávero, M.Sc.
Coordenador do Trabalho de Diplomação


Prof. Edilson Pontarolo, Dr.
Coordenador do Curso

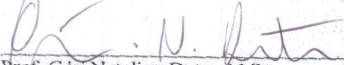
ATA Nº: 216

DEFESA PÚBLICA DO TRABALHO DE DIPLOMAÇÃO DO ALUNO FAUSTO DANIEL ROSSI.

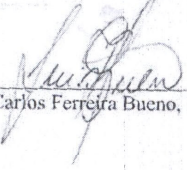
Às 16:08 hrs do dia 18 de setembro de 2013, Bloco V da UTFPR, Câmpus Pato Branco, reuniu-se a banca avaliadora composta pelos professores Robison Cris Brito (Orientador), Gêri Natalino Dutra (Convidado) e Luis Carlos Ferreira Bueno (Convidado), para avaliar o Trabalho de Diplomação do aluno Fausto Daniel Rossi, matrícula 1120280, sob o título **Desenvolvimento de Sistema para Empresas Produtoras de Eventos**; como requisito final para a conclusão da disciplina Trabalho de Diplomação do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, COADS. Após a apresentação o candidato foi entrevistado pela banca examinadora, e a palavra foi aberta ao público. Em seguida, a banca reuniu-se para deliberar considerando o trabalho **APROVADO**. Às 16:38 hrs foi encerrada a sessão.



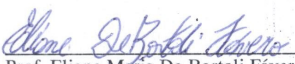
Profa. Beatriz Terezinha Borsoi, Dr.
Orientadora



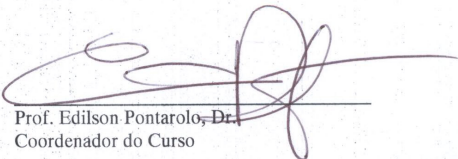
Prof. Gêri Natalino Dutra, M.Sc.
Convidado



Prof. Luis Carlos Ferreira Bueno, M.Sc.
Convidado



Prof. Eliane Maria De Bortoli Fávero, M.Sc.
Coordenadora do Trabalho de Diplomação



Prof. Edilson Pontarolo, Dr.
Coordenador do Curso

RESUMO

FONSECA, Cicero Raldi, ROSSI, Fausto Daniel. Sistema para gerenciamento de empresas produtoras de eventos. 2013. 64f. Monografia de Trabalho de Conclusão de Curso - Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, Universidade Tecnológica Federal do Paraná. Pato Branco, 2013.

A organização de um evento demanda muita informação. Para gerenciar um evento social são necessários contrato entre cliente e empresa organizadora, boletos para pagamento das parcelas e protocolo do evento, dentre outros. A Empresa produtora de eventos (EPE) é, muitas vezes, responsável por gerenciar o evento por completo, tendo assim que armazenar diversas informações sobre o evento, os clientes, os locais de realização do mesmo e demais informações agregadas. Devido à quantidade de informação a ser armazenada, verificou-se a necessidade de um software de gerenciamento de EPE. Esse gerenciamento se refere à possibilidade da EPE poder armazenar todas as informações referentes ao evento em uma base de dados centralizada, com segurança, agilidade na gravação e busca pelas informações desejadas. Este trabalho refere-se ao desenvolvimento de um sistema *web* para gerenciamento de empresas produtoras de eventos. A implementação foi realizada utilizando a ferramenta Microsoft® Visual Studio 2012 Express, Microsoft® SQL Server 2012 Express como banco de dados, a linguagem Microsoft® Visual Basic.NET, Java Script e o Microsoft® Report Viewer para criação e diagramação dos relatórios.

Palavras-chave: Linguagem Visual Basic. Modelagem com UML. Processo unificado.

ABSTRACT

FONSECA, Cicero Raldi, ROSSI, Fausto Daniel. System for managing companies that promote social events. 2013. 64f. Monografia de Trabalho de Conclusão de Curso - Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, Universidade Tecnológica Federal do Paraná. Pato Branco, 2013.

Organizing an event takes a lot of information. To manage a social agreements are needed agreements between client and organizer, invoices, protocol. The company that produce events is often responsible for managing the event altogether, thus having to store various information about the event, customers and other aggregated information. Based on the large amount of information to be stored, there is a need to produce management software of these companies. This management refers to the possibility of these companies store all information relating to the event in a centralized database, with security and flexibility to search for the desired information. This work concerns the development of a web based system for managing companies that produce events. The implementation was performed using the Microsoft® Visual Studio 2012 Express, Microsoft® SQL Server 2012 Express as database, Microsoft® Visual Basic.NET, JavaScript, and Microsoft® Report Viewer to create reports.

Palavras-chave: Linguagem Visual Basic. Modelagem com UML. Processo unificado.

LISTA DE FIGURAS E QUADROS

Figura 1 - Arquitetura típica Rias	18
Figura 2 - Exemplo de interface do Microsoft Report Viewer	23
Figura 3 - Exemplo de interface do Microsoft SQL Server Management Studio	24
Figura 4 - Exemplo de interface do Visual Paradigm for UML.....	25
Figura 5 - Exemplo de Interface do Case Studio 2.....	26
Figura 6 - Diagrama de casos de uso	31
Figura 7 - Representação de parte do diagrama de classes.....	32
Figura 8 - Representação de parte do modelo entidade-relacionamento.....	33
Figura 9 - Tela de inicialização e login	34
Figura 10 - <i>Master page</i>	35
Figura 11 - Tela de cadastro da empresa	36
Figura 12 - Tela exibindo o <i>grid</i> de cadastro de clientes	37
Figura 13 - Tela exibindo os botões incluir e excluir.....	37
Figura 14 - Tela exibindo os botões incluir e excluir.....	37
Figura 15 - Tela de cadastro de clientes, aba principal	38
Figura 16 - Tela de cadastro de clientes, aba Endereço/Obs.....	39
Figura 17 - Tela exibindo o <i>grid</i> de eventos	40
Figura 18 - Tela de cadastro de eventos, aba principal.....	41
Figura 19 - Tela de cadastro de eventos, aba Evento.....	42
Figura 20 - Tela de cadastro de eventos, aba Evento demonstrando os campos alterados ao selecionar o tipo de evento Formatura	43
Figura 21 - Tela de cadastro de contratos.....	44
Figura 22 - Tela de cadastro de eventos, aba Formandos demonstrando a inclusão dos formandos nos eventos do tipo Formatura	45
Figura 23 - Tela de cadastro de contratos.....	46
Figura 24 - Tela de cadastro de contratos demonstrando a aba Eventos	47
Figura 25 - Tela de filtros do relatório de cadastros de eventos.....	48
Figura 26 - Relatório de cadastro de eventos.....	49
Figura 27 - Tabelas para geração do relatório de cadastro de eventos	60
Quadro 1 - Comparação entre aplicações desktop, web e RIA.....	15

LISTAGENS DE CÓDIGO

Listagem 1 - <i>String</i> de conexão.....	50
Listagem 2 - Classe responsável pela conexão com o banco de dados	50
Listagem 3 - Código que carrega os dados no <i>AspxGridView</i>	51
Listagem 4 - Código responsável por chamar a tela de cadastro de eventos	51
Listagem 5 - Código para abertura da tela de cadastros com efeito modal	52
Listagem 6 - Método de inclusão/alteração dos cadastros de eventos	53
Listagem 7 - Método responsável pela paginação do <i>AspxGridView</i>	53
Listagem 8 - Método de carregamento do evento no formulário de cadastro	54
Listagem 9 - Código JavaScript utilizando a biblioteca jQuery para os <i>ToolTips</i>	55
Listagem 10 - Código JavaScript para formatação de máscaras de CEP e CNPJ....	55
Listagem 11 - Código JavaScript responsável pela verificação e atribuição de classe CSS ao registro do <i>AspxGridView</i>	56
Listagem 12 - Código CSS para estilização do <i>AspxGridView</i>	57
Listagem 13 - Código responsável pela abertura da tela de adição de contratos	58
Listagem 14 - Código do componente <i>AspxGridView</i>	59
Listagem 15 - Código responsável pelo carregamento das informações para a impressão do relatório de cadastro de eventos.....	61

LISTA DE SIGLAS

CEP	Código de Endereçamento Postal
CLR	<i>Common Language Runtime</i>
CNPJ	Cadastro Nacional de Pessoas Jurídicas
CSS	<i>Cascading Style Sheets</i>
CSS	<i>Cascading Style Sheets</i>
DER	Diagrama de Entidades e Relacionamentos
EPE	Empresas Produtores de Eventos
HTML	<i>Hypertext Markup Language</i>
MP	<i>Master Page</i>
NET	<i>Microsoft Visual Basic.NET</i>
ORM	<i>Object/Relational Mapping</i>
RIA	<i>Rich Internet Application</i>
SQL	<i>Struntured Query Language</i>
UML	<i>Unified Modeling Language</i>

SUMÁRIO

1. INTRODUÇÃO	11
1.1 Considerações Iniciais	11
1.2 Objetivos	12
1.2.1 Objetivo Geral.....	12
1.2.2 Objetivos Específicos	12
1.3 Justificativa	13
1.4 Estrutura do Trabalho	13
2. GERENCIAMENTO DE EVENTOS E <i>RICH INTERNET APPLICATIONS</i>	14
2.1 Eventos	14
2.2 Gerenciamento de Eventos.....	15
2.3 Aplicações para Internet com Interface Rica.....	15
2.3.1 Histórico de Aplicações para Internet com Interface Rica	16
2.3.2 Conceito	16
2.3.3 Arquitetura RIA.....	18
2.3.4 Vantagens da utilização de RIA.....	19
3. MATERIAIS E MÉTODO	20
3.1 Materiais	20
3.1.1 Microsoft® Visual Basic.NET.....	20
3.1.2 .NET Framework	20
3.1.3 JavaScript.....	21
3.1.4 Ajax Control Tool Kit.....	21
3.1.5 JQuery	21
3.1.6 HTML5.....	21
3.1.7 CSS3.....	22
3.1.8 Noty.....	22

3.1.9	Entity Framework.....	22
3.1.10	Microsoft® Report Viewer.....	23
3.1.11	Microsoft® SQL Server.....	24
3.1.12	Visual Paradigm for UML.....	25
3.1.13	Case Studio 2.....	26
3.2	Método	27
3.2.1	Complementação dos Requisitos	27
3.2.2	Complementação da Análise e Projeto	27
3.2.3	Implementação do Sistema	27
3.2.4	Testes.....	28
4.	RESULTADOS	29
4.1	Apresentação do Sistema	29
4.2	Modelagem do Sistema	29
4.3	Descrição do Sistema	34
4.4	Implementação do Sistema.....	49
5.	CONCLUSÃO	62
	REFERÊNCIAS	63

1. INTRODUÇÃO

Neste capítulo serão apresentadas as considerações iniciais sobre o presente trabalho, bem como seus objetivos, justificativa e a estrutura do texto.

1.1 Considerações Iniciais

A Tecnologia da Informação é um conjunto de todas as atividades e soluções providas por recursos de computação. A informatização aponta várias vantagens na realização das atividades pelas pessoas, facilitando, assim, o cotidiano de seus usuários.

Uma Empresa Produtora de Eventos (EPE) tem como principal objetivo a prestação de serviços e produtos relacionados a eventos como formaturas, casamentos, aniversários, confraternizações e encontros empresariais e afins. Para esses eventos geralmente são prestados serviços como fotografias, filmagens, iluminação, decoração, sonorização, *buffet* e estrutura. Juntamente com a prestação de serviços são, também, locados ou vendidos os produtos para complementar os eventos.

Os autores deste trabalho, após alguns anos atuando no mercado de eventos em Pato Branco e região, foi constatado que a crescente demanda por empresas especializadas na produção, organização e execução de eventos não está sendo acompanhada pelos softwares de gerenciamento para esta área. Na organização e realização de eventos há somas consideráveis de recursos financeiros envolvidos e um enorme volume de informações. Em muitos casos os recursos e informações têm sido gerenciados de forma manual ou com sistemas que pouco atende as necessidades específicas desse tipo de negócio. Percebeu-se, então, que o gerenciamento de eventos poderia ser otimizado por um software especializado.

De acordo com as necessidades encontradas no controle e gerenciamento de eventos em empresas da região, concluiu-se que um software especializado é uma alternativa para empresas do ramo. O software poderá tornar mais ágil e eficaz o levantamento e a conclusão de tarefas relacionadas aos eventos, facilitando também a busca de informações. Portanto, verificou-se a oportunidade de realizar um sistema para gerenciamento de empresas de eventos. A modelagem desse sistema

foi realizada como trabalho de estágio e a sua implementação foi realizada como trabalho de conclusão de curso.

1.2 Objetivos

O objetivo principal está relacionado ao desenvolvimento de um aplicativo computacional para o gerenciamento de EPE. Os objetivos específicos complementam o objetivo geral no sentido das finalidades de uso desse objeto desenvolvido.

1.2.1 Objetivo Geral

Implementar um sistema *web* para gerenciamento de empresas produtoras de eventos.

1.2.2 Objetivos Específicos

- Prover uma maneira de auxiliar EPEs no gerenciamento do processo de controle, organização e execução de eventos;
- Auxiliar no controle de estoque e dos produtos disponibilizados na realização dos eventos;
- Agilizar o controle de orçamentos e contratos;
- Facilitar a busca por informações sobre eventos, contratos, orçamentos, financeiro, produtos de estoque e cadastros gerais;
- Auxiliar no controle de agendamento de eventos;
- Disponibilizar cadastros dos clientes, fornecedores, produtos e serviços da EPE;
- Disponibilizar uma forma de controle de estoque e financeiro para EPEs;
- Possibilitar o cadastro dos eventos do tipo formatura e realizar o controle de contratos dos clientes referentes ao evento.

1.3 Justificativa

Notou-se deficiência quanto à automatização desse nicho de mercado que é o de empresas produtoras de eventos. Atualmente, o controle e o gerenciamento de contratos, arquivamento de dados referente aos eventos e controle financeiro da empresa são feitos em sua maioria de forma manual ou utilizando softwares não especializados, sendo, assim, suscetíveis à falhas e/ou extravio de arquivos e documentos.

Existem vários processos em EPE que podem ser automatizados como o processo de controle de contratos, produtos e serviços envolvidos com os eventos, controle de contas a pagar e receber, lançamentos em caixa, cadastro de clientes, fornecedores e funcionários, *follow ups* e orçamentos. Esses processos, se integrados a um sistema que os gerencie, podem trazer muitos benefícios ao gerente de uma EPE. Isso porque o desvincularia de atividades rotineiras e ele poderia dedicar o tempo que utilizaria para as tarefas que podem ser automatizadas no melhoramento dos processos, no atendimento a clientes, na busca de novos clientes e demais atividades gerenciais e estratégicas.

1.4 Estrutura do Trabalho

Este texto está organizado em capítulos. Este é o primeiro e apresenta a introdução com a visão geral do que trata o trabalho, incluindo os objetivos e a justificativa do mesmo. No Capítulo 2 está o referencial teórico, centrado em gerenciamento de eventos e de aplicações *web* caracterizadas como de interface rica. O Capítulo 3 apresenta os materiais e o método utilizados no desenvolvimento do sistema. No Capítulo 4 é apresentado o resultado obtido com a realização deste trabalho que é o sistema desenvolvido. Os resultados incluem a modelagem do sistema, a apresentação das suas funcionalidades por meio das telas (interface) e exemplos da implementação por meio de listagens de código. No Capítulo 5 estão as considerações finais e as perspectivas futuras para continuidade deste trabalho.

2. GERENCIAMENTO DE EVENTOS E *RICH INTERNET APPLICATIONS*

Este capítulo apresenta o referencial teórico utilizado para fundamentar o sistema proposto. O capítulo inicia com uma explanação sobre eventos e seu gerenciamento, em seguida apresenta conceitos relacionados a aplicações ricas para ambiente Internet.

2.1 Eventos

Evento é sinônimo de acontecimento ou de ação, como o deslocamento de pessoas a determinada localidade para assistir, participar ou palestrar determinado acontecimento.

"A reunião caracteriza-se como o embrião de todos os tipos de eventos. Trata-se do encontro de duas ou mais pessoas, a fim de discutir, debater e solucionar questões sobre determinado tema relacionado com suas áreas de atividade" (MEIRELLES, 1999, p. 89).

Um evento é um acontecimento previamente planejado, com objetivos claramente definidos. Tem um perfil marcante: esportivo, social, comercial, cultural, filantrópico, religioso, entre outros. Sua realização obedece a um cronograma e uma de suas metas é a interação entre seus participantes, público, personalidades e entidades. Giácomo (1993, p. 33) analisa o evento como "componente do 'mix' da comunicação, que tem por objetivo minimizar esforços, fazendo uso da capacidade sinérgica da qual dispõe o poder expressivo, no intuito de engajar pessoas numa ideia ou ação".

Portanto pode-se afirmar que um evento tem como característica principal propiciar uma ocasião extraordinária ao encontro de pessoas, com finalidade específica, a qual constitui o "tema" principal do evento e justifica a sua realização.

2.2 Gerenciamento de Eventos

Na cidade de Pato Branco-PR e região, a realização de eventos sociais, como formaturas, casamentos, aniversários e outros, está cada vez mais frequente e em maior quantidade. Para atender toda essa demanda, empresas especializadas em produção, organização e gerenciamento de eventos são necessárias.

Essas empresas produtoras de eventos têm como principal objetivo a prestação de serviço, organizando e formatando o evento, para que o mesmo seja executado de acordo com o planejamento e a contento dos organizadores, promotores e participantes. E, ainda, que o mesmo fique registrado por meio de filmagens e fotografias. Além do registro, os serviços mais comuns prestados por esse tipo de empresa são: sonorização, *buffet*, recepção, segurança, decoração, iluminação entre outros.

Para que os eventos ocorram conforme o planejado e atenda às expectativas dos participantes e dos clientes, as empresas especializadas precisam ter um controle adequado, comunicação e boa organização, evitando que nenhum detalhe seja esquecido ou deixe a desejar. No entanto, para que esse controle seja bem sucedido são criados contratos de prestação de serviços e a empresa precisa trabalhar com um controle de estoque que seja composto pelos produtos que fazem parte dos pacotes de serviços prestados pela empresa.

Um sistema especializado é a melhor alternativa para fazer esse controle, obtendo informações rapidamente, com o máximo de detalhes possíveis do evento em questão, centralizando as informações gerenciais, de estoque e financeiro da empresa em um só lugar, agilizando as consultas, cadastros, impressões de contratos e emissão de relatórios.

2.3 Aplicações para Internet com Interface Rica

Aplicações Internet com interface que se assemelha aos recursos de interação proporcionados pelas aplicações *desktop* são denominadas de interface rica. Isso porque essas aplicações apresentam recursos muito diferenciados e que proporcionam maior interatividade se comparadas às aplicações *web* baseadas em hipertexto. Os recursos de interação como botões e menus diferenciados e efeitos

de arrastar-e-soltar, por exemplo, que elas apresentam é que fornecem a riqueza de interação, caracterizando-as.

2.3.1 Histórico de Aplicações para Internet com Interface Rica

Nos anos 90 um novo modelo de aplicações foi criado: aplicações para Internet com foco em um desenvolvimento de baixo custo e que fosse mais rápida. O modelo é baseado em *Hypertext Markup Language* (HTML) e nos navegadores, servidores de aplicação cuja função é criar e enviar as páginas aos navegadores de internet. A navegação é fornecida ao cliente de forma estática. Nesse modelo a interação do cliente com a página é enviada ao servidor para processar a requisição, validar, consultar e devolver a página ao cliente com todas as informações devidamente atualizadas (ALLAIRE, 2002).

Em 2002 a Macromedia introduziu o termo RIA – Aplicações Ricas para Internet (*Rich Internet Application*). Este conceito baseia-se no modelo de Aplicações para Internet, buscando as melhores funcionalidades de uma aplicação *desktop* e o baixo custo de desenvolvimento de aplicações web, juntamente com interatividade e tecnologias de comunicação (ALLAIRE, 2002).

O conceito tradicional de internet impede o tipo de navegação interativo e animado, que permite redimensionar, arrastar-e-soltar e muito mais, pois, normalmente é necessário clicar em um *link* para visualizar algo e então clicar em outro, em outro, e assim por diante (DUHL, 2003).

2.3.2 Conceito

Para Guanais (2010), Aplicações de Internet Rica (RIA) são um novo tipo de aplicação *web* com o objetivo de incrementar e melhorar as opções e capacidades das aplicações *web* tradicionais. Segundo López (2005) as atuais limitações da camada de apresentação dos navegadores e da linguagem HTML é o que impulsiona os desenvolvedores a utilizar este novo tipo de aplicações, que permitem entre outras, melhorar a experiência de uso de aplicação pelo usuário, execução de

conteúdo multimídia e a recuperação de dados de aplicações tanto *online*, quanto *off-line*, dependendo da tecnologia RIA que for utilizada.

As RIAs possuem uma série de características gerais, que para López (2005) são as seguintes:

- A aplicação é iniciada a partir de uma página *web*;
- A reação de interação do usuário com a aplicação é imediata, já que nas RIAs não é necessário renderizar uma página *web* inteira, quando novos dados são solicitados, evitando assim que a janela do navegador fique branca enquanto recebe os dados do servidor para renderizar novamente a página *web*;
- A complexidade do desenvolvimento das aplicações RIA não difere muito das aplicações *web* tradicionais;
- A aplicação permite realizar operações comuns de aplicações desktop como redimensionamento, uso do arrastar e soltar e animação de objetos;
- A aplicação utiliza controles de interface de usuário mais modernos como menus e painéis de aba entre outros elementos gráficos de interface que não são utilizados em aplicações *web* tradicionais, aplicações não ricas;
- A aplicação não demanda processamento excessivo ou suporte de alguma plataforma específica e possui compatibilidade com todos os navegadores *web*.

Uma comparação entre uma aplicação *desktop* (cliente/servidor), uma aplicação *web* convencional (baseada em HTML) e uma RIA é apresentada no Quadro 1. O potencial da RIA baseia-se no modelo de distribuição da *web*, interfaces melhoradas e redução da sobrecarga de comunicação (BOZZON, COMAI 2006). Esses benefícios são relevantes para tarefas, como por exemplo, ordenação e filtragem de dados.

Característica	Desktop	Web tradicional	RIA
<i>Cliente universal (browser)</i>	Não possui	Possui	Possui
<i>Instalação do cliente</i>	Complexa	Simple	Simple
<i>Capacidade de interação</i>	Rica	Limitada	Rica

<i>Lógica de negócio do lado do servidor</i>	Sim	Sim	Sim
<i>Lógica de negócio do lado do cliente</i>	Sim	Limitada	Sim
<i>Requer recarregamento da página completa</i>	Não	Sim	Não
<i>Requisição e resposta frequente do servidor</i>	Não	Sim	Não
<i>Comunicação do servidor para o cliente</i>	Sim	Não	Sim
<i>Funcionamento desconectado</i>	Sim	Não	Sim

Quadro 1. Comparação entre aplicações desktop, web e RIAs

Fonte: Bozzon e Comai (2006).

2.3.3 Arquitetura RIA

RIA utiliza o mesmo conceito arquitetônico de aplicações de internet e permite uma experiência de navegação semelhante ao ambiente *desktop* (DUHL, 2003).

De acordo com Guanais (2010) o cliente deve possuir um navegador que é necessário para interpretar os dados que chegam da aplicação que está hospedada no servidor. O armazenamento da interface gráfica de uma RIA se diferencia das aplicações *web* tradicionais, evitando-se as sucessivas renderizações a cada comunicação com o servidor para a exibição de dados. O servidor compila a aplicação *web*, se necessário, e realiza a intermediação da comunicação das aplicações executando no navegador *web* do cliente com outras aplicações ou *servlets*. A arquitetura de uma RIA pode também incluir um gerenciador de dados, que habitualmente é uma base de dados com os dados relacionados ao serviço oferecido pela aplicação. O gerenciador de dados nem sempre é necessário. Há casos de aplicações que não utilizam uma base de dados própria, mas utilizam serviços disponíveis na Internet. A Figura 1 exemplifica a forma de interação entre o navegador, o servidor e o banco de dados.

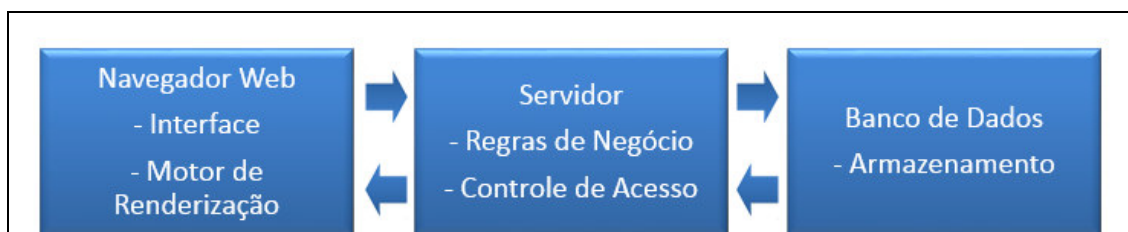


Figura 1 - Arquitetura típica RIAs

Fonte: Adaptado de Oracle (2007)

De acordo com a Figura 1, o cliente deve possuir um navegador (*browser web*) que permita a exibição dos recursos de uma RIA. Isso pode ser realizado por meio de *plugins* que permitem interpretar os dados que chegam da aplicação hospedada no servidor. Esses dados podem ser arquivos de vídeo, textos, áudio, animações, entre outros. A arquitetura de uma RIA se encarrega também de obter os dados necessários para a aplicação que podem estar em uma base de dados qualquer ou em arquivos.

2.3.4 Vantagens da utilização de RIA

A riqueza da interface oferecida ao usuário destaca-se entre as vantagens da utilização de RIA, uma interatividade próxima ao ambiente desktop e o equilíbrio entre cliente/servidor, ou seja, o processamento é realizado no cliente e posteriormente enviado somente o necessário ao servidor. Possibilitando, assim, o atendimento de requisições de outros clientes ao mesmo tempo, tornando a navegação do usuário mais rápida (DUHL, 2003).

A comunicação assíncrona entre o cliente e servidor também é outra vantagem. Com isso as ações executadas pelo usuário na interface, como o clicar em um botão, não precisam esperar um longo tempo pela resposta do servidor, pois a partir da solicitação são carregados antecipadamente os dados para uma resposta posterior acelerada. Isso torna a navegação mais rápida e reduz o tráfego na rede (DUHL, 2003).

3. MATERIAIS E MÉTODO

Este capítulo apresenta informações sobre as ferramentas utilizadas na modelagem e no desenvolvimento do software proposto.

3.1 Materiais

Para o desenvolvimento da modelagem do software utilizou-se a ferramenta Visual Paradigm, para a definição das tabelas do banco de dados o Case Studio, para criação do banco de dado o Microsoft SQL Server e para o desenvolvimento do software o Microsoft Visual Basic.NET.

3.1.1 Microsoft® Visual Basic.NET

Visual Basic.NET (.NET) é uma linguagem de programação orientada a objetos e com suporte a Unified Modeling Language (UML) criada pela Microsoft®. O Microsoft® Visual Basic.NET é um produto diferente do antigo Visual Basic 6.0. Não apenas a maneira de programar foi alterada, mas todo o conceito de orientação a objetos que trouxe agilidade para a linguagem. Isso é decorrente do ambiente de desenvolvimento Visual Studio incorporar recursos com funcionalidades bastante importantes e úteis: Intellisense, snippets. Para o desenvolvimento foi utilizada a versão 2012.

3.1.2 .NET Framework

O .NET Framework é uma plataforma de desenvolvimento para a criação de aplicativos para Windows, Windows Phone, Windows Store, Windows Server e Windows Azure. A plataforma .NET Framework inclui as linguagens de programação C# e Visual Basic, o *Common Language Runtime* (CLR) e uma ampla biblioteca de classes, disponível em <http://msdn.microsoft.com/pt-br/vstudio/aa496123>.

3.1.3 JavaScript

JavaScript é uma linguagem de programação interpretada e com recursos de orientação a objetos e permite que o conteúdo executável seja incluído em páginas *web*. Assim, uma página *web* não precisa mais de HTML estático, mas pode incluir programas que interagem com o usuário, controlam o navegador e criam conteúdo HTML dinamicamente (FLANAGAN, 2002). Seu objetivo é atender as limitações apresentadas pelo HTML possibilitando validações de formulários, no caso do Sistema para Gerenciamento de EPE.

3.1.4 Ajax Control Tool Kit

O Ajax Control Tool Kit é um projeto de código aberto construído com base na estrutura ASP.NET Ajax. É um esforço conjunto entre a Microsoft® e a comunidade Ajax que fornece uma infraestrutura para desenvolvimento de códigos personalizados e reutilizáveis. O Ajax Control Toolkit contém mais de 30 controles que permite a criação rica e simples de página *web* interativas. Para o desenvolvimento do trabalho foi utilizada a versão AjaxControlToolkit 7.0725.

3.1.5 JQuery

JQuery é uma biblioteca JavaScript desenvolvida para simplificar os *scripts* que interagem com o HTML. Sua sintaxe foi desenvolvida para tornar mais simples a navegação do documento HTML e criar animações. A biblioteca oferece a possibilidade de criação de *plugins* sobre ela. Com uma combinação de versatilidade e capacidade de expansão, JQuery mudou a maneira de escrever JavaScript.

3.1.6 HTML5

HTML é uma linguagem de marcação, utilizada para formatação de páginas *web* com texto e informações separadamente. Utilizando qualquer tipo de *browser* como: Internet Explorer, Firefox, Chrome, Opera ou Safari. Esses *browsers* têm a

função de interpretar, formatar e exibir páginas Web. Um documento HTML contém um texto que precisa ser formatado e estruturado.

O projeto HTML5 foi iniciado em 2004 e no início de novembro de 2011 a Adobe anunciou que interromperia o desenvolvimento de Flash para dispositivos móveis e redirecionar seus esforços para o desenvolvimento de ferramentas utilizando HTML5.

3.1.7 CSS3

O *Cascading Style Sheets* (CSS) é responsável pelo aspecto da página, pelo layout, garante acessibilidade, tanto para máquinas quanto para as pessoas. Agiliza a leitura das páginas web pelos navegadores. Não é uma linguagem de programação nem de marcação. É uma sequência de declarações de propriedades e seus valores para a manipulação da forma como o conteúdo de uma página web será exibido (GOMES, 2010).

A grande vantagem do uso de CSS é a de separar a marcação HTML, da apresentação do site. HTML marca e estrutura de textos, cabeçalhos, parágrafos, *links*, botões, formulários, imagens e demais elementos da página e CSS define cores, posicionamento na tela, estilos de linhas, bordas e tudo o mais relacionado à apresentação (SILVA, 2012).

3.1.8 Noty

Noty é um *plugin* JQuery que permite a criação de mensagens de alerta, sucesso, erro, advertência e confirmação como alternativa além das mensagens padrão Java Script. As mensagens podem ser personalizadas em texto, animação, velocidade, inclusão de botões, dentre outros.

3.1.9 Entity Framework

O Microsoft® Entity Framework é um *framework* do tipo *Object/Relational Mapping* (ORM) que permite aos desenvolvedores trabalhar com dados relacionais como objetos de domínio específico, eliminando a necessidade de maior parte dos códigos de acesso de dados que os desenvolvedores geralmente precisam

escrever. O modelo ORM do Entity Framework fornece serviços como rastreamento de alterações, resolução de identidades, *lazy loading* (carga preguiçosa ou tardia) e tradução de consultas para que os desenvolvedores possam se concentrar na lógica de negócios de seus aplicativos em vez dos princípios básicos de acesso a dados.

O Entity Framework é baseado no modelo de provedor existente do ADO.NET, com provedores existentes sendo atualizados para suportar a nova funcionalidade Entity Framework. Por essa razão, as aplicações existentes baseadas no ADO.NET podem ser usadas facilmente no Entity Framework com um modelo de programação familiar aos desenvolvedores ADO.NET.

3.1.10 Microsoft® Report Viewer

O Microsoft® Report Viewer permite que os aplicativos executados no .NET Framework exibam relatórios criados usando a tecnologia de relatórios da Microsoft®. A Figura 2 mostra a interface do Microsoft® Report Viewer, ferramenta utilizada para a criação dos relatórios do Sistema para Gerenciamento de EPE.

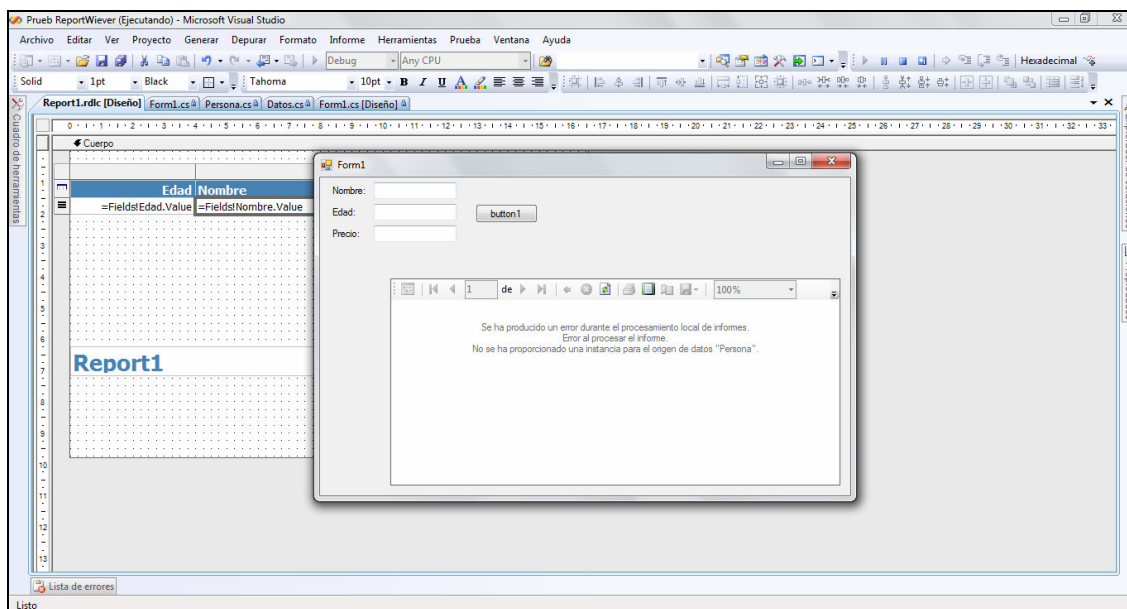


Figura 2 - Exemplo de interface do Microsoft Report Viewer
Fonte: Autoria própria.

3.1.11 Microsoft® SQL Server

O Microsoft® SQL Server é um sistema de gerenciamento e análise de banco de dados para soluções de comércio eletrônico, linha de negócios e *data warehouse*.

O Microsoft® SQL Server 2012, a versão mais recente, apresenta novas soluções de alta disponibilidade e recuperação de desastres, armazenamento em memória focado em desempenho extremamente rápido de consultas, exploração rápida de dados e *business intelligence* escalonável.

A Figura 3 mostra a interface do Microsoft® SQL Server Management Studio, ferramenta utilizada para manipulação de dados do Microsoft® SQL Server 2012. Nela é possível acessar os registros das bases de dados, criar e editar *scripts* em SQL.

The screenshot displays the Microsoft SQL Server Management Studio interface. The main window shows a SQL query in the 'SQLQuery1.sql' file. The query is a complex SELECT statement with multiple joins and conditional logic. Below the query, the 'Resultados' pane shows a table with 17 rows of data. The columns are: ClienteID, Nome, RazaoSocial, Situacao, Sexo, Nascimento, Tipo, Rg, and CPF. The status bar at the bottom indicates 'Consulta executada com êxito.' and '17 linhas'.

ClienteID	Nome	RazaoSocial	Situacao	Sexo	Nascimento	Tipo	Rg	CPF
1	Adriana Antati		Interessado	F	14/11/2012	Novo	34.233.677-7	665.676.877-87
2	Alex Bergamo		Interessado	M	26/09/2012	Novo		
3	Anatel	Agência Nacional de Telefonia	Interessado		26/09/2012	Empresa		
4	Cicero Raldi De Fonseca		Ativo	M	14/11/2012	Novo	0.532.567-0	056.580.212-57
5	Douglas Bandeira		Inativo	M	26/09/2012	Novo		
6	Evelise Fonseca		Ativo	F	14/11/2012	Familiar	32.5652.676-7	345.656.864-35
7	Fausto Daniel Rossi		Ativo	M	14/11/2012	Amigo	9.687.413-0	061.654.139-52
8	Geri Natalino Dutra		Ativo	M	14/11/2012	Amigo	3.125.342-6	455.467.688-98
9	Juca Bala		Ativo	M	26/09/2012	Novo		
10	Lucas Andre Rossi		Ativo	M	26/09/2012	Familiar		
11	Marivete Pluinschi		Ativo	F	14/11/2012	Familiar	45.23556.23.23	456.768.790-96
12	Prefeitura Pato Branco	Prefeitura Pato Branco	Ativo		14/11/2012	Empresa		54.567.899/7654-43
13	R.U.	Restaurante Universitário UTFPR	Ativo		14/11/2012	Empresa		23.231.254/5676-98
14	Sporte Informatica	RC Serviços LTDA.	Ativo		14/11/2012	Empresa		67.498.633/0000-02
15	Tulo Stroni		Inativo	M	26/09/2012	Amigo		
16	Ufpr	Universidade Tecnológica Fed...	Interessado		26/09/2012	Empresa		
17	Viasoft Softwares Empres...	Viasoft Informática LTDA.	Ativo		26/09/2012	Empresa		

Figura 3 - Exemplo de interface do Microsoft SQL Server Management Studio

Fonte: Autoria própria.

3.1.12 Visual Paradigm for UML

Visual Paradigm for UML é uma ferramenta CASE com várias opções de modelagem. A ferramenta possui um ambiente de trabalho que facilita a visualização e a manipulação do projeto de modelagem.

Na Figura 4 é mostrada a interface do software Visual Paradigm for UML, sistema esse utilizado para criação de diagramas do tipo: casos de uso, diagramas de classe, diagramas de sequência, diagramas de atividade e outros.

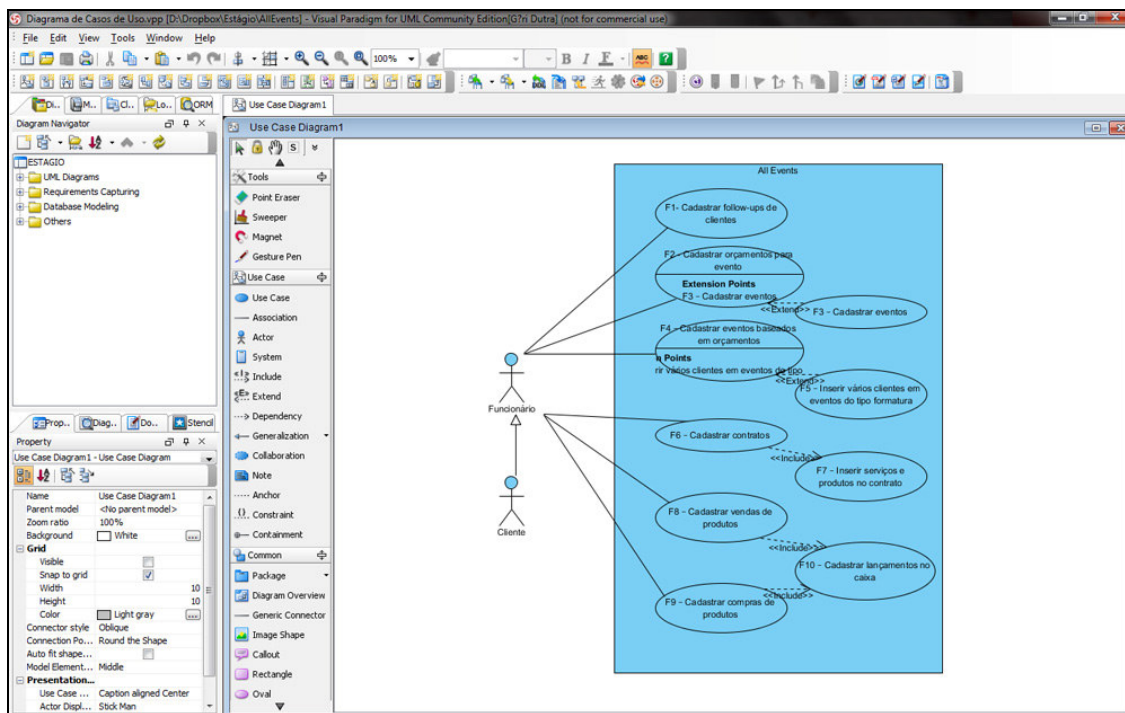


Figura 4 - Exemplo de interface do Visual Paradigm for UML

Fonte: Autoria própria.

A ferramenta Visual Paradigm, além de modelar os diversos tipos de diagrama ainda gera automaticamente outros diagramas através da análise dos casos, como por exemplo: um Diagrama de Entidade Relacionamento (DER) a partir de um diagrama de classe e depois gera o código SQL a partir do DER, ainda é capaz de gerar código PHP, Java, C, C++, C# entre outros (VISUAL PARADIGM, 2012).

O Visual Paradigm for UML foi utilizado nesse projeto para criação dos diagramas de classes e diagrama de casos de uso.

3.1.13 Case Studio 2

CASE Studio 2 é uma ferramenta de projeto de banco de dados profissional. Ele foi projetado para criar DER para vários tipos de bancos de dados. Essa ferramenta permite gerar *scripts* SQL e relatórios detalhados, dentre outros recursos.

O DER foi utilizado no projeto para definir os relacionamentos entre as tabelas da base de dados. Esse modelo baseia-se na percepção de mundo real, que consiste em uma coleção de objetos básicos, chamados entidades, e em relacionamentos entre objetos, como mostra a Figura 5.

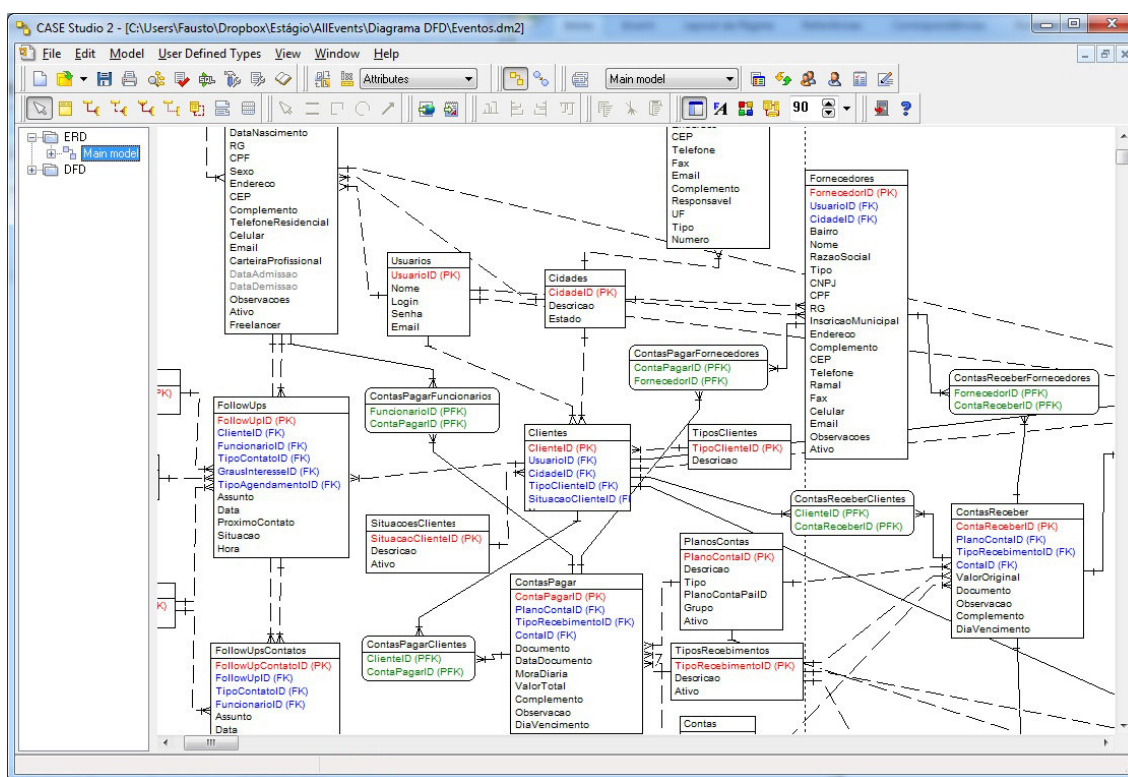


Figura 5 - Exemplo de Interface do Case Studio 2

Fonte: Autoria própria

O DER representa os dados em tabelas e como se relacionam, seguindo algumas regras especificadas *a priori*. O Modelo Relacional mostra explicitamente o mapeamento entre as diversas tabelas são ditas "relações" ou "instâncias" do banco de dados, uma vez que pode existir física ou apenas logicamente (LEITE, 2008).

3.2 Método

Para o desenvolvimento deste trabalho foram realizadas as etapas a seguir que estão descritas juntamente com suas atividades principais.

O método de pesquisa utilizado para desenvolvimento deste trabalho classifica-se como bibliográfica e descritiva que para Thomas, Nelson e Silverman (2007, p.54) “tem por premissa buscar a resolução de problemas melhorando as práticas por meio da observação, análise e descrições objetivas, através de entrevistas com peritos para a padronização de técnicas e validação de conteúdo”.

3.2.1 Complementação dos Requisitos

O levantamento de definição dos requisitos foi realizado como trabalho de estágio e teve como base EPEs que podem tornar-se usuários do sistema. No trabalho de conclusão em questão, os requisitos foram novamente verificados e complementados. Uma EPE foi utilizada como parâmetro para um melhor detalhamento dos requisitos o que auxiliou e facilitou a definir as funcionalidades que deveriam ser efetivamente implementadas.

3.2.2 Complementação da Análise e Projeto

A documentação da modelagem do sistema foi realizada como trabalho de estágio e nessa fase todos os documentos foram analisados, complementados e validados. Um diagrama de classes e diagrama de casos de uso completo foi gerado, bem como o modelo entidade relacionamento visando definir quais as tabelas e de que forma as mesmas se relacionam no banco de dados.

3.2.3 Implementação do Sistema

Alguns cadastros simples foram implementados no trabalho de estágio para que os autores pudessem estudar as tecnologias utilizadas. Como trabalho de conclusão de curso os demais requisitos e funcionalidades definidas durante o levantamento de requisitos foram implementadas.

3.2.4 Testes

Os testes foram realizados pelos próprios autores deste trabalho. Esses testes tiveram como objetivo encontrar erros de código e verificar a facilidade que o software para gerenciamento de EPE possui de ser claramente compreendido e utilizado pelos usuários.

4. RESULTADOS

Neste capítulo é apresentado o sistema desenvolvido como resultado deste trabalho. Inicialmente é apresentada a descrição do mesmo, seguida por sua modelagem e exemplos dos códigos gerados durante sua fase de implementação.

4.1 Apresentação do Sistema

O sistema para gerenciamento de EPE desenvolvido como resultado da realização deste trabalho de conclusão de curso visa oferecer uma maneira de auxiliar EPEs no gerenciamento do processo de controle, organização e execução de eventos, auxiliar no controle de estoque, controle de orçamentos, contratos, de forma que agilize a busca de informações destes.

4.2 Modelagem do Sistema

Na primeira etapa de definição do sistema foi realizado o levantamento dos requisitos funcionais, responsáveis por delimitar quais seriam as funcionalidades que o sistema deveria atender, ou seja, seu escopo de atuação. Nos vários encontros realizados com a gerência de uma EPE, ficaram definidos como necessidades da mesma os seguintes requisitos:

- Cadastrar dos dados da EPE;
- Cadastrar usuários do sistema;
- Cadastrar cargos de funcionários;
- Cadastrar funcionários;
- Cadastrar cidades;
- Cadastrar tipo de cliente;
- Cadastrar clientes;
- Cadastrar tipo de contato de *follow up*;
- Cadastrar *follow up* de clientes;
- Cadastrar fabricantes;
- Cadastrar fornecedores;

- Cadastrar tipo de produtos;
- Cadastrar produtos do estoque;
- Cadastrar tipo de serviços;
- Cadastrar serviços que podem fazer parte de contratos;
- Cadastrar plano de contas;
- Cadastrar orçamentos para evento;
- Emitir relatório de orçamento de determinado evento;
- Cadastrar tipos de eventos;
- Cadastrar eventos (casamentos, formaturas, aniversários, etc.);
- Cadastrar eventos baseados em orçamentos existentes;
- Inserir vários clientes em eventos do tipo formatura;
- Emitir relatório de determinado evento;
- Cadastrar contratos (baseados em eventos existentes);
- Inserir serviços e produtos no contrato do evento;
- Cadastrar valores devidos pelo cliente do contrato de determinado evento;
- Registrar cancelamento do contrato de determinado cliente do evento;
- Emitir contrato do evento dos clientes;
- Emitir boletos dos contratos;
- Cadastrar vendas de produtos;
- Cadastrar contas a receber baseadas em vendas;
- Cadastrar contas a receber;
- Registrar recebimento de contas;
- Cadastrar compras de produtos;
- Cadastrar contas a pagar baseadas em compras;
- Cadastrar contas a pagar;
- Registrar pagamento de contas;
- Cadastrar lançamentos no caixa.

Os eventos sociais prestados por esse tipo de empresa são divididos da seguinte forma:

- Casamentos;
- Formaturas;
- Aniversários/bodas;

- Outros (eventos genéricos, que não ocorrem com tanta frequência).

Os valores cobrados para cada evento variam da estrutura, produção, produtos e serviços envolvidos com o mesmo. Os valores são definidos com base nos materiais usados, quantidades dos mesmos, tamanho de fotos, número de câmeras utilizadas na gravação do evento, serviços relacionados, como por exemplo: telão, *buffet*, recepcionistas e seguranças.

O sistema ainda permitirá a inclusão, exclusão e edição de funcionários com suas devidas permissões de uso.

Os requisitos funcionais foram representados sob a forma de um diagrama de casos de uso, apresentado na figura 6.

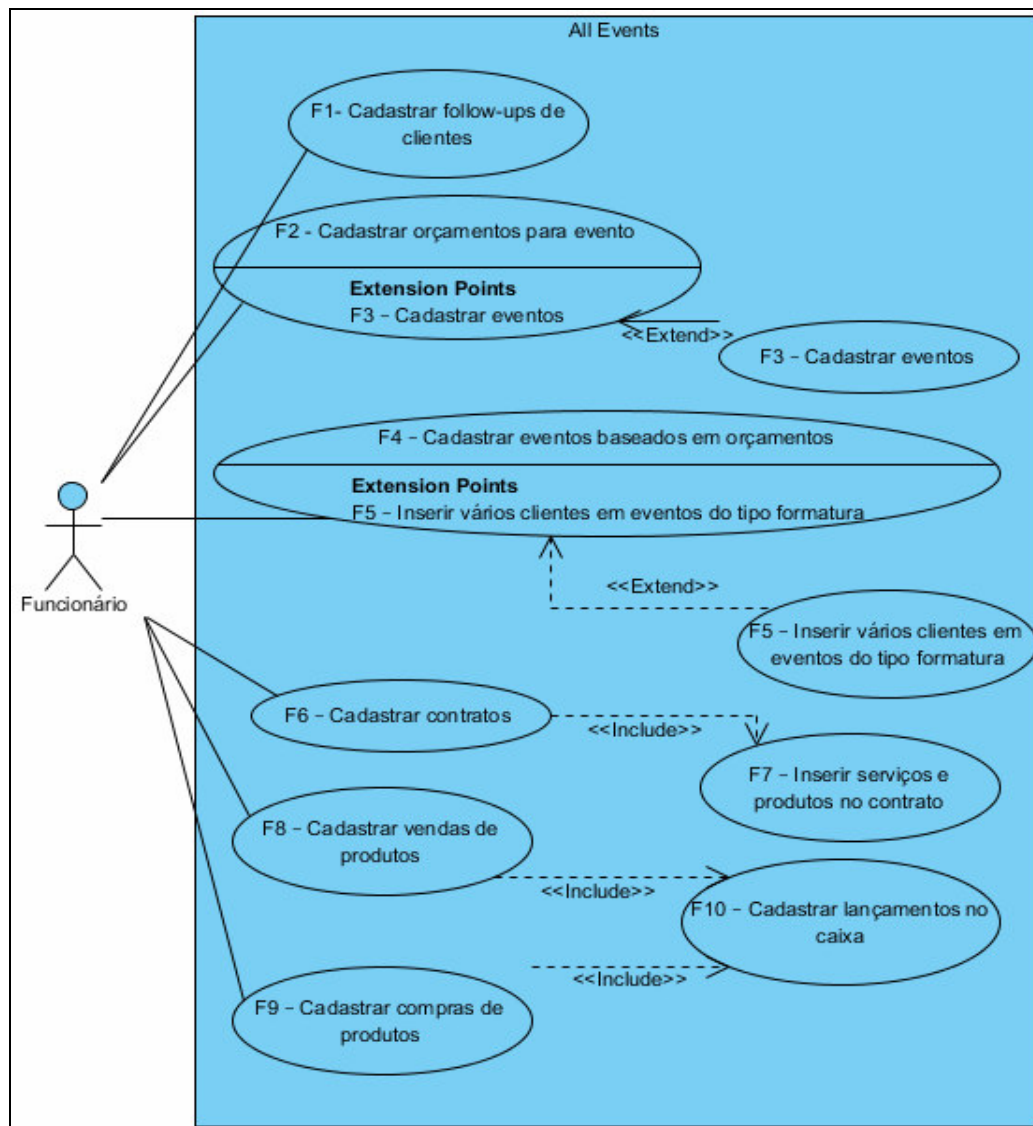


Figura 6 - Diagrama de casos de uso

Fonte: Autoria própria

A Figura 7 apresenta partes do diagrama de classes que compõem o sistema implementado neste projeto. Este diagrama é composto por algumas classes referentes a dados cadastrais do sistema e também dados que serão armazenados para o gerenciamento dos eventos.

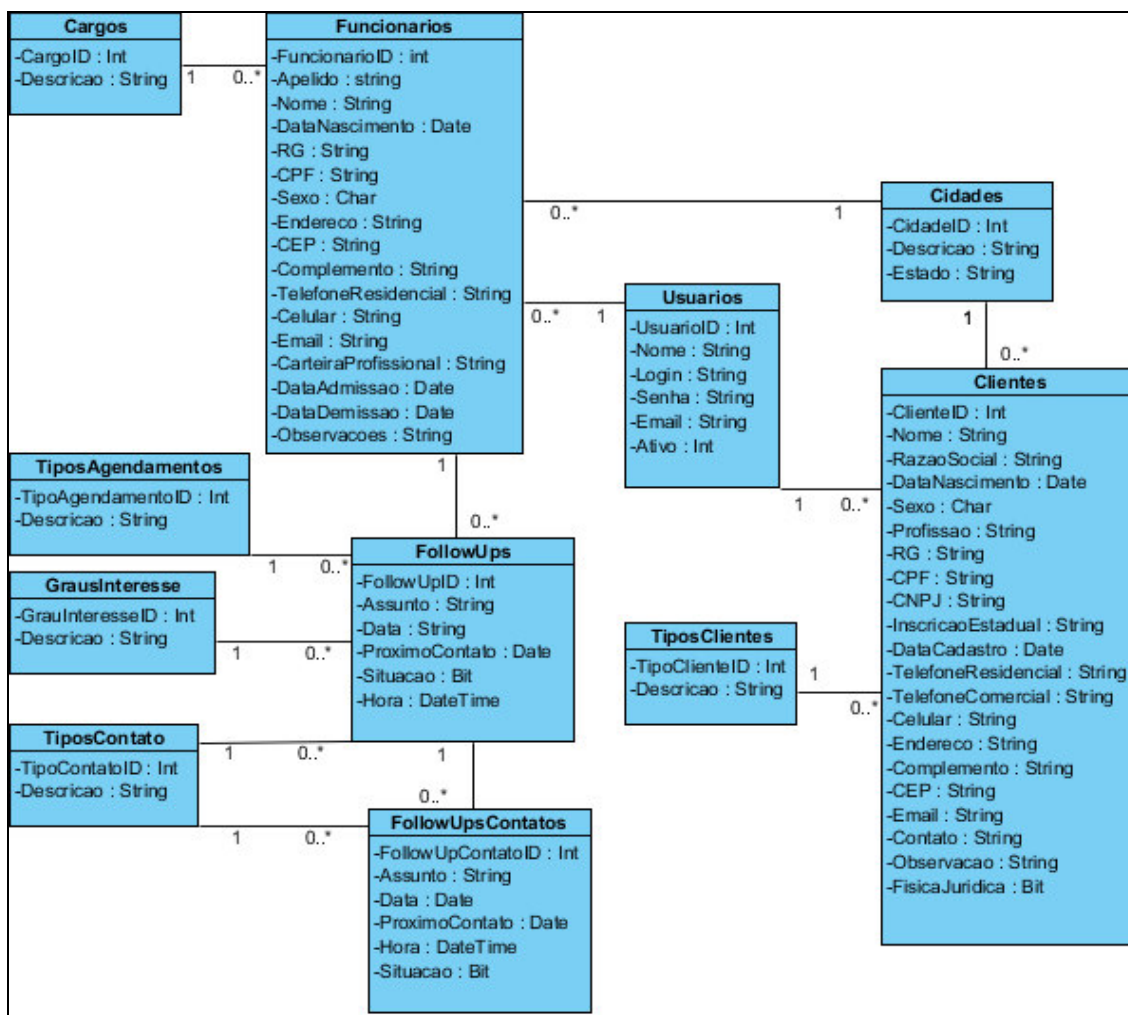


Figura 7 - Representação de parte do diagrama de classes

Fonte: Autoria própria

A Figura 8 apresenta o modelo E-R projetado.

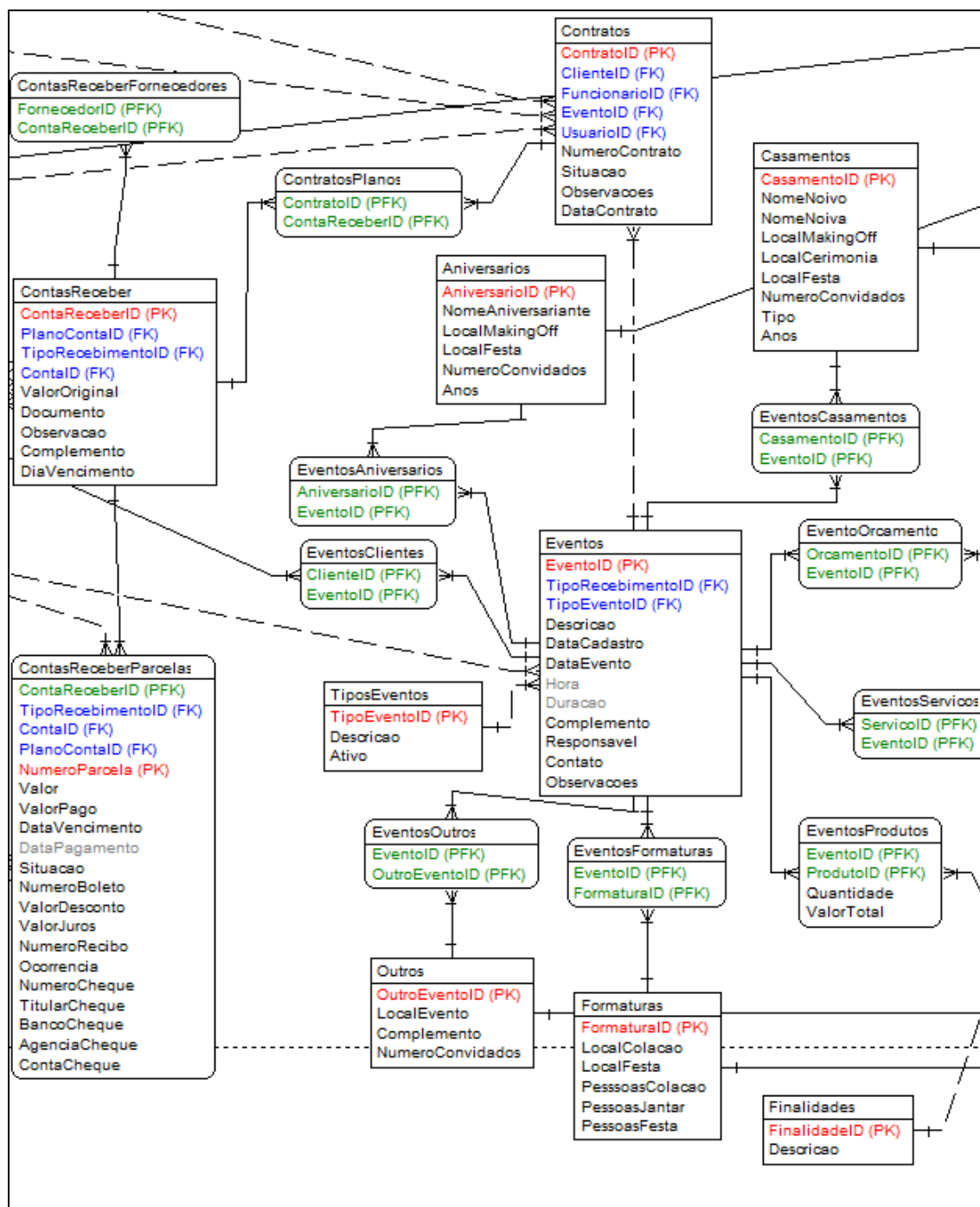


Figura 8 - Representação de parte do modelo entidade-relacionamento

Fonte: Autoria própria

As tabelas que possuem a nomenclatura composta pelo nome de duas tabelas, como por exemplo: *ContasReceberFornecedores* (ligação da tabela *ContasReceber* com *Fornecedores*), são tabelas geradas pelo tipo de relacionamento n para n entre duas tabelas.

4.3 Descrição do Sistema

A descrição do sistema apresentada nesta seção visa mostrar as funcionalidades e recursos das telas do sistema.

Na Figura 9 tem-se a tela principal que é apresentada para o usuário quando o sistema é iniciado. A partir dessa tela o usuário poderá logar-se no sistema.

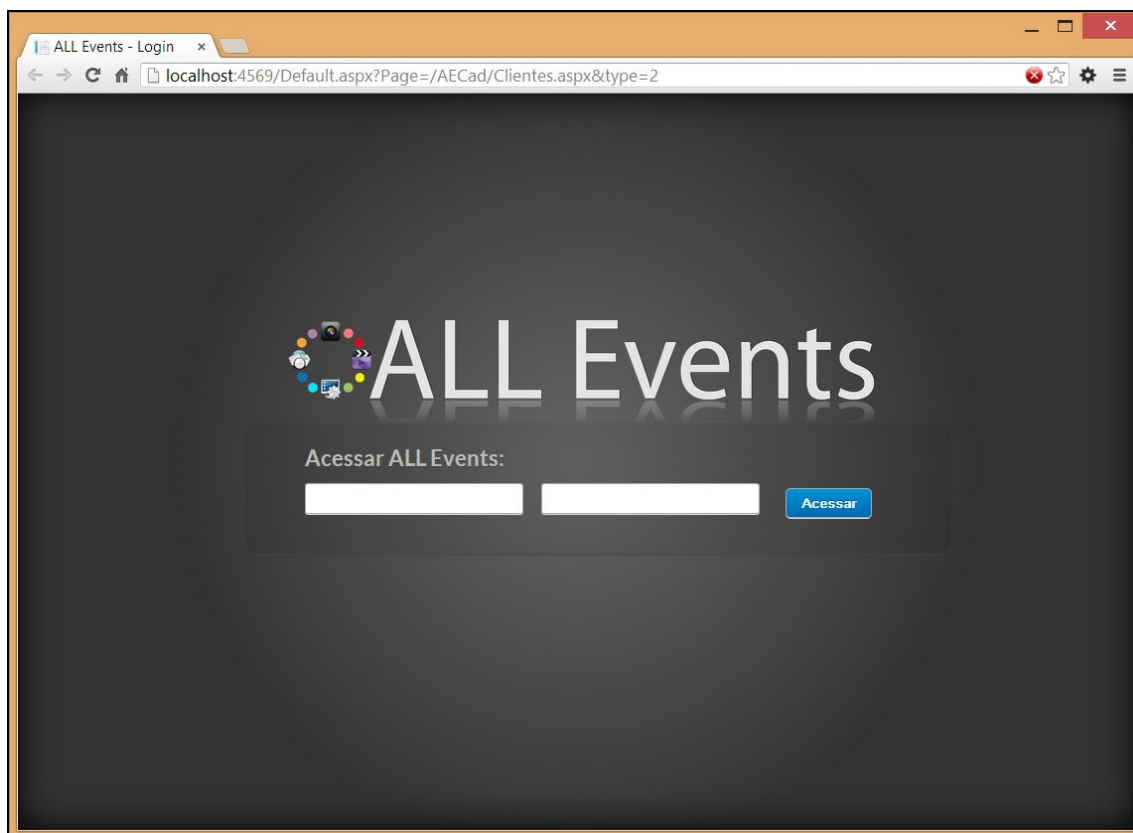


Figura 9 - Tela de inicialização e login

Fonte: Autoria própria

A Figura 10 mostra a *Master Page* (MP) do sistema. Todas as demais páginas do sistema são heranças da MP. A partir dessa tela o usuário logado tem acesso a todos os módulos do sistema por meio do menu rápido, localizado à esquerda da página ou por meio do menu principal que está no topo da página.

O menu rápido permite acesso instantâneo as principais funcionalidades do sistema. O logotipo da empresa fica localizado no topo esquerdo da página e poderá ser alterado de acordo com a necessidade do usuário. Abaixo do logotipo encontra-se a foto personalizável do usuário logado.

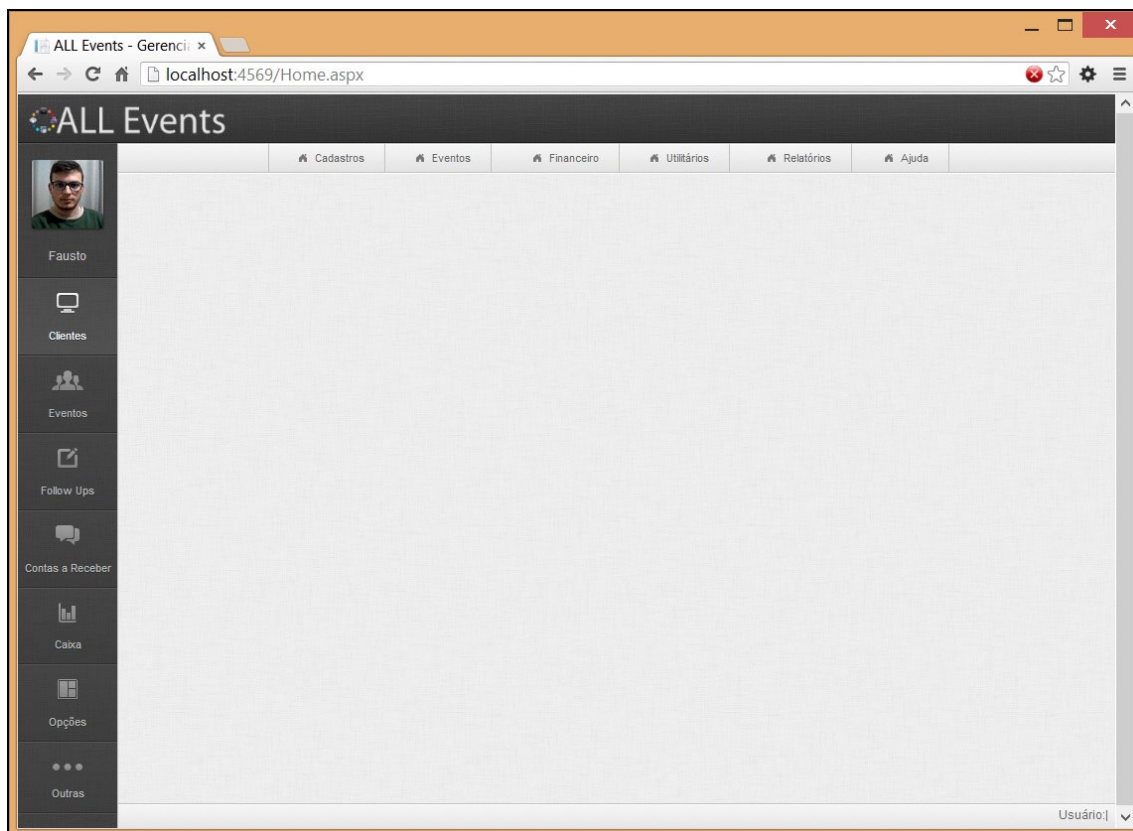


Figura 10 - Master page

Fonte: Autoria própria

A Figura 11 apresenta o cadastro da empresa que utiliza o sistema. Informações estas que serão utilizadas em recibos, cadastros e relatórios.

The screenshot shows a web browser window titled 'Dados da Empresa' with the URL 'localhost:4041/AECad/DadosEmpresa.aspx'. The page header features the 'ALL Events' logo and a navigation menu with items: Cadastros, Eventos, Financeiro, Utilitários, Relatórios, and Ajuda. On the left side, there is a vertical sidebar with icons and labels: Fausto (with a profile picture), Clientes, Eventos, Follow Ups, Contas a Receber, Caixa, Opções, and Outras. The main content area displays a form for entering company data. The form fields are as follows:

Nome	Razão Social	
All Events Soft	Rossi & Fonseca LTDA.	
C.N.P.J.	Inscrição Estadual	Inscrição Municipal
23.154.685/0001-32	23423.54-1	34564.546.1
Endereço		C.E.P.
Rua Governador Jorge Lacerda, 431		85501-546
Cidade	Bairro	
Pato Branco	Centro	
Telefone	Fax	E-mail
(46) 3254-1354	(46) 3597-9456	contato@allevents.com

A 'Salvar' button is positioned at the bottom left of the form.

Figura 11 - Tela de cadastro da empresa

Fonte: Autoria própria

Os campos numéricos como CNPJ (Cadastro Nacional de Pessoas Jurídicas) e CEP (Código de Endereçamento Postal), são campos que restringem a entrada de dados para entradas somente numéricas.

A Figura 12 apresenta a listagem dos clientes já cadastrados. Do lado esquerdo da listagem é possível selecionar um ou vários clientes através de um *checkbox*. Do lado esquerdo superior estão localizados os botões de inclusão de um novo cliente, apresentado na Figura 14 e exclusão de um ou mais clientes selecionados, apresentado na Figura 13.

ALL Events

Cadastros | Eventos | Financeiro | Utilitários | Relatórios | Ajuda

Cientes | Filtros Avançados

	Tipo	Status	Fís./Jur.	Nome/Razão Social	CPF/CNPJ	RG	Fone	Fone Com.	Celular	CEP	Usuário	Cadastro
<input type="checkbox"/>	Novo	Interessado	Física	Cicero Fonseca	651.896.518-96	8.645.487-2	(46) 8189-4964	(46) 8189-4964	(46) 9598-2687	84657-465	Fausto Rossi	08/08/2013
<input checked="" type="checkbox"/>	Novo	Ativo	Física	Fausto Daniel Rossi	061.638.139-52	9.682.413-0	(46) 3025-2918	(46) 3025-2918	(46) 9107-0570	85501-500	Fausto Rossi	07/08/2013
<input checked="" type="checkbox"/>	Novo	Ativo	Física	Geri Natalino Dutra	648.916.156-64	8.456.877-7					Fausto Rossi	11/08/2013
<input checked="" type="checkbox"/>	Novo	Inativo	Física	Leonan Milani	062.545.841-98	9.978.966-1	(46) 3225-9207	(46) 3225-9207	(46) 9916-7062	85501-449	Fausto Rossi	09/08/2013
<input type="checkbox"/>	Familiar	Ativo	Física	Maira Vanessa Da Cas	878.941.654-86	8.987.789-0	(46) 5618-6878	(46) 5618-6878	(16) 5135-1386		Fausto Rossi	11/08/2013
<input type="checkbox"/>	Novo	Ativo	Física	Omero Francisco Bertol	987.465.615-11	2.248.654-2					Fausto Rossi	11/08/2013

Primeira ◀ ▶ Última

Figura 12 - Tela exibindo o *grid* de cadastro de clientes

Fonte: Autoria própria

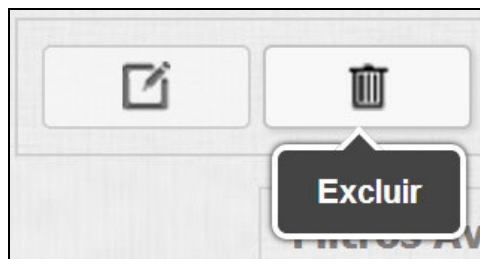


Figura 13 - Tela exibindo os botões incluir e excluir

Fonte: Autoria própria



Figura 14 - Tela exibindo os botões incluir e excluir

Fonte: Autoria própria

A listagem exibe um número limitado de registros evitando sobrecarga do servidor da aplicação, o que torna mais rápido a exibição dos dados, pois são carregados sob demanda. Se o número de registros a serem listados for superior ao limite configurado, ficarão visíveis botões numéricos identificando o número de páginas, possibilitando a navegação do usuário.

Ao clicar no botão Incluir, apresentado na Figura 14, será aberta, sobrepondo a listagem de clientes, uma nova tela para inclusão das informações referentes a um novo cliente, conforme apresentado na Figura 15.

The image shows a web browser window with the URL 'localhost:4041/AECad/Cientes.aspx'. The application is titled 'ALL Events'. A sidebar on the left contains navigation icons for 'Fausto', 'Clientes', 'Eventos', 'Follow Ups', 'Contas a Receber', 'Caixa', 'Opções', and 'Outras'. The main content area displays a 'Novo Cliente' form. The form is divided into sections: 'Dados Principais' (Name, Social Reason, Status, Type, C.P.F., R.G., Birth Date, Profession, Sex), 'Comunicação' (Residential, Commercial, Cellular, Fax phone numbers), and 'Emails' (Email address, Contact name). The form is currently filled with data for 'Geri Natalino Dutra'. At the bottom of the form are buttons for 'Salvar', 'Novo', 'Cancelar', and 'Sair'. In the background, a table of clients is visible, showing columns for 'usuário' and 'Cadastro' with dates ranging from 08/08/2013 to 11/08/2013.

Figura 15 - Tela de cadastro de clientes, aba principal

Fonte: Autoria própria

Para uma melhor visualização, as telas de cadastro, quando necessário serão organizadas em abas, dependendo da quantidade de informação. As abas permitem diminuir a quantidade de campos apresentadas em uma tela e agrupar campos de acordo com informações em comum. Campos com máscara foram utilizados a fim de facilitar a validação de dados. Os campos de data estão configurados para formatar a informação no padrão de data brasileiro ordenando a informação em dia, mês e ano (dd/mm/aaaa).

A segunda aba do cadastro de clientes reúne os campos que contém informação em comum, como os campos de Endereço apresentados na Figura 16. Na parte inferior da tela de cadastro estão localizados os botões Salvar, Novo e Sair. O botão Salvar armazena as informações no banco de dados, o botão Novo reinicia o valor dos campos para que um novo cadastro seja iniciado e o botão Sair fecha a tela em questão.

The screenshot shows a web application interface for client registration. The main window is titled 'Novo Cliente' and has two tabs: 'Dados do cliente' and 'Endereço/Obs.'. The 'Endereço/Obs.' tab is active. The form contains the following fields:

- Endereço:**
 - Rua/Avenida:** Text input with value 'Rua Saudade do Iguaçu, 56'.
 - Complemento:** Text input with value 'Segunda casa da rua'.
- Cidade:** Dropdown menu with value 'Pato Branco'.
- Bairro:** Text input with value 'Vila Izabel'.
- C.E.P.:** Text input with value '85501-864'.

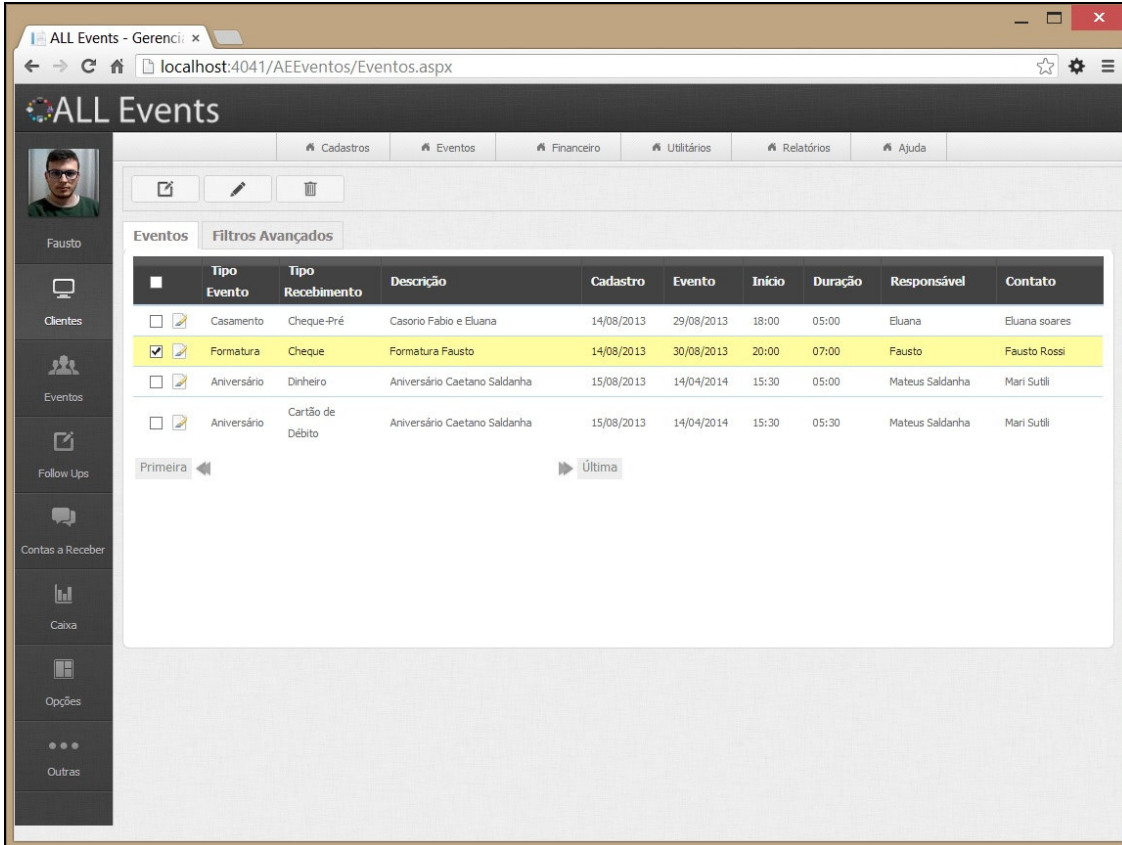
Below these fields is a text area for 'Observação' containing the text: 'Em primeiro contato, Géri disse que está interessado em fechar um pacote de fotos para o casamento do seu filho'.

At the bottom of the form are four buttons: 'Salvar', 'Novo', 'Cancelar', and 'Sair'. The background shows a list of clients with columns for 'usuário' and 'Cadastro'.

Figura 16 - Tela de cadastro de clientes, aba Endereço/Obs

Fonte: Autoria própria

A Figura 17 apresenta a listagem de eventos já cadastrados.



The screenshot displays the 'ALL Events' web application. The browser address bar shows 'localhost:4041/AEEEventos/Eventos.aspx'. The application has a dark header with the 'ALL Events' logo and a navigation menu with options: Cadastros, Eventos, Financeiro, Utilitários, Relatórios, and Ajuda. A sidebar on the left contains a user profile for 'Fausto' and a menu with icons for 'Clientes', 'Eventos', 'Follow Ups', 'Contas a Receber', 'Caixa', 'Opções', and 'Outras'. The main content area features a 'Eventos' tab and 'Filtros Avançados'. Below this is a table with the following data:

	Tipo Evento	Tipo Recebimento	Descrição	Cadastro	Evento	Início	Duração	Responsável	Contato
<input type="checkbox"/>	Casamento	Cheque-Pré	Casorio Fabio e Eluana	14/08/2013	29/08/2013	18:00	05:00	Eluana	Eluana soares
<input checked="" type="checkbox"/>	Formatura	Cheque	Formatura Fausto	14/08/2013	30/08/2013	20:00	07:00	Fausto	Fausto Rossi
<input type="checkbox"/>	Aniversário	Dinheiro	Aniversário Caetano Saldanha	15/08/2013	14/04/2014	15:30	05:00	Mateus Saldanha	Mari Sutili
<input type="checkbox"/>	Aniversário	Cartão de Débito	Aniversário Caetano Saldanha	15/08/2013	14/04/2014	15:30	05:30	Mateus Saldanha	Mari Sutili

Navigation controls include 'Primeira' and 'Última' buttons.

Figura 17 - Tela exibindo o grid de eventos

Fonte: Autoria própria

Ao clicar em Inserir, para inserção de um novo evento será aberta a tela de cadastro, conforme apresentado na Figura 18. O usuário poderá selecionar o tipo de evento através dos *radio buttons* localizados no canto superior direito da tela, na aba principal também poderá preencher os campos referentes às informações semelhantes aos tipos de eventos.

The screenshot displays the 'ALL Events' web application interface. A modal window titled 'Novo Evento' is open, showing the 'Dados do Evento' form. The form is divided into several sections:

- Dados Principais:**
 - Descrição:** Bodas de Ouro Géni e Marli
 - Tipo Recebimento:** Cheque-Pré
 - Data Cadastro:** 20/09/2013
 - Data Evento:** 25/10/2013
 - Hora:** 13:30
 - Duração:** 06:30
 - Responsável:** Marli Dutra
- Complemento:** Será comemorado também o Aniversário de 3 anos do Segundo filho
- Contato:** Marli Dutra
- Observações:** O contrato apenas será firmado, com a autorização da Marli Dutra.

At the bottom of the modal, there are three buttons: 'Salvar', 'Novo', and 'Sair'. The background shows the application's navigation menu with options like 'Clientes', 'Eventos', 'Follow Ups', 'Contas a Receber', 'Caixa', 'Opções', and 'Outras'.

Figura 18 - Tela de cadastro de eventos, aba principal

Fonte: Autoria própria

A Figura 19 apresenta a segunda aba do cadastro de eventos, com informações particulares a cada tipo de evento, os campos serão exibidos de acordo com o tipo de evento selecionado.

The screenshot displays the 'ALL Events' web application interface. A modal window titled 'Novo Evento' is open, showing the 'Evento' tab. The form is for a wedding (Casamento) and includes the following fields:

- Nome do Noivo:** Géri Natalino Dutra
- Nome da Noiva:** Marii Dutra
- Local do Making Off:** Stúdio Moriá
- Local da Cerimônia:** Igreja Matriz
- Local da Festa:** Clube Pinheiros
- Nº Convidados:** 860
- Tipo de Casamento:** Bodas
- Ano(s)?** 50

Buttons at the bottom of the modal are 'Salvar', 'Novo', and 'Sair'. The background shows a sidebar with navigation options like 'Clientes', 'Eventos', and 'Follow Ups', and a top navigation bar with 'Cadastros', 'Eventos', 'Financeiro', 'Utilitários', 'Relatórios', and 'Ajuda'.

Figura 19 - Tela de cadastro de eventos, aba Evento

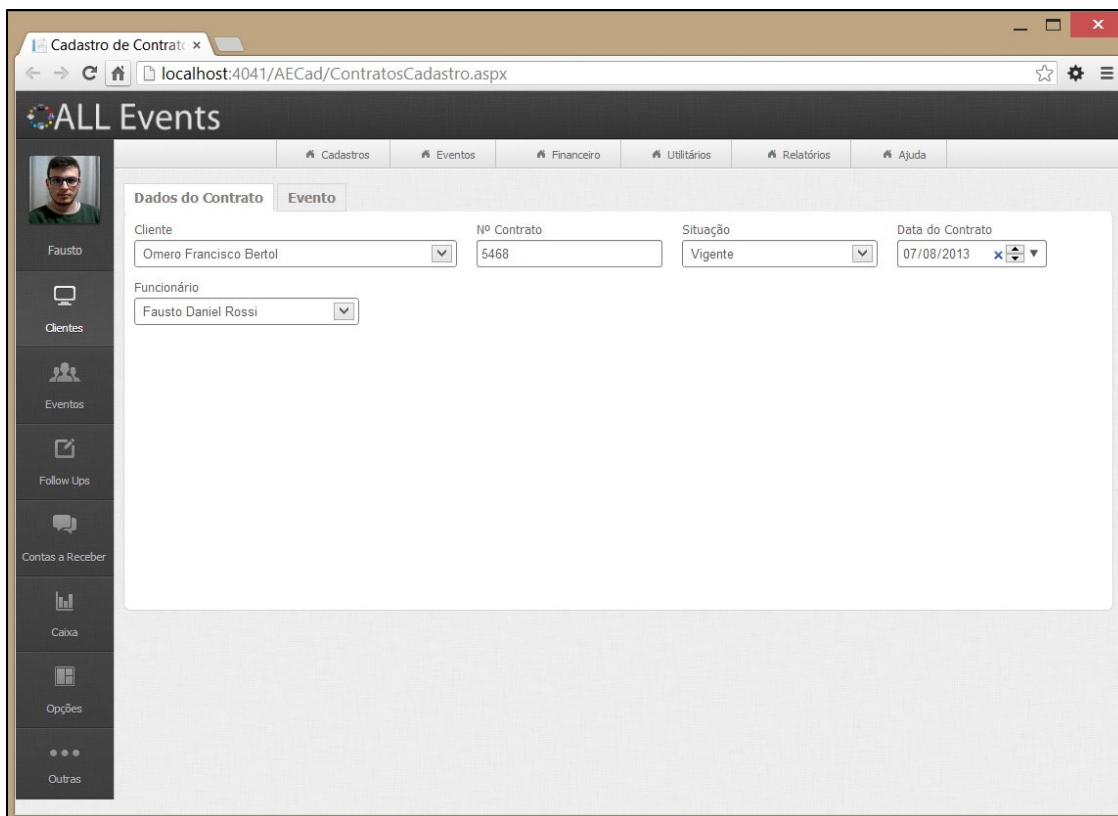
Fonte: Autoria própria

A Figura 20 apresenta a tela contendo os campos referentes ao tipo de evento Formatura.

The screenshot shows a web application interface for managing events. The main window is titled 'ALL Events - Gerenci...' and the URL is 'localhost:4041/AEEventos/Eventos.aspx'. The application has a navigation menu with options like 'Cadastros', 'Eventos', 'Financeiro', 'Utilitários', 'Relatórios', and 'Ajuda'. A sidebar on the left shows a user profile for 'Fausto' and various management tools. A modal window titled 'Novo Evento' is open, showing a form for creating a new event. The form has two tabs: 'Dados do Evento' and 'Evento'. The 'Evento' tab is active, and the event type is set to 'Formatura' (Graduation). The form contains several input fields: 'Local da Colação' (University Tecnológica Federal do Paraná (UTFPR)), 'Local da Festa' (Clube Pinheiros), 'Nº Pessoas Colação' (280), 'Nº Pessoas Jantar' (950), and 'Nº Pessoas Baile' (1230). There are also buttons for 'Salvar', 'Novo', and 'Sair'.

Figura 20 - Tela de cadastro de eventos, aba Evento demonstrando os campos alterados ao selecionar o tipo de evento Formatura
Fonte: Autoria própria

A Figura 21 apresenta a tela de cadastro de contratos, também organizadas em abas. Na aba em questão o usuário poderá inserir as informações principais referentes ao contrato como cliente, data, situação e número de contrato.



Cadastro de Contratos

localhost:4041/AECad/ContratosCadastro.aspx

ALL Events

Cadastros | Eventos | Financeiro | Utilitários | Relatórios | Ajuda

Dados do Contrato | Evento

Cliente: Omerno Francisco Bertol

Nº Contrato: 5468

Situação: Vigente

Data do Contrato: 07/08/2013

Funcionário: Fausto Daniel Rossi

Fausto

Clientes

Eventos

Follow Ups

Contas a Receber

Caixa

Opções

Outras

Figura 21 - Tela de cadastro de contratos

Fonte: Autoria própria

A Figura 22 apresenta a aba Formandos que é exibida quando o tipo do evento selecionado for do tipo Formatura. Nessa aba são adicionados os formandos pertencentes ao evento em questão.

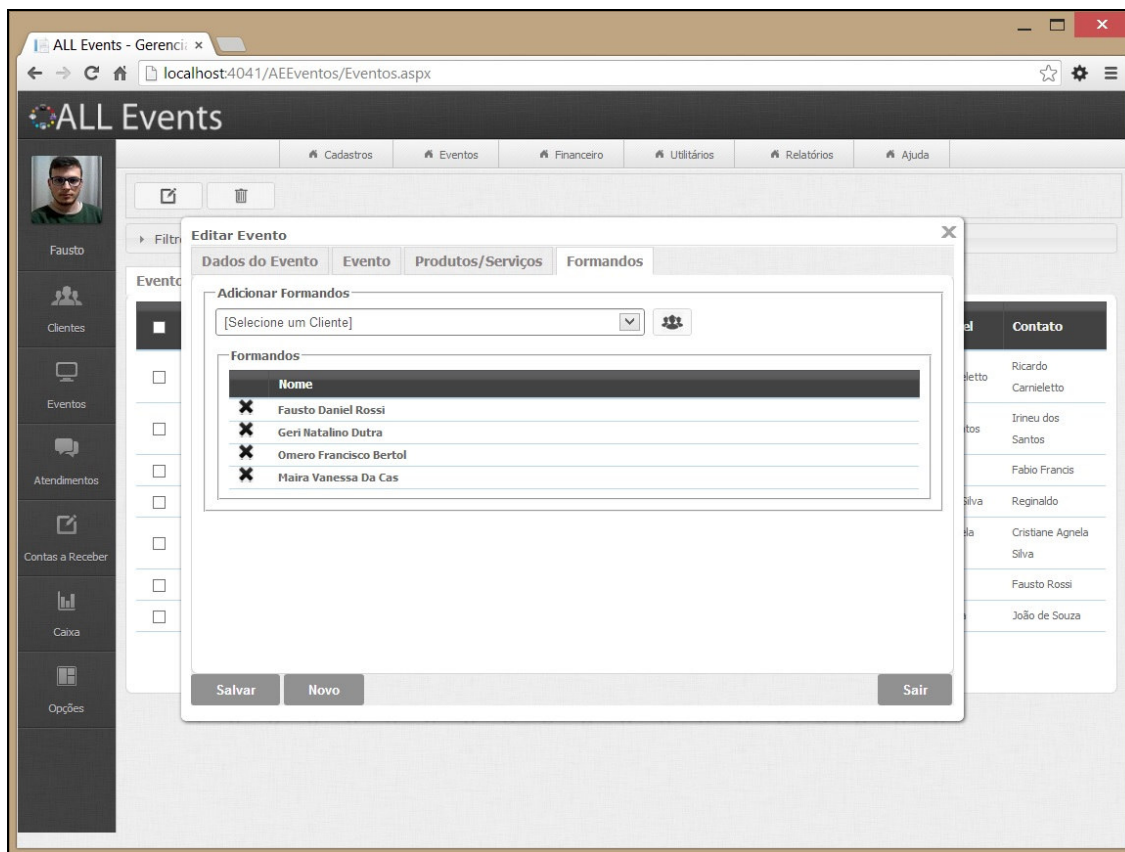


Figura 22 - Tela de cadastro de eventos, aba Formandos demonstrando a inclusão dos formandos nos eventos do tipo Formatura
Fonte: Autoria própria

A Figura 23 apresenta a tela de cadastro de contratos, também organizadas em abas. Na primeira aba são inseridas as informações principais referentes ao contrato como cliente, data, situação e número de contrato. A segunda aba é apresentada na Figura 24.

Cadastro de Contratos x

localhost:4041/AECad/ContratosCadastro.aspx

ALL Events

Cadastros | Eventos | Financeiro | Utilitários | Relatórios | Ajuda

Dados do Contrato | Evento

Cliente: Omerno Francisco Bertol

Nº Contrato: 5468

Situação: Vigente

Data do Contrato: 07/08/2013

Funcionário: Fausto Daniel Rossi

Fausto

Clientes

Eventos

Follow Ups

Contas a Receber

Caixa

Opções

Outras

Figura 23 - Tela de cadastro de contratos

Fonte: Autoria própria

A Figura 24 apresenta o evento que será relacionado ao contrato. O usuário pode selecionar o tipo de evento e em um *DropDownList* serão carregados todos os eventos correspondentes ao tipo selecionado, permitindo ao usuário a seleção do evento desejado. Após a seleção as informações referentes ao evento selecionado serão carregadas apenas para visualização do usuário, não permitindo alterações.

The screenshot displays the 'ALL Events' web application interface. The browser address bar shows 'localhost:4041/AEEventos/ContratosCadastro.aspx'. The application header includes the 'ALL Events' logo and a navigation menu with 'Cadastros', 'Eventos', 'Financeiro', 'Utilitários', 'Relatórios', and 'Ajuda'. The main content area is titled 'Dados do Contrato' and 'Evento'. The 'Evento' section contains a dropdown menu for 'Tipo do Evento' with 'Casamento' selected. Below this is a text input field for 'Descrição do Evento' containing 'Casamento Fábio e Eluana Mattei'. To the right is a dropdown for 'Tipo do Recebimento' with 'Cheque-Pré' selected. Further down are input fields for 'Data do Evento' (17/06/2014), 'Hora' (19:30), and 'Duração' (07:00). At the bottom, there are two tables: 'Relação de Produtos' and 'Relação de Serviços'. The 'Relação de Produtos' table lists 'Album 25x60', 'Papel Picado', and 'Foto 60x90'. The 'Relação de Serviços' table lists 'Telão', 'Projetor', 'Ponto de Luz', and 'Filmagem 2 Câmeras'. A sidebar on the left shows the user profile 'Fausto' and navigation icons for 'Clientes', 'Eventos', 'Atendimentos', 'Contas a Receber', 'Caixa', and 'Opções'.

Figura 24 - Tela de cadastro de contratos demonstrando a aba Eventos

Fonte: Autoria própria

A Figura 25 apresenta os filtros à disposição do usuário para a geração do relatório de cadastro de eventos. É possível filtrar o relatório pela descrição de evento, tipo de recebimento, tipo de evento e datas de realização e cadastro.

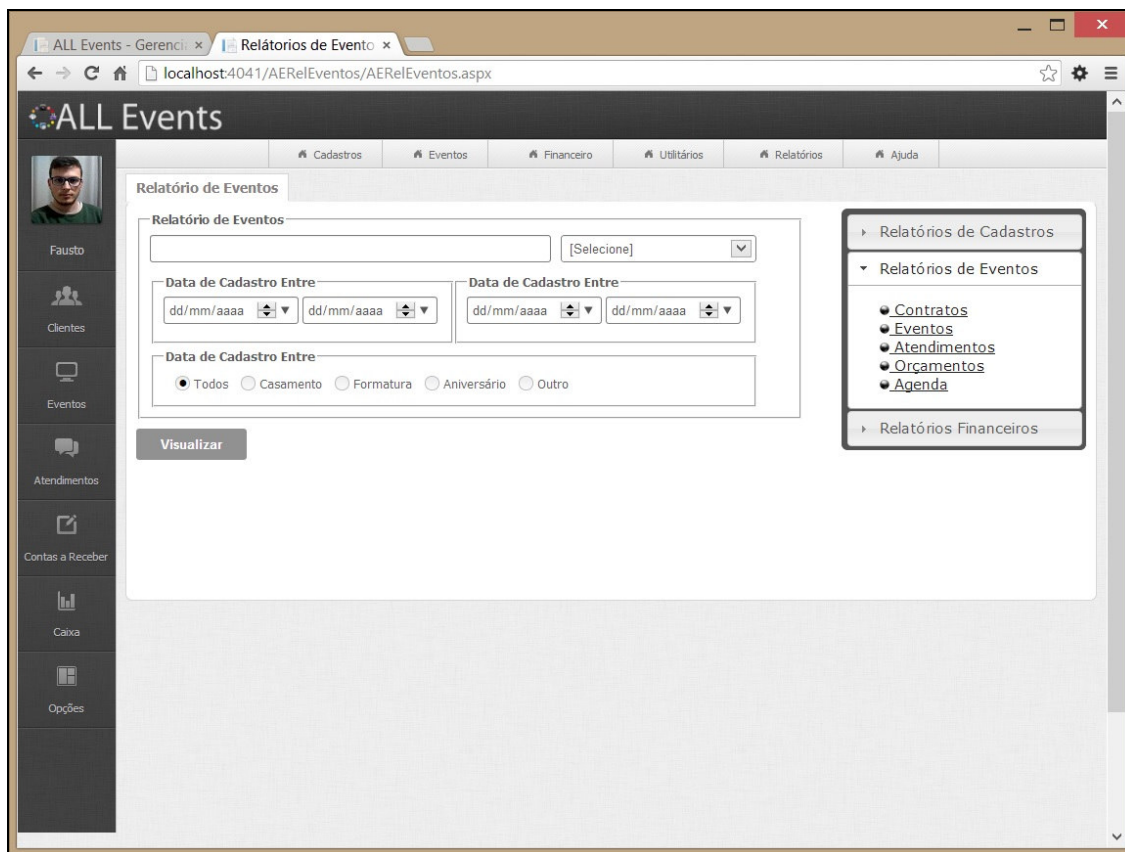


Figura 25 - Tela de filtros do relatório de cadastros de eventos

Fonte: Autoria própria

A Figura 26 apresenta o relatório gerado após a seleção dos filtros. O relatório foi gerado pelo Report Viewer e é possível exportá-lo para arquivos nos seguintes formatos: .XLS, .PDF e .DOCX. Também é possível localizar uma palavra ou frase utilizando o campo de pesquisa localizado no topo do relatório e navegar entre as páginas quando o relatório possuir mais de uma.

Tipo	Descrição	Data Evento	Responsável	Contato
Casamento	Casamento Fábio e Eluana Mattei	17/06/2014 00:00:00	Eluana Soares	Fabio Mattei
Formatura	Formatura Fausto Daniel Rossi	27/09/2013 00:00:00	Fausto Rossi	Fausto Rossi
Aniversário	Aniversário 3º Ano Sophia Da Cas	18/03/2014 00:00:00	Fernando Da Cas	Andreia Copatti
Casamento	Casamento Leonan Milani e Gabriela	13/11/2014 00:00:00	Gabriela Veronese	Leonan Milani
Aniversário	Aniversário Amanda Milani	16/07/2015 00:00:00	Gabriela Milani	Leonan Milani
Outro	Conferência Regional OAB	30/08/2013 00:00:00	Marcos Antônio	Marcos Antônio
Outro	Jantar Dançante Preto e Branco	30/08/2013 00:00:00	João de Souza	João de Souza
Outro	Baile das Secretárias	16/01/2014 00:00:00	Maria Almeida	Maria Almeida
Formatura	Formatura Direto Mater Dei	31/08/2013 00:00:00	Cristiane Agnela Silva	Cristiane Agnela Silva
Casamento	Bodas de Ouro Tere e Celso Da Cas	25/12/2014 00:00:00	Celso Da Cas	Celso Da Cas
Formatura	Formatura Administração UTFPR	25/12/2015 00:00:00	Fabio Francis	Fabio Francis
Formatura	Formatura Análises Clínicas	20/11/2013 00:00:00	Reginaldo da Silva	Reginaldo
Outro	Festa Rotary P. Bco Araucária	19/12/2014 00:00:00	Ricardo Carnieletto	Ricardo Carnieletto
Outro	Confraternização Sponte Informática	25/10/2013 00:00:00	Reginaldo da Silva	Reginaldo da Silva
Casamento	Casamento Igor e Dorotéia	20/11/2014 00:00:00	Igor Borges	Igor Borges
Casamento	Casamento Cicero e Caroline	20/11/2013 00:00:00	Caroline Fonseca	Caroline
Formatura	Formatura 3ºão Colégio Mater Dei	02/02/2015 00:00:00	Irineu dos Santos	Irineu dos Santos
Outro	Festa 15 anos Sponte	14/09/2013 00:00:00	Reginaldo	Reginaldo
Outro	Cerimonia de Abertura	06/09/2013 00:00:00	João Pereira	João Pereira

All Events Softwares - 08/09/2013 18:37

Figura 26 - Relatório de cadastro de eventos

Fonte: Autoria própria

4.4 Implementação do Sistema

Nesta seção são apresentados exemplos da codificação do sistema, com o objetivo de mostrar o uso das tecnologias adotadas para esta finalidade.

A Listagem 1 apresenta a *string* de conexão do sistema com o banco de dados. Esse trecho de código encontra-se no arquivo “web.config”. Esse arquivo armazena as configurações gerais da aplicação. Essa conexão listada é gerada automaticamente quando é realizado o mapeamento do banco de dados.

```

26 <connectionStrings>
27 <add name="EventosEntities" connectionString="metadata=res://*/Eventos.csdl|res://*/
28 Eventos.ssd|res://*/Eventos.msl;provider=System.Data.SqlClient;
29 provider connection string="data source=ROSSI\SQLEXPRESS;initial catalog=Eventos;
30 persist security info=True;user id=sa;password=13Ae9v7FC31rf;
31 MultipleActiveResultSets=True;App=EntityFramework";
32 providerName="System.Data.EntityClient" />
33 <add name="EventosConnectionString" connectionString="Data Source=ROSSI\SQLEXPRESS;
34 Initial Catalog=Eventos;Persist Security Info=True;User ID=sa;Password=13Ae9v7FC31rf;
35 MultipleActiveResultSets=True;Application Name=EntityFramework"
36 providerName="System.Data.SqlClient" />
37 </connectionStrings>

```

Listagem 1 - String de conexão

A Listagem 2 apresenta a classe “EventosEntities” que é responsável por fazer a conexão com o banco de dados utilizando a *string* de conexão mostrada na Listagem 1. A linha 18 passa por parâmetro o nome da conexão para que seja feita a comunicação do banco de dados com a aplicação. A partir da linha 25, estão mapeadas as classes que serão utilizadas pelo sistema para as operações de cadastro.

```

14 Partial Public Class EventosEntities
15     Inherits DbContext
16
17     Public Sub New()
18         MyBase.New("name=EventosEntities")
19     End Sub
20
21     Protected Overrides Sub OnModelCreating(modelBuilder As DbModelBuilder)
22         Throw New UnintentionalCodeFirstException()
23     End Sub
24
25     Public Property Aniversarios() As DbSet(Of Aniversario)
26     Public Property Bancos() As DbSet(Of Banco)
27     Public Property Cargos() As DbSet(Of Cargo)
28     Public Property Casamentos() As DbSet(Of Casamento)
29     Public Property Cidades() As DbSet(Of Cidade)
30     Public Property Clientes() As DbSet(Of Cliente)

```

Listagem 2 - Classe responsável pela conexão com o banco de dados

A Listagem 3 apresenta o código que recupera os dados no banco de dados e os insere no *AspxGridView* para visualização do usuário. O método “CarregaGridEventos” recebe por parâmetro três variáveis que são responsáveis pela paginação dos registros. A paginação é feita para otimizar o retorno da informação para o usuário, pois serão carregados apenas 12 registros para cada página.

```

27 Public Sub CarregaGridEventos(Optional ByVal lCalcularTotal As Boolean = True, _
28                               Optional ByVal nFirstRecord As Integer = 0, _
29                               Optional ByVal nLastRecord As Integer = 0)
30     Dim evento As New Evento
31     Using context As New EventosEntities
32
33         If nLastRecord = 0 Then
34             nLastRecord = 12
35         End If
36
37         Dim eventos = (From e In context.Eventos
38                       Join te In context.TiposEventos On te.TipoEventoID Equals e.TipoEventoID
39                       Join tr In context.TiposRecebimentos On
40                       tr.TipoRecebimentoID Equals e.TipoRecebimentoID
41                       Select e.EventoID, TipoEvento = te.Descricao, TipoRecebimento = tr.Descricao,
42                       e.Descricao, e.DataCadastro, e.DataEvento, e.Hora, e.Duracao, e.Responsavel) _
43                       .OrderBy(Function(e) e.Descricao).Skip(nFirstRecord).Take(nLastRecord).ToList()
44
45         grdEventos.DataSource = eventos
46         grdEventos.DataBind()
47
48         Dim eveCount = (From e In context.Eventos).Count()
49         pager1.ItemCount = eveCount
50
51         updGrid.Update()
52     End Using
53 End Sub

```

Listagem 3 - Código que carrega os dados no AspxGridView

Ainda na Listagem 3, na linha 30, é instanciada a classe do tipo “Evento”, por meio da qual são carregados os métodos e objetos da classe. A linha 31 instancia a classe “EventosEntities” que é responsável pela abertura da conexão com o banco de dados. Entre as linhas 37 e 43 apresenta-se uma expressão *lambda* utilizada pelo Entity Framework para a busca das informações no banco de dados que recebe por parâmetro os valores para a realização da paginação. A lista de registros retornados pela expressão *lambda* é armazenada em uma variável do tipo *System.Collections.Generic.List*. A variável com os registros armazenados é atribuída para o *DataSource* do AspxGridView, como mostra a linha 45.

A Listagem 4 apresenta o código responsável por chamar uma tela para novo cadastro de Eventos. O componente responsável por permitir a abertura do formulário dentro da página na qual ele foi chamado é o *HighSlide*.

```

23 Private Sub btnAdicionar_Click(sender As Object, e As EventArgs) Handles btnAdicionar.Click
24     ScriptManager.RegisterStartupScript(Me, Me.GetType(), "cadeve", _
25     "openWindow(1, 'cadeve', 'Novo Evento', 'EventosCadastro.aspx', " & _
26     "false, 500, 780, true, false, false, null);", True)
27 End Sub

```

Listagem 4 - Código responsável por chamar a tela de cadastro de eventos

A Listagem 5 apresenta o código JavaScript responsável pela abertura da tela de cadastros em *Modal*. Essa forma de abertura de telas internamente na página

que a chamou é possível, pois foi utilizado o componente HighSlide. Ao clicar no botão de inclusão ou edição as funções apresentadas abaixo são chamadas para abrir a nova tela de cadastro em uma janela modal.

```

158 function openWindow(type, id, title, url, modal, height, width, center, resize, scroll, executeOnClose) {
159     var params;
160
161     return openWebWindow2(id, title, url, modal, width, height, resize);
162 }
163
164 function openWebWindow2(id, title, url, modal, width, height, resize) {
165     var wrapClass = 'draggable-header';
166     var nOpacity = modal ? 0.7 : 0; //SOMENTE é válido para páginas que utilizam Highslide FULL
167
168     if (!resize)
169         wrapClass += ' no-footer';
170
171     hs.htmlExpand(id, null, {
172         objectType: 'iframe', outlineType: 'rounded-white', wrapperClassName: wrapClass,
173         outlineWhileAnimating: false, preserveContent: false, width: width, height: height,
174         minWidth: width, minHeight: height, headingText: title, src: url, dimmingOpacity: nOpacity
175     });
176 }

```

Listagem 5 - Código para abertura da tela de cadastros com efeito modal

A Listagem 6 apresenta a função de gravação dos eventos no banco de dados. O mesmo recebe por parâmetro as informações preenchidas no formulário de cadastro, em seguida é instanciado um novo objeto da classe Evento e do contexto da entidade de conexão com o BD “EventosEntities”.

Na linha 59 é feita a verificação de inclusão ou alteração do cadastro, se o evento a ser gravado é uma alteração, então é recuperada a chave do registro e também o valor dos seus atributos. Entre as linhas 64 e 75, os atributos da classe Evento recebem os valores recebidos por parâmetro do formulário. Na linha 77 é feita novamente a verificação de inclusão ou alteração, caso seja uma alteração é apenas gravado o objeto com as alterações, do contrário o objeto é adicionado à entidade Eventos e então é gravada na base de dados.

Ao final da inclusão/alteração uma variável de retorno recebe a chave do registro para ser passado por parâmetro nas inclusões das dependências do evento.

```

51 Public Function GravaEvento(Tipo As Byte, EventoID As Long, TipoRecebimentoID As Integer, _
52     TipoEventoID As Integer, Descricao As String, DataCadastro As Date, _
53     DataEvento As Date, Hora As String, Duracao As String, _
54     Complemento As String, Responsavel As String, Contato As String, _
55     Observacoes As String) As Long
56
57     Dim evento As New Evento, db As New EventosEntities, nRet As Long = 0
58
59     If Tipo = 1 Then
60         Dim nEventoID As Integer = Convert.ToInt32(EventoID)
61         evento = db.Eventos.SingleOrDefault(Function(e) e.EventoID = nEventoID)
62     End If
63
64     With evento 'cria/atualiza os dados do objeto
65         .TipoRecebimentoID = TipoRecebimentoID
66         .TipoEventoID = TipoEventoID
67         .Descricao = Descricao
68         .DataCadastro = DataCadastro
69         .DataEvento = DataEvento
70         .Hora = "1900-01-01 " & Hora
71         .Duracao = "1900-01-01 " & Duracao
72         .Complemento = Complemento
73         .Responsavel = Responsavel
74         .Contato = Contato
75         .Observacoes = Observacoes
76
77         If Tipo = 1 Then 'persiste as informação no banco de dados
78             db.SaveChanges()
79         Else 'cria um novo contexto
80             db.Eventos.Add(evento)
81             db.SaveChanges()
82         End If
83         nRet = evento.EventoID
84     End With
85
86     Return nRet
87 End Function

```

Listagem 6 - Método de inclusão/alteração dos cadastros de eventos

Na Listagem 7 está o código responsável por fazer a paginação do AspxGridView. Na linha 19 é recuperada a página selecionada pelo usuário e na linha seguinte é passada ao componente paginador a informação da página destino. A linha 22 chama a função de carregamento do AspxGridView passando por parâmetro o primeiro registro da página selecionada pelo usuário e o total de registros dessa página.

```

15 Protected Sub pager_Command(ByVal sender As System.Object, _
16     ByVal e As System.Web.UI.WebControls.CommandEventArgs) _
17     Handles pager1.Command
18
19     Dim currnetPageIndx As Int32 = CType(e.CommandArgument, Int32)
20     pager1.CurrentIndex = currnetPageIndx
21
22     CarregaGridEventos(False, (pager1.CurrentIndex - 1) * pager1.PageSize, _
23         (pager1.CurrentIndex * pager1.PageSize) + 1)
24
25     grdEventos.DataBind()
26 End Sub

```

Listagem 7 - Método responsável pela paginação do AspxGridView

A Listagem 8 apresenta o código responsável pelo carregamento das informações gravadas no banco de dados, no formulário de cadastro. Na linha 361 é verificado o tipo de evento a ser carregado, para que sejam buscadas as informações da respectiva tabela. Nas linhas 379 e 380 são carregados os `AspxGridView` dos produtos e serviços pertencentes ao evento.

```

343 Protected Sub CarregaEvento(ByVal nEventoID As Long)
344     Dim evento As New Evento, db As New EventosEntities, nRet As Long = 0
345
346     evento = db.Eventos.SingleOrDefault(Function(e) e.EventoID = nEventoID)
347     hdnEventoID.Value = evento.EventoID
348
349     txtDescricao.Text = evento.Descricao
350     cmbTipoRecebimento.Selectedvalue = evento.TipoRecebimentoID
351     txtDataCadastro.Text = Format(evento.DataCadastro, "yyyy-MM-dd")
352     txtDataEvento.Text = Format(evento.DataEvento, "yyyy-MM-dd")
353     txtHora.Text = Format(evento.Hora, "HH:mm")
354     txtDuracao.Text = Format(evento.Duracao, "HH:mm")
355     txtResponsavel.Text = evento.Responsavel
356     txtComplemento.Text = evento.Complemento
357     txtContato.Text = evento.Contato
358     txtObservacoes.Text = evento.Observacoes
359     rdbEvento.Selectedvalue = Val(evento.TipoEventoID)
360
361     Select Case Val(evento.TipoEventoID)
362     Case 1
363         CarregaCasamento(nEventoID)
364         DesabilitaPanel(evento.panelEvento.panelCasamento)
365     Case 2
366         CarregaFormatura(nEventoID)
367         CarregaClientesFormaturas(nEventoID)
368         CarregaComboClientes()
369         tabFormandos.Visible = True
370         DesabilitaPanel(evento.panelEvento.panelFormatura)
371     Case 3
372         CarregaAniversario(nEventoID)
373         DesabilitaPanel(evento.panelEvento.panelAniversario)
374     Case 4
375         CarregaOutro(nEventoID)
376         DesabilitaPanel(evento.panelEvento.panelOutro)
377     End Select
378
379     CarregaProdutos(nEventoID)
380     CarregaServicos(nEventoID)
381 End Sub

```

Listagem 8 - Método de carregamento do evento no formulário de cadastro

A Listagem 9 apresenta o código JavaScript utilizando a biblioteca jQuery para a estilização e visualização dos *ToolTip's* nos componentes das páginas. O

código abaixo é responsável por atribuir a classe CSS e pelo posicionamento do *ToolTip*.

```

43 jQuery(document).ready(function () {
44     $(function () {
45         $(document).tooltip({
46             position: {
47                 my: "center bottom+65px",
48                 at: "center top",
49             },
50             using: function (position, feedback) {
51                 $(this).css(position);
52                 $("

")
53                     .addClass("arrow")
54                     .addClass(feedback.vertical)
55                     .addClass(feedback.horizontal)
56                     .appendTo(this);
57             }
58         });
59     });
60 });


```

Listagem 9 - Código JavaScript utilizando a biblioteca jQuery para os *ToolTips*

A Listagem 10 apresenta duas funções JavaScript para formatação de máscaras. Entre as linhas 127 e 134 é feita a formatação da máscara de CEP. Para isso o componente *TextBox* ao receber o foco, a máscara é carregada. Entre as linhas 137 e 144 é criada a máscara para formatação do CNPJ.

```

126 //Mascara CEP
127 var sMaskCep = "99999-999"
128 function MascaraCep() {
129     //Todos os campos que tiverem [data-cep], carrega a mascara.
130     $('[data-cep='true']").each(function (i) {
131         var me = $(this)
132         me.mask(sMaskCep);
133     });
134 }
135
136 //Mascara CNPJ
137 var sMaskCNPJ = "99.999.999/9999-99"
138 function MascaraCNPJ() {
139     //Todos os campos que tiverem [data-cnpj], carrega a mascara.
140     $('[data-cnpj='true']").each(function (i) {
141         var me = $(this)
142         me.mask(sMaskCNPJ);
143     });
144 }

```

Listagem 10 - Código JavaScript para formatação de máscaras de CEP e CNPJ

A Listagem 11 apresenta o código JavaScript utilizando a biblioteca jQuery. A função tem como objetivo identificar se a primeira coluna do AspxGridView possui o componente *CheckBox* para adicionar o evento *Change*. Em seguida, verificar se o *CheckBox* está marcado. Caso o usuário marque o componente, a função abaixo atribui uma classe CSS ao registro selecionado, para diferenciá-lo dos demais.

```

62 | jQuery(document).ready(function () {
63 |     /* Localizar todos os checkbox dentro da primeira coluna de uma table */
64 |     jQuery('table tbody tr').each(function () {
65 |         /* localizar o checkbox dentro da primeira td da linha
66 |         corrente e atribuir o evento change */
67 |         jQuery(this).find('td:first input[type="checkbox"]').bind('change', function () {
68 |             /* verificar se o checkbox esta sendo checado */
69 |             if (jQuery(this).is(':checked')) {
70 |                 /* this = checkbox, localizar o primeiro elemento pai tr */
71 |                 jQuery(this).parents('tr').addClass('rowselected');
72 |             } else {
73 |                 jQuery(this).parents('tr').removeClass('rowselected');
74 |             }
75 |         });
76 |     });
77 |
78 |     /* Localizar todos os checkbox dentro de uma table com um css mGrid
79 |     (não legal usar class pra identificar) */
80 |     jQuery('table tbody tr:first th:first input[type="checkbox"]').bind('change', function () {
81 |         var checked = jQuery(this).prop('checked');
82 |         /* percorrer linsha por linha do grid */
83 |         jQuery(this).parents('table').find('tbody tr').each(function () {
84 |             /*
85 |             * this = tr agora..., para cada linha, localizar um checkbox dentro da primeira td
86 |             * após, invocar o evento change no checkbox que foi registrado no trecho acima.
87 |             */
88 |             jQuery(this).find('td:first input[type="checkbox"]').prop('checked',
89 |                 checked).trigger('change');
90 |         });
91 |     });
92 | });

```

Listagem 11 - Código JavaScript responsável pela verificação e atribuição de classe CSS ao registro do AspxGridView

Na Listagem 12 é apresentada a classe *.mGrid* do arquivo de estilo do projeto. Essa classe é responsável pela estilização do AspxGridView. Nele são definidas propriedades como tamanho, borda, cores, espaçamento e tamanho de fonte.

```

2497  .mGrid {
2498      width: 100%;
2499      background-color: #fff;
2500      margin: 5px 0 10px 0;
2501      border: solid 1px #525252;
2502      border-collapse: collapse;
2503  }
2504
2505  .mGrid td {
2506      padding: 2px;
2507      border: solid 1px #c1c1c1;
2508      color: #1d1d1d;
2509  }
2510
2511  .mGrid th {
2512      padding: 4px 2px;
2513      color: #fff;
2514      background: #424242 url(images/gridview/grd_head.png) repeat-x top;
2515      border-left: solid 1px #525252;
2516      font-size: 0.9em;
2517  }

```

Listagem 12 - Código CSS para estilização do AspxGridView

A Listagem 13 apresenta o código responsável pela edição dos contratos. Ao clicar no botão editar no *grid* de registros, o contrato é aberto em uma nova página. O código abaixo recupera a chave do registro clicado e ao abrir a nova página carrega as informações.

Ainda na Listagem 13, na linha 60 é recuperado o índice da linha na qual houve o evento do *click* do botão editar, para que, na linha 61, de acordo com o índice selecionado seja recuperado o “ContratoID”, chave da tabela de contratos. O “ContratoID” é passado por parâmetro no *request* da página, para que no evento *PageLoad* da página “ContratosCadastro.aspx” sejam carregadas as informações do contrato em questão. Em seguida é executado o código JavaScript responsável pela abertura da nova página.

```

57 Private Sub grdContratos_RowCommand(sender As Object, e As GridViewCommandEventArgs) _
58     Handles grdContratos.RowCommand
59
60     Dim index As Integer = Int32.Parse(DirectCast(e.CommandArgument, String))
61     Dim nContratoID As String = DirectCast(grdContratos.DataKeys(index).Values("ContratoID"), Int32)
62
63     Dim url As String, target As String, windowFeatures As String = ""
64
65     url = "ContratosCadaastro.aspx?id=" & nContratoID
66     target = "_blank"
67     If (String.IsNullOrEmpty(target) OrElse target.Equals("_self", StringComparison.OrdinalIgnoreCase)) _
68         AndAlso String.IsNullOrEmpty(windowFeatures) Then
69         Response.Redirect(url)
70     Else
71         Dim page As Page = DirectCast(HttpContext.Current.Handler, Page)
72         If page Is Nothing Then
73             Throw New InvalidOperationException("Cannot redirect to new window outside Page context.")
74         End If
75         url = page.ResolveClientUrl("ContratosCadaastro.aspx?id=" & nContratoID)
76         Dim script As String
77         If Not [String].IsNullOrEmpty(windowFeatures) Then
78             script = "window.open('{0}', '{1}', '{2}'); return false;"
79         Else
80             script = "window.open('{0}', '{1}'); return false;"
81         End If
82         script = [String].Format(script, url, target, windowFeatures)
83         ScriptManager.RegisterStartupScript(Me, GetType(Page), "Redirect", script, True)
84     End If
85 End Sub

```

Listagem 13 - Código responsável pela abertura da tela de adição de contratos

A Listagem 14 apresenta o código do componente `AspxGridView`. Para possibilitar a personalização de acordo com as necessidades do projeto, todos os componentes usados no projeto foram criados herdando do componente nativo do `Aspx`, ou seja, o exemplo abaixo mostra o `AspxGridView` utilizado no projeto que foi implementado de acordo com o `WebControls.GridView`. Com isso é possível a personalização total do componente, podendo assim ser criadas as propriedades e atributos necessários para a aplicação.

O código abaixo apresenta algumas propriedades declaradas para o componente de Grid, na linha 31 é declara a propriedade que recebe qual será o primeiro registro da página do *grid*. Na linha 37 está declarada a propriedade que recebe o último registro da página. A linha 43 é responsável pelo totalizador de registros do *grid*, ou seja, recebe o total de registros retornados pelo *DataSource*. Na linha 49 a propriedade *PageCount* recebe o total de registros dividido pelo número de registros por página, para então serem geradas o total de páginas.

```

18 <ToolboxData("<{0}:GridView runat=server></{0}:GridView"> _
19 Public Class AEGridView
20     Inherits System.Web.UI.WebControls.GridView
21     Implements IPagableItemContainer
22
23     Private nPageIndex As Integer
24     Private nMaxPages As Integer = 15
25
26     ''' <summary>
27     ''' TotalRowCountAvailable event key
28     ''' </summary>
29     Private Shared ReadOnly EventTotalRowCountAvailable As New Object()
30
31     Public ReadOnly Property FirstRecord() As Integer
32     Get
33         Return ((Me.PageIndex * Me.PageSize) + 1)
34     End Get
35 End Property
36
37     Public ReadOnly Property LastRecord() As Integer
38     Get
39         Return ((Me.PageIndex + 1) * Me.PageSize)
40     End Get
41 End Property
42
43     Public ReadOnly Property totalRows() As Integer
44     Get
45         Return (Me.Rows.Count)
46     End Get
47 End Property
48
49     Public Overrides ReadOnly Property PageCount() As Integer
50     Get
51         Return System.Convert.ToInt32(ViewState(Me.ID & "_PageCount"))
52     End Get
53 End Property

```

Listagem 14 - Código do componente AspxGridView

A Figura 27 apresenta o *DataSet* que foi criado para a geração de relatórios. Três tabelas foram usadas para compor o relatório de cadastro de Eventos.

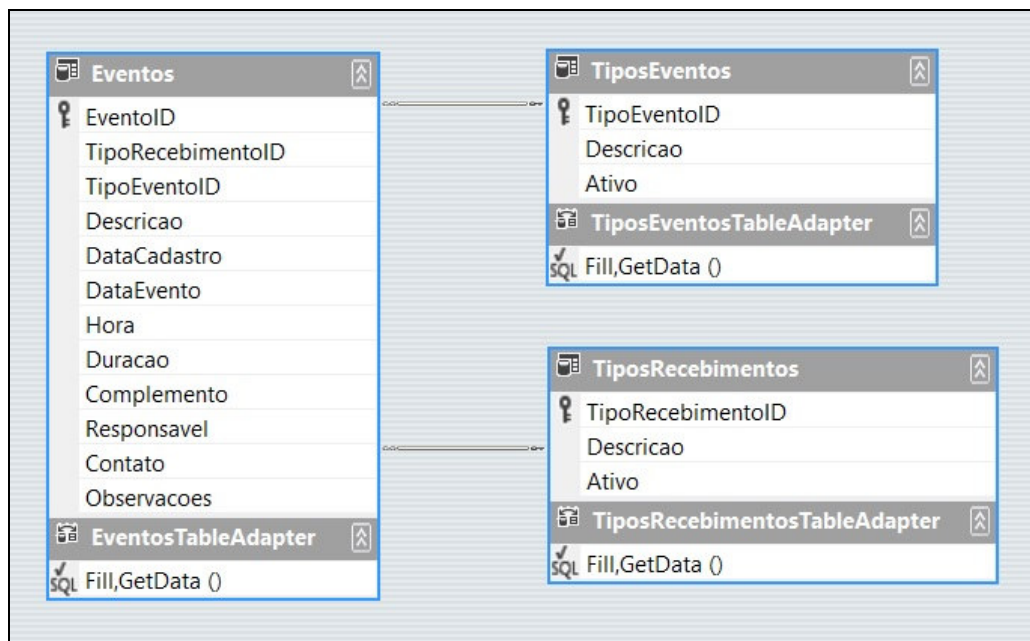


Figura 27 - Tabelas para geração do relatório de cadastro de eventos

Fonte: Autoria Própria

A Listagem 15 apresenta o código responsável pela recuperação das informações do banco de dados para passar para o *DataSource* do relatório. As linhas 15 e 19 recuperam o local onde está o “arquivo.rdlc” do relatório de eventos. Na linha 25 é declarado o objeto que recebe os registros recuperados do banco de dados, para que seja passado para uma variável do tipo *ReportDataSource* e em seguida essa variável é adicionada ao *DataSource* do relatório.


```
13 Private Sub ImprimeRelatorio()  
14     ReportViewer1.ProcessingMode = ProcessingMode.Local  
15     Dim rep As LocalReport = ReportViewer1.LocalReport  
16     'obtem o caminho do relatório  
17     Dim caminhoRpt As String = HttpContext.Current.Server.MapPath("~/")  
18     'define o local do relatório criado  
19     rep.ReportPath = caminhoRpt & "/AERelEventos/relEventos.rdlc"  
20  
21     Dim evento As New Evento  
22     Dim cx As New EventosEntities  
23     Dim dt As New DataTable  
24  
25     Dim eventos = (From e In cx.Eventos  
26                     Join te In cx.TiposEventos On te.TipoEventoID Equals e.TipoEventoID  
27                     Join tr In cx.TiposRecebimentos On tr.TipoRecebimentoID Equals e.TipoRecebimentoID  
28                     Select TipoEvento = te.Descricao, TipoRecebimento = tr.Descricao, e.Descricao,  
29                             e.DataCadastro, e.DataEvento, e.Hora, e.Duracao, e.Responsavel, e.Contato).ToList()  
30  
31     dt = Utilitarios.ToDataTable(eventos)  
32  
33     'Cria uma fonte de dados no relatório para o dataset vendas  
34     Dim dsVendasRpt As New ReportDataSource()  
35     'define o nome do datasource  
36     dsVendasRpt.Name = "EventosDataSet"  
37     'usa primeira tabela do dataset  
38     dsVendasRpt.Value = dt  
39     'atribui o datasource ao relatório  
40     rep.DataSources.Add(dsVendasRpt)  
41 End Sub
```

Listagem 15 - Código responsável pelo carregamento das informações para a impressão do relatório de cadastro de eventos

5. CONCLUSÃO

O objetivo principal do trabalho de conclusão de curso foi evidenciar que com o uso das tecnologias Microsoft® Visual Basic.NET, .NET Framework, Java Script, HTML5, CSS3 e demais, apresentadas no Capítulo 3 deste trabalho foi possível implementar todas as funcionalidades pretendidas para o sistema *web* para gerenciamento de EPE, denominado RIA pelos conceitos e recursos utilizados em sua implementação. Esses recursos definem uma interface caracterizada como rica pelas funcionalidades de interação do usuário com o aplicativo que oferecem.

Os recursos que a linguagem Visual Basic .NET, como seu conceito de orientação a objetos, trouxeram mais agilidade para a linguagem. A linguagem é semelhante ao Visual Basic 6.0, que foi utilizada para desenvolver uma primeira versão do sistema, assim, a migração e o uso do Visual Basic.NET se tornou fácil. O ambiente de desenvolvimento apresenta recursos que facilitam e agilizam o trabalho do desenvolvedor.

Como perspectiva futura pretende-se complementar a implementação das funcionalidades do sistema de forma a adaptá-lo para o mercado com o intuito de comercializá-lo. A funcionalidade essencial do sistema foi implementada por completo, atendendo, assim, aos objetivos proposto neste trabalho.

Pode-se perceber que o uso de RIA está diretamente ligado à *web* 2.0 e considerando que a *web* está cada vez mais dinâmica e interativa, as RIAs tendem a ser a tecnologia que continuará acompanhando essa mudança na *web*. A velocidade com que essas tecnologias se expandirão é imprevisível, no entanto as RIAs tendem a durar até que novas características, ideias e conceitos transformem em algo novo que requeira um novo conjunto de tecnologias para atender tal necessidade, provavelmente uma evolução das RIAs.

REFERÊNCIAS

ALLAIRE, Jeremy. **Macromedia Flash MX: a next-generation rich client**. 2002. Disponível em: <<http://www.adobe.com/devnet/flash/whitepapers/richclient.pdf>>. Acesso em: 17 mar. 2013.

BOZZON, Alessandro, COMAI, Sara. **Conceptual modeling and code generation for rich internet applications**. 6th International Conference on Web Engineering (ICWE '06), 2006, p. 353-360.

DUHL, Joshua. **Rich internet applications**. 2003. White Paper. In: IDC Opinion. Disponível em: <http://www.adobe.com/platform/whitepapers/idc_impact_of_rias.pdf>. Acesso em: 19 maio 2013.

FLANAGAN, David. **JavaScript o guia definitivo**. Porto Alegre: Editora ARTMED, 2002.

GOMES, Ana Laura. **XHTML/CSS criação de páginas web**. São Paulo: Editora Senac São Paulo, 2010.

GUANAIS, Kaio Araújo. **Aplicações ricas de internet**. 2010. Disponível em: <<http://www3.iesam-pa.edu.br/ojs/index.php/sistemas/article/viewFile/544/412>>. Acesso em: 20 ago. 2013.

LEITE, M. **Técnicas de programação: Uma Abordagem Moderna**. Rio de Janeiro: BRASPORT, 2008.

LÓPEZ, Xavier Farré. **Rich internet application**. 2005. Trabalho de conclusão de curso. Universidade Politécnica da Catalunia, Espanha. Disponível: <http://biblioteca.upc.es/PFC/arxiu/migrats/40624-4.pdf>. Acesso em: 17 jun. 2013.

MEIRELLES, G. F. **Tudo sobre eventos**. São Paulo: STS, 1999.

ORACLE. **Oracle**. 2007. Disponível em: <http://www.oracle.com/technology.com/oramag/oracle/04-jul/o44de_trends.html> Acesso em: jun. 2013.

SILVA, Mauricio S. **Site sobre CSS e padrões web**, 2012. Disponível em: <<http://www.maujor.com/index.php>>. Acesso em: 8 set. 2013.

THOMAS, J.R.; NELSON, J.K.; SILVERMAN, S.J. **Métodos de pesquisa em atividade**. Porto Alegre: Artmed, 2007.

VISUAL PARADIGM. **Visual Paradigm** Disponível em: <<http://www.visual-paradigm.com/>>. Acesso em: 25 out. 2013.