

LIGA DE ENSINO DO RIO GRANDE DO NORTE
CENTRO UNIVERSITÁRIO DO RIO GRANDE DO NORTE
CURSO DE PÓS-GRADUAÇÃO EM DESENVOLVIMENTO DE SISTEMAS
CORPORATIVOS

INÁCIO DE LOIOLA SOUZA SILVA

DESENVOLVIMENTO DE UM SISTEMA DE REDE SOCIAL UTILIZANDO
A PLATAFORMA JAVA (ENTERPRISE EDITION)

DESENVOLVIMENTO DE UM SISTEMA DE REDE SOCIAL UTILIZANDO
A PLATAFORMA JAVA (ENTERPRISE EDITION)

desenvolvimento de sistemas corporativos
do centro universitário do Rio Grande do
Norte - UNI-RN, como parte dos
requisitos para obtenção do diploma de
especialista

Orientador: Prof. Msc. Ricardo Wendel
Rodrigues da Silva

INÁCIO DE LOIOLA SOUZA SILVA

**DESENVOLVIMENTO DE UM SISTEMA DE REDE SOCIAL UTILIZANDO
A PLATAFORMA JAVA (ENTERPRISE EDITION)**

Relatório de Projeto apresentado ao Curso de pós-graduação em desenvolvimento de sistemas corporativos do centro universitário do Rio Grande do Norte – UNI-RN, como parte dos requisitos para obtenção do diploma de especialista.

Orientador: Prof. Msc. Ricardo Wendell Rodrigues da Silveira

NATAL/RN
2012

UNI-RN

UNIVERSIDADE DO RIO GRANDE DO NORTE

UNI-RN
BIBLIOTECA

RB: 36072
Cód de Barra: MP00031
Data: 29 / 01 / 13

004.43
55862
2012
ex. 01

AGRADECIMENTOS

A Deus, que sem ele nada seríamos.

A família, que graças ao apoio incondicional e à paciência que tiveram em todos estes momentos.

Aos amigos e colegas de curso que estiveram juntos por toda essa jornada.

Aos professores, que nos ajudaram e incentivaram nesta etapa de nossas vidas, em particular ao professor Msc. Ricardo Wendell Rodrigues da Silveira, que me orientou no desenvolvimento deste projeto.

Dedico este trabalho a todos que de forma direta ou indireta me ajudaram no seu desenvolvimento.

De forma especial dedico a minha família, que me apoia em todos os momentos e novos desafios que a vida nos leva.

E com grande saudade, dedico ao meu avô Raimundo Bezerra de Souza, que foi um homem guerreiro, corajoso e trabalhador e sempre me incentivou a correr atrás dos meus objetivos.

AGRADECIMENTOS

A Deus, que sem ele nada seríamos.

À família, que graças ao apoio incondicional e a paciência que tiveram em todos estes momentos.

Aos amigos e colegas de curso que estiveram juntos por toda essa jornada.

Aos professores, que nos ajudaram e incentivaram nesta etapa de nossas vidas, em particular ao professor Msc. Ricardo Wendell Rodrigues da Silveira, que me orientou no desenvolvimento deste projeto.

"O homem nunca pode parar de sonhar.
O sonho é o alimento da alma, como a comida
é o alimento do corpo."

(Paulo Coelho)

RESUMO

Atualmente existe um grande número de pessoas conectadas diariamente à Internet compartilhando informações e conhecimentos de maneira frequente, principalmente através de sistemas de redes sociais que tratam com dados pessoais, muitas vezes com o intuito de divulgar informações e promover eventos. Este trabalho apresenta um sistema de eventos online, baseado em um sistema que suporta redes sociais existentes em eventos, organizado por áreas de interesse e com o objetivo de contribuir para a divulgação de eventos e promover associações, dessa forma facilitando o planejamento do evento e relacionando as pessoas antes e após a realização do mesmo. O sistema web foi implementado utilizando tecnologias de plataforma Java(Enterprise Edition) como Java Server Faces (JSF), Hibernate e Spring.

Palavras-Chave: Sistema, Redes Sociais, Eventos, Internet.

"O homem nunca pode parar de sonhar.
O sonho é o alimento da alma, como a comida
é o alimento do corpo."
(Paulo Coelho)

RESUMO

Atualmente existe um grande número de pessoas conectadas diariamente à Internet, compartilhando informações e comunicando-se de maneira frequente, principalmente através de sistemas de redes sociais que tratam de criar laços virtuais entre os perfis das pessoas na rede. Motivado por esse novo cenário de relações interpessoais, este trabalho apresenta uma proposta e protótipo de um sistema que suporta redes sociais interessadas em eventos, organizados por áreas de interesse e com o objetivo de contribuir para a divulgação de empresas promotoras associadas, dessa forma facilitando o planejamento do evento e relacionando as pessoas antes e após a realização do mesmo. O sistema web foi implementado utilizando tecnologias da plataforma Java(Enterprise Edition) como: Java Server Faces (JSF), Hibernate e Spring.

Palavras-Chave: Sistema, Redes Sociais, Eventos, Internet.

LISTA ABSTRACT

Figura 1 - Exemplo de um Diagrama de Caso de Uso..... 13

Currently there are a large number of people connected to the Internet every day, sharing information and communicating so often, mainly through systems that deal with social networks to create virtual links between the profiles of people on the network. Motivated by this new scenario of interpersonal relations, this paper presents a proposal and a prototype system that supports social networks interested in events organized by areas of interest and in order to contribute to promoting the dissemination of associated companies, thus facilitating event planning and relating to people before and after completion. The system was implemented using web technologies of the Java platform (Enterprise Edition) as: Java Server Faces (JSF), Hibernate and Spring.

Figura 2 - Diagrama de Sequência da classe Evento..... 22

Keywords: System, Social Networks, Events, Internet.

Figura 10 - Tela para Cadastro de Evento..... 24

Figura 11 - Tela para atribuir permissão de administrador..... 26

Figura 12 - Tela para adicionar um grupo de interesse..... 27

Figura 13 - Contorno de caso de uso para ator com perfil de usuário padrão..... 28

Figura 14 - Tela de atualização de perfil..... 29

Figura 15 - Tela de visualização de eventos..... 30

Figura 16 - Exemplo de como o projeto foi implementado..... 32

Figura 17 - Pasta WEB-INF do projeto..... 33

LISTA DE ILUSTRAÇÕES

Figura 1 – Exemplo de um diagrama de Casos de Uso	15
Figura 2 – Exemplo de um Diagrama de Classes.....	15
Figura 3 – Exemplo de um Diagrama de Sequência	16
Figura 4 - Exemplo da Arquitetura do Hibernate	18
Figura 5 - Exemplo de estados de um objeto.....	19
Figura 6 – Ciclo de vida de uma requisição JSF	21
Figura 7 - Diagrama de Classes.....	22
Figura 8 - Diagrama de Sequência da classe Evento.....	23
Figura 9 - Casos de Uso – Cenário para ator com perfil administrativo	24
Figura 10 - Tela para Cadastro de Evento	24
Figura 11- Tela para atribuir permissão de administrador	26
Figura 12 - Tela para cadastrar um grupo de interesse	27
Figura 13 - Cenário de casos de uso para ator com perfil de usuário padrão	28
Figura 14 - Tela de atualização de perfil	29
Figura 15 - Tela de visualização de eventos	30
Figura 16 - Exemplo de como o projeto foi implementado.....	32
Figura 17 - Pasta WEB-INF do projeto	33
5.1.1 Diagrama de Classes.....	22
5.1.2 Diagrama de Sequência.....	23
5.1.3 Diagramas de Casos de Uso	24
5.1.3.1 Cenário de casos de uso para o ator com perfil de administrador	24
5.1.3.2 Tela de layout de cadastro evento.....	24
5.1.3.3 Descrição do caso de uso cadastrar evento.....	25
5.1.3.4 Tela de layout de caso de uso atribuir permissão administrativa	26
5.1.3.5 Descrição do caso de uso para atribuir permissão administrativa	26
5.1.3.6 Tela de layout do caso de uso cadastrar grupo interesse	27
5.1.3.7 Descrição do caso de uso cadastrar grupo de interesse.....	27
5.1.3.8 Cenário de casos de uso para o ator com perfil de usuário padrão	28
5.1.3.9 Tela de visualização e atualização de perfil.....	29
5.1.3.10 Descrição de caso de uso atualizar perfil.....	29
5.1.3.11 Tela de visualização de eventos	30
5.1.3.12 Descrição do caso de uso adicionar interesse	30

SUMÁRIO

1 INTRODUÇÃO	10
2 REDES SOCIAIS	12
2.1 A INTERNET COMO AGENTE DE MARKETING	13
3 FERRAMENTAS UTILIZADAS PARA A ANÁLISE	14
3.1 UML	14
3.1.1 Diagrama de Casos de Uso.....	14
3.1.2 Diagrama de Classes.....	15
3.1.3 Diagrama de Sequência.....	15
4 FERRAMENTAS PARA DESENVOLVIMENTO	17
4.1 JAVA	17
4.2 POSTGRE SQL.....	17
4.3 HIBERNATE.....	18
4.4 SPRING	20
4.5 JAVA SERVER FACES	20
4.6 TOMCAT.....	21
5 METODOLOGIA.....	22
5.1 ANÁLISE DE REQUISITOS.....	22
5.1.1 Diagrama de Classes.....	22
5.1.2 Diagrama de Sequência.....	23
5.1.3 Diagramas de Casos de Uso	24
5.1.3.1 Cenário de casos de uso para o ator com perfil de administrador	24
5.1.3.2 Tela de layout de cadastrar evento.....	24
5.1.3.3 Descrição do caso de uso cadastrar evento.....	25
5.1.3.4 Tela de layout do caso de uso atribuir permissão administrativa	26
5.1.3.5 Descrição do caso de uso para atribuir permissão administrativa	26
5.1.3.6 Tela de layout do caso de uso cadastrar grupo interesse	27
5.1.3.7 Descrição do caso de uso cadastrar grupo de interesse	27
5.1.3.8 Cenário de casos de usos para o ator com perfil de usuário padrão	28
5.1.3.9 Tela de visualização e atualização de perfil.....	29
5.1.3.10 Descrição de caso de uso atualizar perfil.....	29
5.1.3.11 Tela de visualização de eventos	30
5.1.3.12 Descrição do caso de uso adicionar interesse.....	30

5.1.3.13 Descrição do caso de uso postar mensagem.....	31
5.2 IMPLEMENTAÇÃO.....	32
6 CONCLUSÃO.....	35
REFERÊNCIAS	36
ANEXOS	

O apogeu das redes sociais se deu a partir dos primeiros anos do século XXI, mas atualmente está em maior diversidade devido ao acesso a internet terem aumentado, também com o avanço tecnológico e a evolução da tecnologia 3G, os quais possibilitam aos usuários dos centros urbanos terem 24 horas conectados.

Os benefícios das redes sociais são muitos, pois existem redes direcionadas para nichos distintos de cidadãos, por exemplo, existe uma rede bem conhecida que tem como objetivo principal criar laços entre empregados e empregadores, mantendo um espaço de diálogo dos empregados e registro das melhores empresas, isso facilita tanto para os funcionários que aumentam seu networking no mercado de trabalho quanto para as empresas que são poder selecionar mais rapidamente os possíveis perfis para novos colaboradores.

Este trabalho tem como objetivo principal propor o desenvolvimento de um protótipo de rede social para diversos grupos de interesse, como por exemplo: cinema, teatro, shows e etc. As principais funcionalidades para os usuários serão: visualizar as informações sobre eventos, comentar evento, participar de eventos, fazer comentários, enviar fotos e vídeos, adicionar amigos, entre outras atividades.

O desenvolvimento deste projeto será realizado utilizando tecnologias de plataforma Java(Enterprise Edition), que são O Hibernate que trata do mapeamento objeto-relacional entre as classes Java e o SGBD (sistema gerenciador de banco de dados), no nosso caso o PostgreSQL, além de realizar todas as operações referentes a persistência, exclusão e consulta nos dados. O Spring Framework, o qual gerencia a instância de classes e injeção de dependências, resultando em um bom acoplamento entre classes, também utilizaremos o Spring Security que é um projeto do Spring Framework que visa garantir a segurança do sistema. Por último a parte de visualização usaremos o JSP (Java Server Pages), que é um framework voltado para a camada de visualização, ou seja, criação das páginas web que irão interagir com o usuário, é

1 INTRODUÇÃO

As soluções em tecnologias da informação estão cada vez mais presentes no dia a dia das pessoas, seja para ler e-mails, assistir vídeos, ouvir músicas, agendar tarefas e também encontrar amigos através das redes sociais, que são um fenômeno no mundo inteiro.

O apogeu das redes sociais se deu a partir dos primeiros anos do século XXI, mas atualmente está com maior divergência, devido às facilidades de acesso a internet terem aumentado, também com o menor custo dos smartphones e a implantação da tecnologia 3G, os quais possibilitam aos usuários dos centros urbanos ficarem 24 horas conectados.

Os benefícios das redes sociais são muitos, pois existem redes direcionadas para nichos distintos de cidadãos, por exemplo, existe uma rede bem conhecida que tem como objetivo principal criar laços entre empregados e empregadores, mantendo uma espécie de currículo dos empregados e registro das melhores empresas, isso facilita tanto para os funcionários que irão aumentar seu networking no mercado de trabalho quanto para as empresas que irão poder selecionar mais rapidamente os possíveis perfis para novos colaboradores.

Este trabalho tem como objetivo principal propor o desenvolvimento de um protótipo de rede voltada para eventos diversos, organizados por grupos de interesse, como por exemplo: cinema, teatro, shows e etc. As principais funcionalidades para os usuários serão: visualizar as informações sobre eventos, comentar evento, participar de sorteios, fazer comentários, enviar fotos e vídeos, convidar amigos, entre outras atividades;

O desenvolvimento deste projeto será realizado utilizando tecnologias da plataforma Java(Enterprise Edition), que são: O Hibernate, que trata do mapeamento objeto/relacional entre as classes Java e o SGBD (sistema gerenciador de banco de dados), no nosso caso o PostgresSql, além de realizar todas as operações referentes a persistência, exclusão e consulta aos dados. O Spring Framework, o qual gerencia a instanciação de classes e injeção de dependências, resultando em um baixo acoplamento entre classes, também utilizaremos o Spring Security que é um projeto do Spring Framework que visa garantir a segurança do sistema. Por último o Java Server Faces (JSF), que é um framework voltado para a camada de visualização, ou seja, criação das páginas web que irão interagir com o usuário, é

baseado no padrão MVC que tem como princípio a divisão da aplicação em três camadas: modelo, controle e visão.

No primeiro capítulo expõe-se o Referencial Teórico, que inclui os subcapítulos relativos ao negócio das redes sociais, as ferramentas utilizadas para análise e as ferramentas utilizadas para o desenvolvimento. O segundo aborda a metodologia utilizada para a elaboração do projeto. No terceiro é exposto os resultados obtidos após a implementação. Por último temos as conclusões onde se relatam as aprendizagens adquiridas, as dificuldades obtidas e as melhorias possíveis.

A presença dos elementos tecnológicos na sociedade vem transformando o modo dos indivíduos se comunicarem, se relacionarem e construírem comunidades. Desde hoje praticamos novas tecnologias (ALVES 2009 p.15).

Com as novas tecnologias os indivíduos têm uma pluralidade gigantesca de ferramentas para que possam adquirir conhecimento e compartilhar com outros. Essa tecnologia faz parte da vida de grande parte da população mundial, criando um mundo "sem fronteiras" onde as pessoas não precisam sair de casa para se relacionarem com outras pessoas. Isso acontece porque as ferramentas tecnológicas estão se tornando cada vez mais acessíveis e os indivíduos estão se tornando cada vez mais dependentes delas. Isso acontece porque as ferramentas tecnológicas estão se tornando cada vez mais acessíveis e os indivíduos estão se tornando cada vez mais dependentes delas.

A facilidade de acesso à internet e a facilidade de uso das ferramentas tecnológicas são fatores que contribuem para a popularização das redes sociais. Isso acontece porque as ferramentas tecnológicas estão se tornando cada vez mais acessíveis e os indivíduos estão se tornando cada vez mais dependentes delas. Isso acontece porque as ferramentas tecnológicas estão se tornando cada vez mais acessíveis e os indivíduos estão se tornando cada vez mais dependentes delas.

2 REDES SOCIAIS

2.1 A INTERNET COMO AGENTE DE MARKETING

Os seres humanos desde os primórdios se relacionam com outros formando grupos sociais que sempre estão em volta de alguma semelhança entre eles, por exemplo temos os grupos religiosos que frequentemente se encontram para fortalecer os laços espirituais, esses conglomerados de pessoas vão formando redes que se entrelaçam e tornam cada vez mais a vida em sociedade algo compartilhado e não singular. Com a diversificação dos meios digitais e a internet, as pessoas começaram a ter outro leque de possibilidades para se comunicarem e formar laços sociais.

A presença dos elementos tecnológicos na sociedade vem transformando o modo dos indivíduos se comunicarem, se relacionarem e construir conhecimentos. Somos hoje praticamente vividos pelas novas tecnologias! (ALVES, 2002, p.15).

Com as novas tecnologias os indivíduos têm uma pluralidade gigantesca de ferramentas para que possam adquirir conhecimento e compartilhá-lo com outros. Essa tendência tecnológica já faz parte da vida de grande parte da população mundial, criando um chamado "mundo virtual" que está inserido de forma direta na vida de cada um. Alguns partem do pressuposto que o espaço virtual não faz parte da realidade e se trata apenas de um ambiente paralelo que as pessoas utilizam como forma de lazer, mas já se pode comprovar bem claramente que essa "vida virtual" está diretamente ligada a realidade, um bom exemplo disso são empresas que estão pesquisando os perfis virtuais dos candidatos para tomar conclusões sobre alguns fatores ligados aquela pessoa.

A facilidade de acesso e a inclusão digital esta levando todos os diferentes níveis sociais a possibilidade de utilizar as redes sociais e os usuários não são somente os jovens, mas também adultos e idosos, sendo que em sua maioria geralmente são adolescentes até porque eles são totalmente da geração digital. Com essa diversidade de pessoas utilizando as redes, estão começando a surgir redes direcionadas para setores ou atividades específicas, como por exemplo o LinkedIn que é uma ferramenta voltada para os profissionais criarem sua identidade facilitando o acesso das empresas interessadas no seu perfil e melhorando seu networking com outros da mesma área.

2. FERRAMENTAS UTILIZADAS PARA A ANÁLISE

2.1 A INTERNET COMO AGENTE DE MARKETING

Atualmente a necessidade de conexão é algo predominante em boa parte da população mundial, onde as pessoas precisam estar sempre conectadas para realizar varias atividades. Essa grande demanda de internautas gerou um campo forte para as empresas de marketing, daí surgiu a publicidade na web que hoje movimentam bilhões de dólares por ano. Quando o usuário navega em um site, vê sua rede social ou até pesquisa sobre algum assunto fica com os olhos presos a tela do computador e geralmente é aproveitando essa atenção momentânea que os agentes publicitários criam banners e páginas atrativas ao ponto de fazer o internauta ao menos ler o anúncio e as vezes clicar e ser redirecionado para outra página com mais detalhes sobre a propaganda.

As redes sociais são febre no mundo inteiro e dão uma boa margem para o marketing, pois facilitam o acesso a grupos de pessoas restritos que são filtrados previamente pelos seus perfis, por exemplo, com isso é possível enviar anúncios de utilidades domésticas para donas de casa ou produtos dos times para os torcedores, criando uma boa chance de consumir a venda, pois aquelas pessoas tem mais atração de comprarem produtos pelos quais se interessam.

2.1.1 Diagrama de Casos de Uso

Através de uma linguagem simples auxilia na compreensão das funções e serviços que serão disponíveis no sistema, neste estágio os desenvolvedores não se preocupam na implementação das funções, tratando assim de visão do usuário no sistema. O diagrama de caso de uso é de grande ajuda na análise de requisitos e é frequentemente utilizado no início do desenvolvimento e pode sofrer alterações ao decorrer do projeto.

3 FERRAMENTAS UTILIZADAS PARA A ANÁLISE

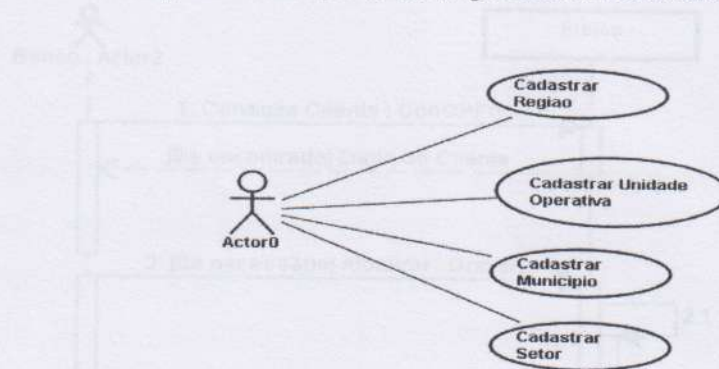
3.1 UML

Para podermos entender a Unified Modeling Language (UML) devemos saber um pouco de sua historia. A UML foi criada com a união de três metodologias de modelagem que são: Object Modeling Technique (OMT), BOOCH e Object-oriented software engineering (OOSE). Estas metodologias eram muito utilizadas pelos profissionais na área de desenvolvimento de software na década de noventa, as três utilizavam modelagem orientada a objetos. Esta união só pôde ser realizada graças ao apoio da "Rational Software" que financiou o projeto que culminou com o lançamento de uma metodologia robusta ao mesmo tempo ágil e eficaz. A UML é uma linguagem visual de modelagem de dados que é orientada objetos e auxilia no desenvolvimento de sistemas computacionais que está sendo muito utilizada hoje em dia. A UML possui vários diagramas que auxiliam, simplificando na hora do desenvolvimento, melhorando a compreensão do sistema e facilitando o entendimento para os desenvolvedores e clientes. Os quais são descritos conforme Thorwald (2006, p.20) afirma "A UML tornou se nos últimos anos a linguagem padrão de modelagem adotada internacionalmente pela indústria de engenharia de software".

3.1.1 Diagrama de Casos de Uso

Através de uma linguagem simples auxilia na compreensão das funções e serviços que serão disponíveis no sistema, neste estagio os desenvolvedores não se preocupam na implementação das funções, tratando assim da visão do usuário no sistema. O diagrama de caso de uso é de grande ajuda na etapa de análise de requisitos é freqüentemente utilizado no inicio do desenvolvimento e pode sofrer alterações no decorrer do projeto.

Figura 1 – Exemplo de um diagrama de Casos de Uso



Fonte: O autor (2012)

3.1.2 Diagrama de Classes

É o diagrama mais utilizado da UML ele prioriza a visualização das classes assim como seus atributos e métodos, também mostra como as classes se relacionam, complementam e troca de informações entre si. O diagrama de classes é tão importante que dele podemos fazer a maioria dos outros diagramas da UML.

Figura 2 – Exemplo de um Diagrama de Classes

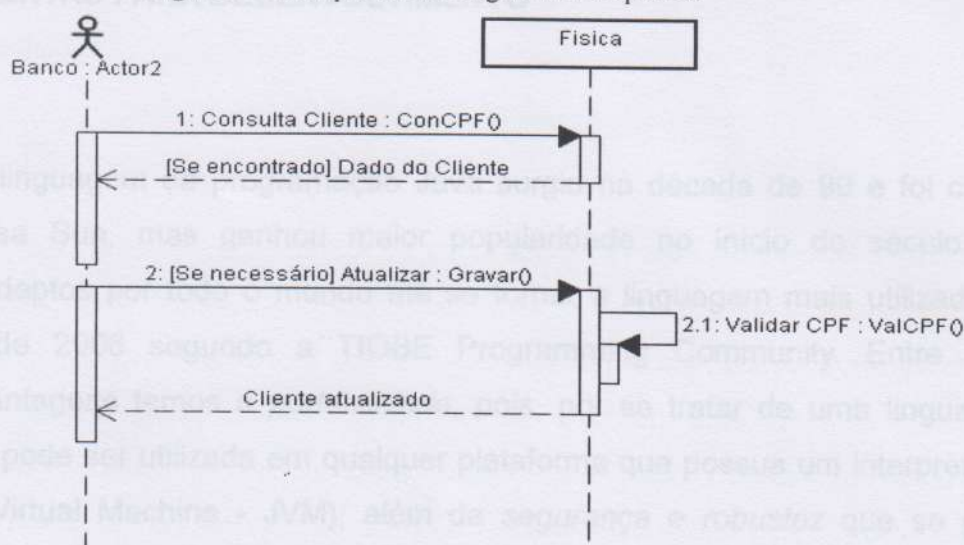


Fonte UML – Uma Abordagem Prática 2006

3.1.3 Diagrama de Sequência

Este diagrama se preocupa com a ordem cronológica que os eventos são acionados. O diagrama de sequência na maioria das vezes se identifica com alguns casos de usos que podem ser acionados pelo usuário. Este diagrama também depende do diagrama de classes, pois pode conter métodos que serão disparados entre os objetos.

Figura 3 – Exemplo de um Diagrama de Seqüência



Fonte UML – Uma Abordagem Prática 2006

4 FERRAMENTAS PARA DESENVOLVIMENTO

4.1 JAVA

A linguagem de programação Java surgiu na década de 90 e foi criada pela empresa Sun, mas ganhou maior popularidade no início do século XXI ganhando adeptos por todo o mundo até se tornar a linguagem mais utilizada no mundo desde 2006 segundo a TIOBE Programming Community. Entre suas principais vantagens temos a *portabilidade*, pois, por se tratar de uma linguagem interpretada pode ser utilizada em qualquer plataforma que possua um interpretador java (Java Virtual Machine - JVM); além da *segurança e robustez* que se pode alcançar com utilização de sistemas em Java que devido a sua execução se dá na JVM protege o sistema contra ataques e outras ameaças; por se tratar de um *linguagem orientada a objetos* permite a herança e a reutilização de códigos de forma dinâmica e estática facilitando o trabalho dos programadores; o *alto desempenho* do Java permite a utilização de vários recursos como: multithreading, compilação just-in-time e utilização de código nativo.

4.2 POSTGRE SQL

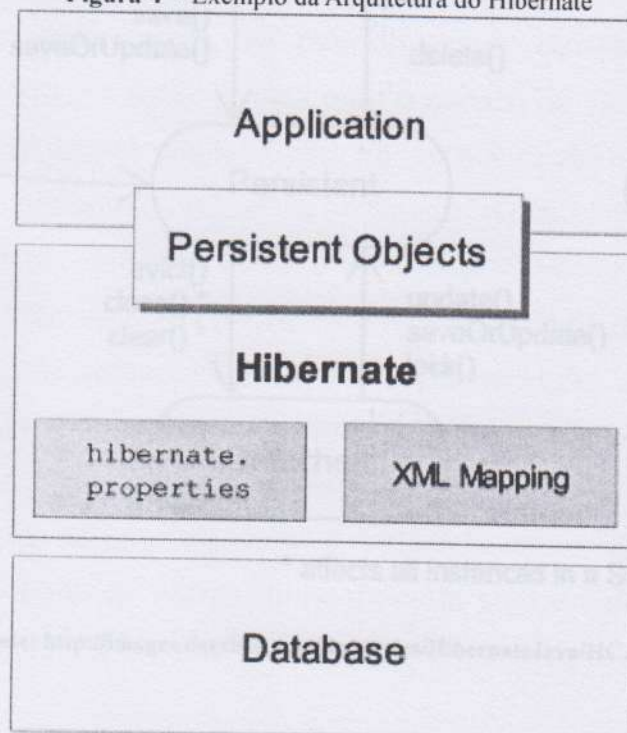
O PostgreSQL é um sistema gerenciador de banco de dados objeto relacional, que tem como base um outro SGBDOR o "Postgres 4.2" que foi desenvolvido por uma equipe que tinha como orientador o professor Michael Stonebraker do departamento de ciências da computação em berkely na Califórnia em 1986 e também contava com o apoio de vários colaboradores como Army Research Office (ARO), National Science Foundation (NSF) entre outros. A sua primeira versão foi lançada em 1989, posteriormente passou a ser desenvolvido versões que logo em seguida passariam a ser lançada anualmente em que seria corrigindo os erros e bugs, já em 1991 o código do Postgres foi obtido pela "Illustra Information Technologies". Em 1994 foi atribuído um interpretador da linguagem do SQL e esta versão ficou conhecida como Postgres95. O nome PostgreSQL só começou a ser utilizado em 1996 e a primeira versão do PostgreSQL 6.0 foi lançada em 1997, depois de três anos foi lançada a versão 7.0 que continha grandes inovações como um melhor suporte a SSL, outer join entre outros desde então

passou a se popularizar e hoje em dia esta na sua versão 8.3.6 e é um dos SGBD de grande destaque.

4.3 HIBERNATE

O hibernate é um framework de mapeamento objeto/relacional utilizado na linguagem Java, este framework facilita bastante o trabalho dos desenvolvedores nas questões que dizem respeito a camada de persistência e recuperação de dados no SGBD (Sistema Gerenciador de Banco de Dados), pois, com o hibernate podemos desenvolver classes persistentes utilizando Java convencional, implementar mecanismos de mapeamento e consultas como o uso de annotations para mapear as classes e das ferramentas HQL(Hibernate Query Language) e Criteria Queries para realizar consultas no banco de dados. Atualmente o hibernate é bastante utilizado devido as grandes vantagens que trás para o projeto, especialmente no fator de independência entre camadas e na diminuição do tempo de implementação, pois sem o uso do hibernate os programadores desprendiam bastante tempo para criar estratégias que funcionassem para o mapeamento objeto/relacional. Abaixo pode-se ver uma figura que trás uma descrição de alto nível da arquitetura do Hibernate.

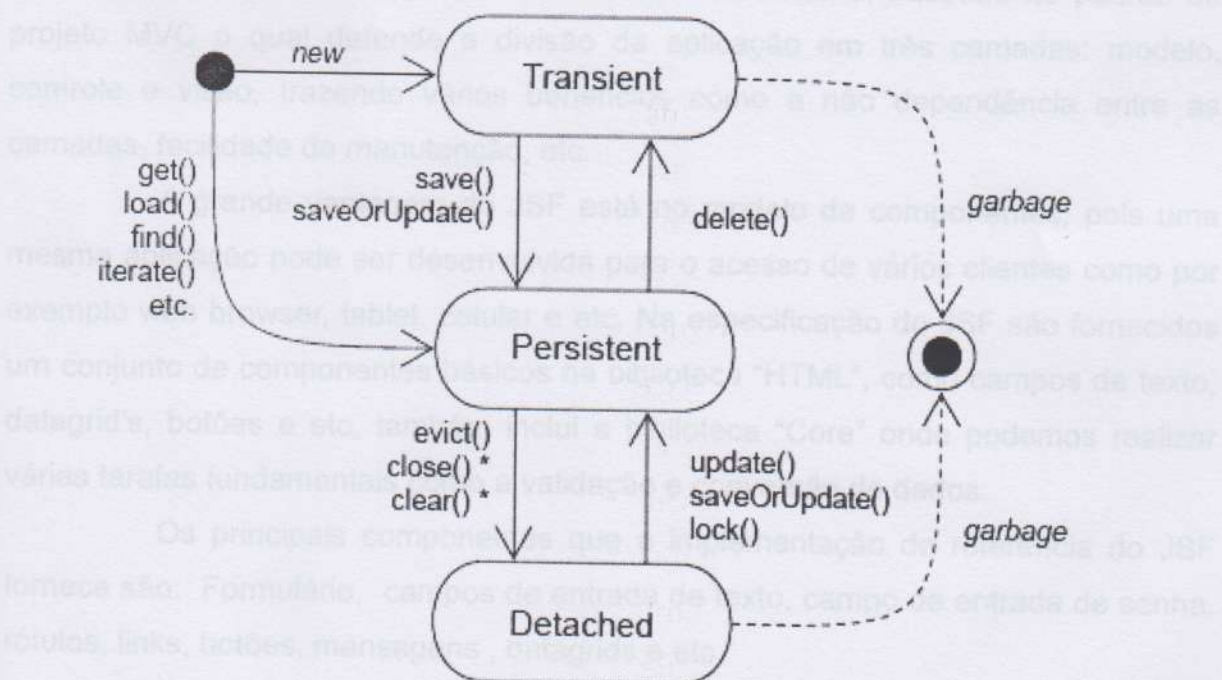
Figura 4 - Exemplo da Arquitetura do Hibernate



Fonte: <http://docs.jboss.org/hibernate/orm/3.5/reference/pt-BR/html/images/overview.png>

Para o hibernate existem três tipos de estado em que um objeto pode estar, que são: persistente, transiente e detached. Um objeto no estado persistente significa que sua instância está associada a um contexto no banco de dados, podendo este ser persistindo, consultado, atualizado ou removido do SGBD (Sistema Gerenciador de Banco de Dados), já o estado transiente trata-se de objetos que suas instâncias não estão vinculadas ao banco de dados, então eles são instanciados, utilizados e após serem destruídos não podem ser reconstruídos automaticamente, por fim temos o estado detached que são objetos que já foram persistentes mas que devido a um fechamento de sessão ou outro motivo foram desvinculados, trata-se de um estado intermediário entre persistente e transiente. A figura abaixo demonstra o fluxo de estados de um objeto desde sua instanciação até o momento em que o objeto não estava mais referenciado ou foi deletado e posteriormente foi destruído pelo Garbage Collector (Coletor de Lixo) liberando a memória para otimizar o processamento.

Figura 5 - Exemplo de estados de um objeto



* affects all instances in a Session

Fonte: <http://images.devshed.com/da/stories/HibernateJava/HC.JPG>

4.4 SPRING

O Spring Framework, é uma ferramenta da plataforma Java(Enterprise Edition), baseado nos padrões de Inversão de Controle(IoC), que tem como objetivo principal a injeção de dependências em classes Java, ou seja, instanciação de classes injetadas com base em definições de um arquivo xml ou anotações(annotations) nas próprias classes, o que resulta em um baixo acoplamento entre as mesmas.

O Spring fornece também outros projetos que trazem recursos adicionais as suas definições básicas, como por exemplo: o Spring WebService, o Spring WebFlow e o Spring Security, este último trata da segurança e controle de acesso em aplicações, abrange desde a segurança de diretórios até o acesso a dados de usuários e controle de permissões, sendo bastante utilizado e confiável.

4.5 JAVA SERVER FACES

O Java Server Faces (JSF) trata-se de um framework criado para facilitar o desenvolvimento da interface com o usuário do sistema, baseado no padrão de projeto MVC o qual defende a divisão da aplicação em três camadas: modelo, controle e visão, trazendo vários benefícios como a não dependência entre as camadas, facilidade de manutenção, etc.

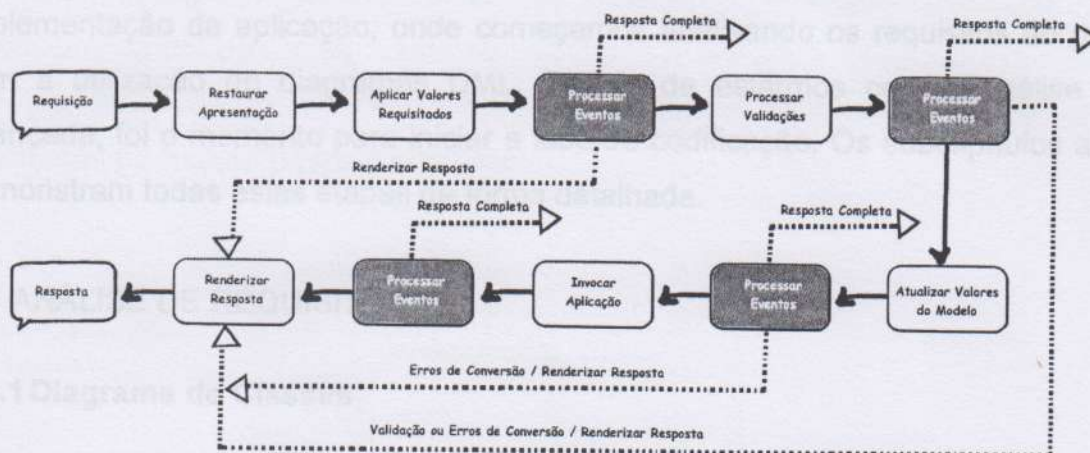
A grande vantagem do JSF está no modelo de componentes, pois uma mesma aplicação pode ser desenvolvida para o acesso de vários clientes como por exemplo web browser, tablet, celular e etc. Na especificação do JSF são fornecidos um conjunto de componentes básicos na biblioteca "HTML", como campos de texto, datagrid's, botões e etc, também inclui a biblioteca "Core" onde podemos realizar várias tarefas fundamentais como a validação e conversão de dados.

Os principais componentes que a implementação de referência do JSF fornece são: Formulário, campos de entrada de texto, campo de entrada de senha, rótulos, links, botões, mensagens , datagrids e etc.

No JSF usa-se uma classe java, intitulada como Managed Bean, a qual tem o objetivo de separar as regras de negócio da organização visual e do layout, fornecendo informações a serem exibidas na página web e métodos a serem executados.

A figura abaixo ilustra o ciclo de vida de uma requisição JSF.

Figura 6 – Ciclo de vida de uma requisição JSF



Fonte: <http://www.edsongoncalves.com.br/wp-content/uploads/2010/02/Figura-6-300x124.jpg>

4.6 TOMCAT

O Tomcat é um servidor de aplicações web para Java, surgiu por intermédio do projeto Jakarta da empresa Apache, possui código aberto assim faz parte da categoria de softwares livres. A Sun Microsystems oficializou o tomcat com implementação de referência para as tecnologias Java Server Pages (JSP) e Servlets que fazem parte da especificação J2EE.

5 METODOLOGIA

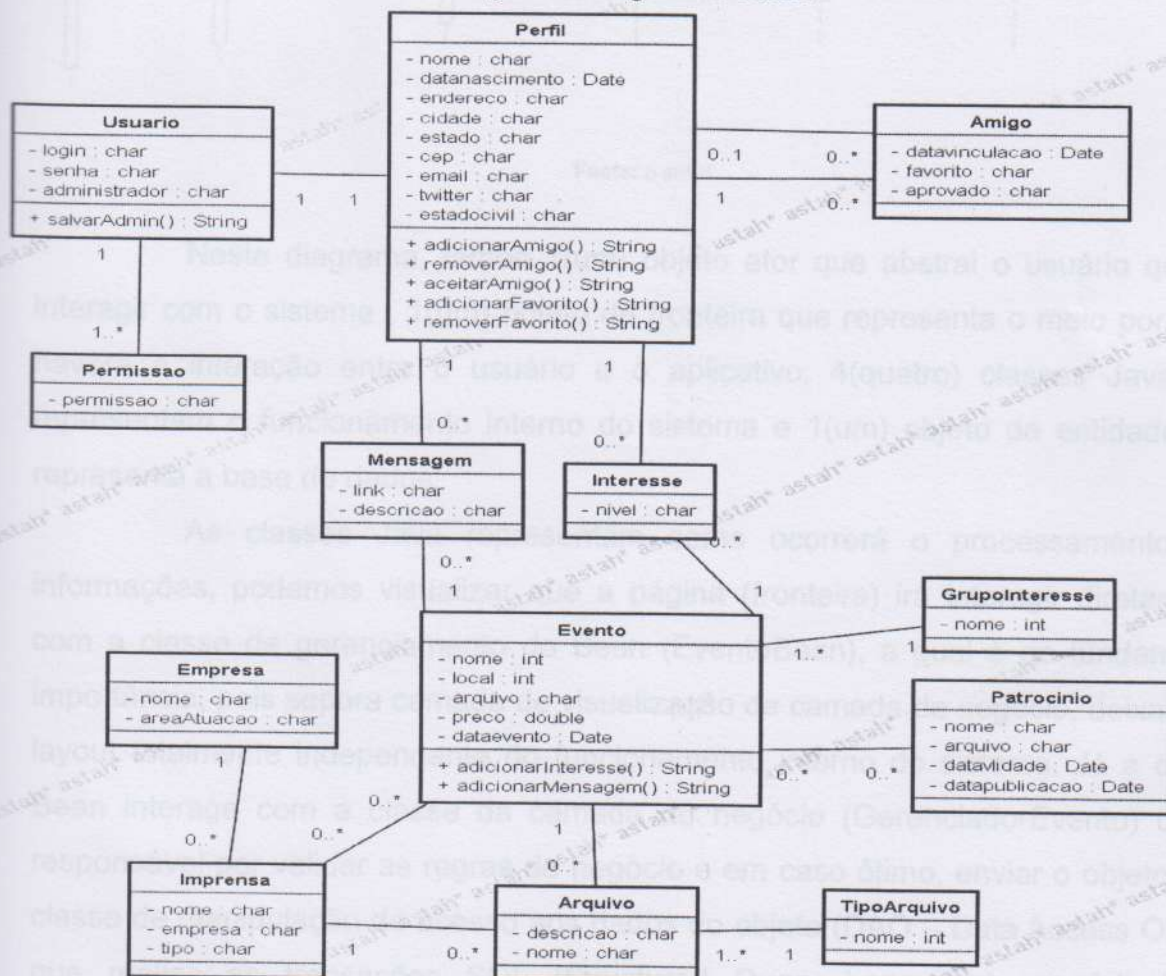
Neste capítulo são apresentados os detalhes de modelagem e implementação da aplicação, onde começamos analisando os requisitos do projeto com a utilização de diagramas UML. Depois de estarmos com a análise mais avançada, foi o momento para iniciar a fase de codificação. Os subcapítulos abaixo demonstram todas estas etapas de forma detalhada.

5.1 ANÁLISE DE REQUISITOS

5.1.1 Diagrama de Classes

Este diagrama mostra o sistema todo com suas classes, seus métodos, atributos bem como as suas cardinalidade dando uma visibilidade mais completa do sistema.

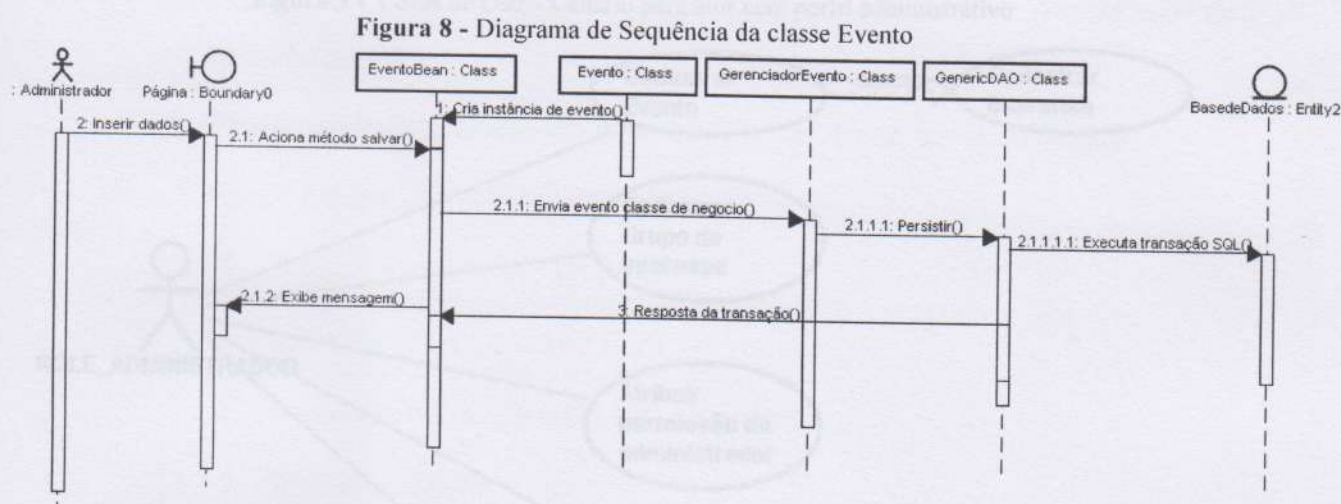
Figura 7 - Diagrama de Classes



Fonte: o autor

5.1.2 Diagrama de Sequência

O diagrama de sequência demonstra a ordem cronológica na qual os eventos ocorrem em referência a uma ou mais funcionalidades do sistema. Abaixo temos o exemplo de um diagrama que abstrai a maior parte das funcionalidades do sistema desenvolvido, o mesmo ilustra a sequência de ações para persistir um objeto do tipo Evento no banco de dados.



Fonte: o autor

Neste diagrama, temos 1(um) objeto ator que abstrai o usuário que irá interagir com o sistema, 1(um) objeto de fronteira que representa o meio por onde haverá a interação entre o usuário e o aplicativo, 4(quatro) classes Java que representam o funcionamento interno do sistema e 1(um) objeto de entidade que representa a base de dados.

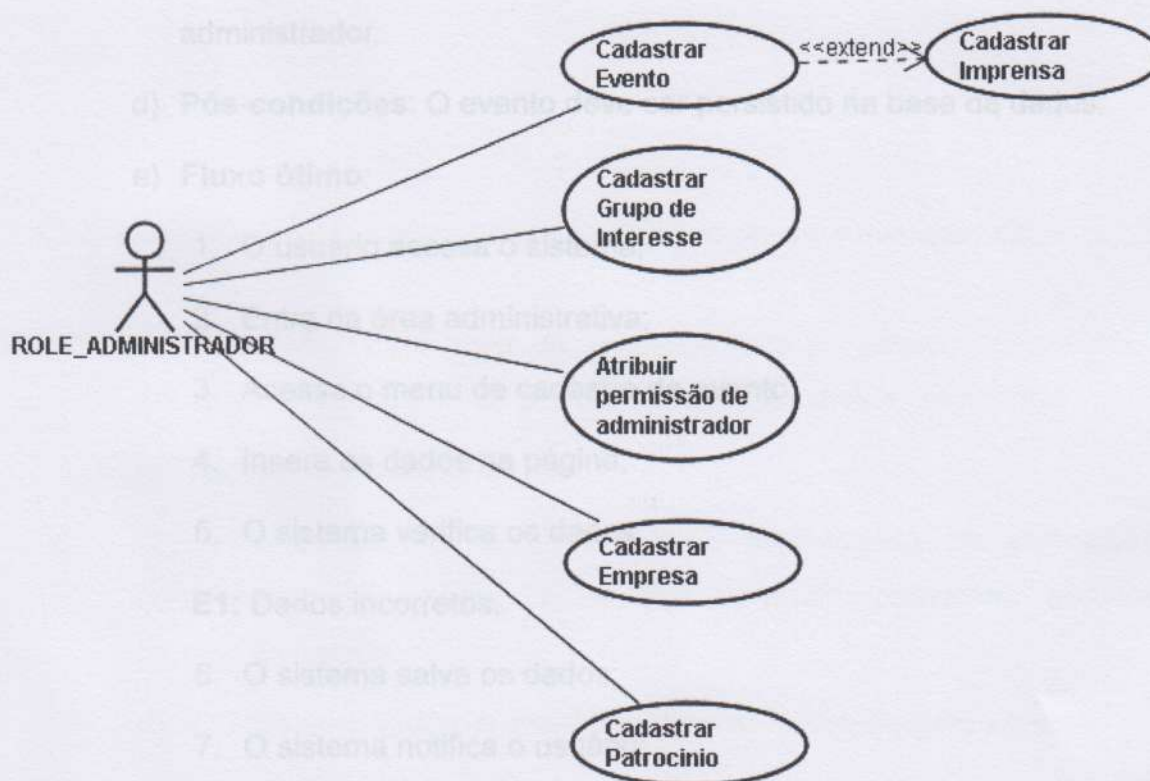
As classes Java representam como ocorrerá o processamento das informações, podemos visualizar que a página (fronteira) irá interagir diretamente com a classe de gerenciamento do Bean (EventoBean), a qual é de fundamental importância, pois separa camada de visualização da camada de negócio, deixando o layout totalmente independente do funcionamento interno do sistema. Já a classe Bean interage com a classe da camada de negócio (GerenciadorEvento) que é responsável por validar as regras de negócio e em caso ótimo, enviar o objeto para classe de manipulação de acesso aos dados do objeto (DAO – Data Access Object) que realiza as transações SQL (Structured Query Language), persistindo as informações no SGBD(Sistema Gerenciador de Banco de Dados).

5.1.3 Diagramas de Casos de Uso

Abaixo são expostos os diagramas de casos de uso, onde se demonstram as funcionalidades do sistema, os mesmos foram modelados por cenários, ou seja, de acordo com o perfil de funcionalidades que cada ator poderá executar no sistema.

5.1.3.1 Cenário de casos de uso para o ator com perfil de administrador

Figura 9 - Casos de Uso – Cenário para ator com perfil administrativo



Fonte: o autor

5.1.3.2 Tela de layout de cadastrar evento

Figura 10 - Tela para Cadastro de Evento

CADASTRAR EVENTO

Social Eventos

Todos se encontram aqui!

Cadastros ▾ Listagens ▾ Meu Perfil
⏻

Nome:

Local:

Data:

Preço: 0.0

Fonte: o autor

5.1.3.3 Descrição do caso de uso cadastrar evento

Caso de uso Cadastrar Evento

a) **Descrição:** Este caso de uso descreve as etapas percorridas pelo administrador para cadastrar um evento.

b) **Atores:** administrador.

c) **Pré-condições:** O usuário deve estar logado e ter permissão de administrador.

d) **Pós-condições:** O evento deve ser persistido na base de dados.

e) **Fluxo ótimo:**

1. O usuário acessa o sistema;
2. Entra na área administrativa;
3. Acessa o menu de cadastro de evento;
4. Insere os dados na página;
5. O sistema verifica os dados.

E1: Dados incorretos.

6. O sistema salva os dados;
7. O sistema notifica o usuário;

f) **Fluxo alternativo:**

E1: Dados incorretos.

1. O sistema mostra uma tela de erro devido a algum dado incorreto;
2. O usuário verifica as informações;
3. O sistema retorna para o passo 4 (quatro) do fluxo ótimo.

5.1.3.4 Tela de layout do caso de uso atribuir permissão administrativa

Figura 11- Tela para atribuir permissão de administrador

CADASTRO DE USUÁRIO ADMINISTRADOR
Social Eventos
 Todes se encontram aqui!

Cadastros ▾ Listagens ▾ Meu Perfil

Nome Usuário: inacio
 Administrador: ☐
 Salvar

Fonte: o autor

5.1.3.5 Descrição do caso de uso para atribuir permissão administrativa

a) **Descrição:** Este caso de uso descreve as etapas percorridas pelo administrador para atribuir permissão administrativa a um usuário.

b) **Atores:** administrador.

c) **Pré-condições:** O usuário deve está logado, ter permissão de administrador, o usuário a receber permissão deve ter somente a permissão padrão.

d) **Pós-condições:** A alteração deve ser validada pelo sistema.

e) **Fluxo ótimo:**

1. O usuário acessa o sistema;
2. Entra na área administrativa;
3. Acessa o menu de listagens e busca o usuário que receberá a permissão.
4. Clica em adicionar administrador;
5. Marca a opção de checagem para atribuir a permissão.
6. O sistema verifica os dados.

E1: Dados incorretos.

7. O sistema salva os dados;

8. O sistema notifica o usuário;

f) **Fluxo alternativo:**

E1: Dados incorretos.

1. O sistema mostra uma tela de erro devido a algum dado incorreto;
2. O usuário verifica as informações;
3. O sistema retorna para o passo 4 (quatro) do fluxo ótimo.

5.1.3.6 Tela de layout do caso de uso cadastrar grupo interesse

Figura 12 - Tela para cadastrar um grupo de interesse

CADASTRAR GRUPO DE INTERESSE

Social Eventos
Tudo se encontra aqui!

Cadastros ▾ Listagens ▾ Meu Perfil

Nome:

Salvar

Fonte: o autor

5.1.3.7 Descrição do caso de uso cadastrar grupo de interesse

- a) **Descrição:** Este caso de uso descreve as etapas percorridas pelo administrador para cadastrar um grupo de interesse.
- b) **Atores:** administrador.
- c) **Pré-condições:** O usuário deve estar logado e ter permissão de administrador.
- d) **Pós-condições:** O grupo de interesse deve ser persistido na base de dados.
- e) **Fluxo ótimo:**
 1. O usuário acessa o sistema;
 2. Entra na área administrativa;
 3. Acessa o menu de cadastro de grupo de interesse;

4. Insere os dados na página;

5. O sistema verifica os dados.

E1: Dados incorretos.

6. O sistema salva os dados;

7. O sistema notifica o usuário;

f) **Fluxo alternativo:**

E1: Dados incorretos.

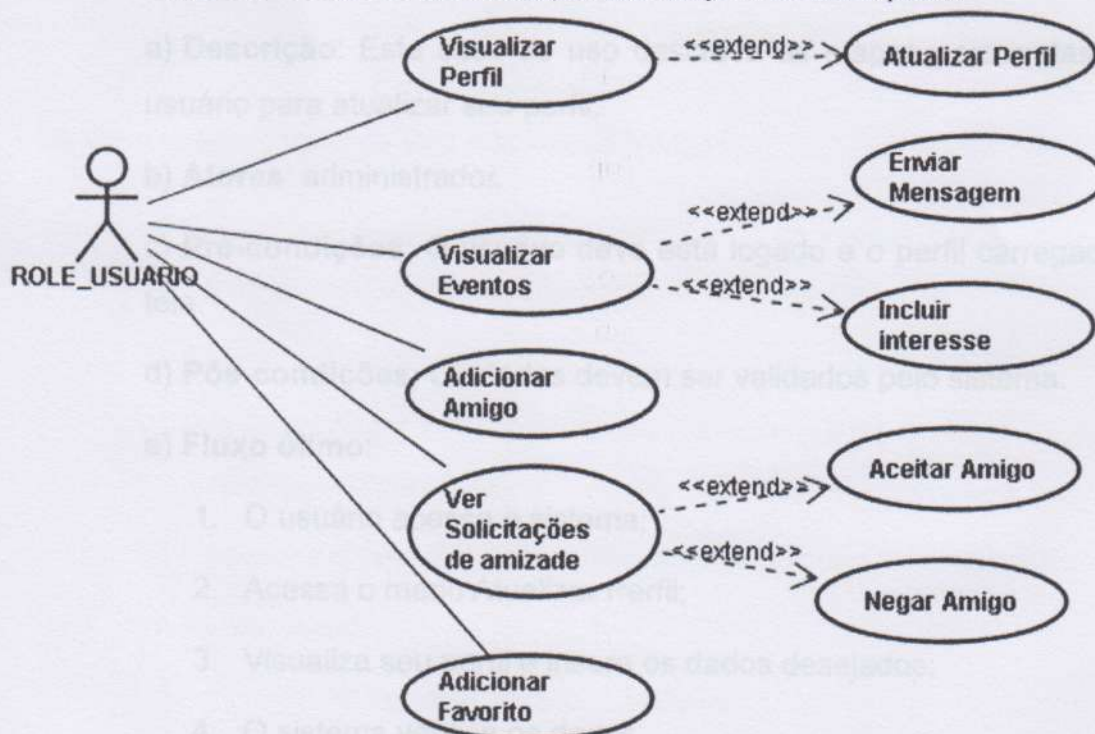
1. O sistema mostra uma tela de erro devido a algum dado incorreto;

2. O usuário verifica as informações;

3. O sistema retorna para o passo 4 (quatro) do fluxo ótimo.

5.1.3.8 Cenário de casos de usos para o ator com perfil de usuário padrão

Figura 13 - Cenário de casos de uso para ator com perfil de usuário padrão



Fonte: o autor

5.1.3.9 Tela de visualização e atualização de perfil

Figura 14 - Tela de atualização de perfil

ATUALIZAR PERFIL

Home Perfil ▾ Amigos ▾ Eventos

Nome: INACIO DE LOIOLA

Endereço:

Cidade: Digite sua cidade

Estado:

Data Nascimento:

Estado Civil:

Email:

Twitter:

Pesquisar...

Foto: [Upload](#) | [Clear](#)

Salvar

Fonte: o autor

5.1.3.10 Descrição de caso de uso atualizar perfil

a) **Descrição:** Este caso de uso descreve as etapas percorridas pelo usuário para atualizar seu perfil.

b) **Atores:** administrador.

c) **Pré-condições:** O usuário deve estar logado e o perfil carregado na tela.

d) **Pós-condições:** Os dados devem ser validados pelo sistema.

e) **Fluxo ótimo:**

1. O usuário acessa o sistema;
2. Acessa o menu Atualizar Perfil;
3. Visualiza seu perfil e insere os dados desejados;
4. O sistema verifica os dados.

E1: Dados incorretos.

5. O sistema salva os dados;
6. O sistema notifica o usuário;

f) **Fluxo alternativo:****E1:** Dados incorretos.

1. O sistema mostra uma tela de erro devido a algum dado incorreto;
2. O usuário verifica as informações;
3. O sistema retorna para o passo 3 (três) do fluxo ótimo.

5.1.3.11 Tela de visualização de eventos

Figura 15 - Tela de visualização de eventos

EVENTOS **Social Eventos**
Todos se encontram aqui!

Home Perfil ▾ Amigos ▾ Eventos  

Lista de Eventos					
Nome	Data	Local	Preço		
DORGIVAL DANTAS	08/11/2012	ACARI	20.0	Adicionar Interesse	Postar Mensagem
O ESPETACULAR HOMEM ARANHA	08/09/2012	NATAL-RN	18.0	Adicionar Interesse	Postar Mensagem
CARNATAL	12/06/2012	NATAL-RN	200.0	Adicionar Interesse	Postar Mensagem

Fonte: o autor

5.1.3.12 Descrição do caso de uso adicionar interesse

- a) **Descrição:** Este caso de uso descreve as etapas percorridas pelo usuário para adicionar interesse em um evento.
- b) **Atores:** administrador.
- c) **Pré-condições:** O usuário deve está logado e deve existir ao menos um evento disponível.
- d) **Pós-condições:** Os dados devem ser validados pelo sistema.
- e) **Fluxo ótimo:**
 1. O usuário acessa o sistema;
 2. Acessa o menu Eventos;
 3. Visualiza os eventos disponíveis e clica no botão adicionar interesse no evento desejado.

4. Escolhe o nível de interesse no evento e clica em salvar.

5. O sistema verifica os dados.

E1: Dados incorretos.

6. O sistema salva os dados;

7. O sistema notifica o usuário;

f) Fluxo alternativo:

E1: Dados incorretos.

1. O sistema mostra uma tela de erro devido a algum dado incorreto;

2. O usuário verifica as informações;

3. O sistema retorna para o passo 4(três) do fluxo ótimo.

5.1.3.13 Descrição do caso de uso postar mensagem

a) **Descrição:** Este caso de uso descreve as etapas percorridas pelo usuário para postar mensagem em um evento.

b) **Atores:** administrador.

c) **Pré-condições:** O usuário deve está logado, deve existir ao menos um evento disponível.

d) **Pós-condições:** Os dados devem ser validados pelo sistema.

e) Fluxo ótimo:

1. O usuário acessa o sistema;

2. Acessa o menu Eventos;

3. Visualiza os eventos disponíveis e clica no botão postar mensagem no evento desejado.

4. Insere os dados da mensagem.

5. O sistema verifica os dados.

E1: Dados incorretos.

6. O sistema salva os dados;

7. O sistema notifica o usuário;

f) Fluxo alternativo:

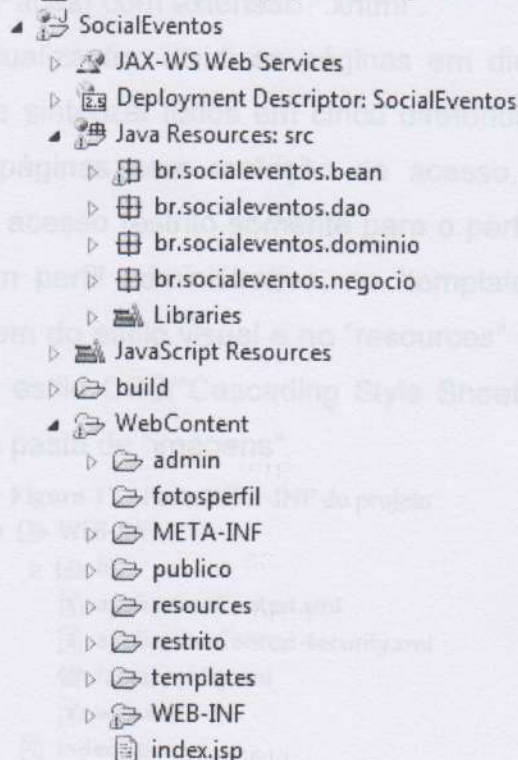
E1: Dados incorretos.

- 1) O sistema mostra uma tela de erro devido a algum dado incorreto;
- 2) O usuário verifica as informações;
- 3) O sistema retorna para o passo 4(três) do fluxo ótimo.

5.2 IMPLEMENTAÇÃO

Após analisar o sistema e desenvolver alguns diagramas que modelam o seu funcionamento, foi iniciada a fase de codificação. Para implementação foi utilizado como IDE (Integrated Development Environment), o Eclipse, versão Helios, o qual fornece um ambiente integrado de desenvolvimento que facilita e agiliza essa importante etapa da elaboração do projeto.

Figura 16 - Exemplo de como o projeto foi implementado



Fonte: o autor

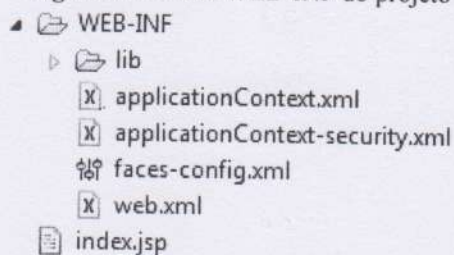
A ilustração na página anterior (figura 16) demonstra como foi organizada a aplicação e como foram criados os pacotes de forma a garantir que as classes fiquem organizadas de acordo com suas funções, sendo assim: temos no pacote "br.socialeventos.bean" somente as classes Bean da aplicação, no "br.socialeventos.dao" estão as classes de acesso a dados, no "br.socialeventos.dominio" temos as classes de entidade ou modelo e em "br.socialeventos.negocio" se encontram as classes que validam as regras de negócio.

O padrão de projeto MVC (Model – View - Control), prega a divisão da aplicação em três camadas como forma de facilitar a manutenibilidade, tornando a aplicação escalável e segmentado a visualização do controle de fluxo, e o mesmo do modelo e negócio da aplicação, ou seja, podemos implementar várias formas visuais de interação com o sistema sem precisar alterar suas classes de funcionamento interno.

Neste projeto utilizo o padrão MVC, de forma que os pacotes "br.socialeventos.dominio", "br.socialeventos.dao" e "br.socialeventos.negocio" constituem o modelo, o pacote "br.socialeventos.bean" trata do controle de fluxo e no diretório WebContent tenho a minha camada de visualização, representada pelas páginas JSF (Java Server Faces) com extensão ".xhtml".

Na parte da visualização, dividi as páginas em diretórios referentes as suas respectivas classes e sintetizei todos em cinco diretórios principais, que são: "publico" onde ficam as páginas sem restrição de acesso, "restrito" que estão presentes as páginas com acesso restrito somente para o perfil de usuário padrão, "admin" para usuários com perfil administrativo, no "templates" ficam as páginas responsáveis pela montagem do estilo visual e no "resources" são armazenados em uma pasta os arquivos de estilo CSS("Cascading Style Sheets") que configuram a apresentação de layout e a pasta de "imagens".

Figura 17 - Pasta WEB-INF do projeto



Fonte: o autor

6 CONCLUSÃO

Dentro da pasta WebContent, temos também o diretório WEB-INF, que é responsável por armazenar de forma segura os arquivos de configuração (Anexo A) da aplicação e a página padrão "index.jsp", que é chamada ao iniciar o servidor web e redireciona para a página de login (Anexo B), além das bibliotecas utilizadas no projeto, como no nosso sistema por exemplo: as bibliotecas do hibernate, spring, JSF e etc. Nessa pasta temos o arquivo de configuração web.xml que define regras e parâmetros da aplicação, o applicationContext.xml que define as configurações do Spring Framework, o faces-config.xml que parametriza o JSF e o applicationContext-security.xml que é utilizado pelo Spring Security para configurar as formas de acesso ao sistema, permissões e autenticação de usuários.

6 CONCLUSÃO

Como visto no decorrer dos capítulos, este projeto tratou elaboração de um protótipo de sistema focado em redes sociais interessadas em eventos, o mesmo foi desenvolvido com o propósito de servir como objeto avaliativo para obtenção do título de Especialista em Desenvolvimento de Sistemas Corporativos, concedido pela Faculdade Natalense para o Desenvolvimento do Rio Grande Do Norte.

Neste projeto utilizei tecnologias utilizadas no desenvolvimento de sistemas corporativos a fim de aprender mais sobre as mesmas, trazendo como benefícios a aprendizagem adquirida e a atualização profissional que me torna mais competitivo no âmbito do mercado de trabalho.

Algumas dificuldades foram vivenciadas no decorrer do projeto, como a aptidão na linguagem Java, entre outras, mas que com empenho e perseverança consegui ultrapassar. Este projeto pode ainda ser objeto de estudo, no sentido de trazer melhorias para o mesmo, tendo em vista uma gama de possibilidades e inovações que podem ser implementadas.

ANEXO A – A CONFIGURAÇÃO REFERÊNCIAS

BEZERRA, Eduardo. **Princípios de análise e projeto de sistemas com UML**. Rio de Janeiro: Elsevier, 2002.

LUCKOW, Décio Heinzemann; MELO, **Web Programação Java para**. São Paulo, SP: Novatec, 2010.

MARCELO, Antonio. **Apache - Configurando o servidor WEB para Linux**. Rio de Janeiro, RJ: BRASPORT, 2006.

MILANI, André. **Livro PostgreSQL: Guia do Programador**. São Paulo: Novatec, 2008.

SANTANA, Camila Lima S. e. **Redes Sociais na internet: potencializando interações sociais**. In: SIMPOSIO HIPERTEXTO, EDUCAÇÃO E TECNOLOGIAS, 1., 2006, Recife. Revista do NEHTE, 2006.

THORWALD, Gilleanes. **UML: Uma Abordagem Prática**. São Paulo, SP: Novatec, 2006.

TOMAÉL, Maria Inês; ALCARÁ, Adriana Rosecler ; DI CHIARA, Ivone Guerreiro . Das redes sociais à inovação. **Ciência da Informação**, Brasília, v. 34, n.2, p. 93-104, 2005.

ANEXO A – ARQUIVOS DE CONFIGURAÇÃO

Arquivo de configuração da aplicação

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://java.sun.com/xml/ns/javaee"
xmlns:web="http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd"
xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd" id="WebApp_ID"
version="2.5">
  <display-name>SocialEventos</display-name>
  <welcome-file-list>
    <welcome-file>index.html</welcome-file>
    <welcome-file>index.htm</welcome-file>
    <welcome-file>index.jsp</welcome-file>
    <welcome-file>default.html</welcome-file>
    <welcome-file>default.htm</welcome-file>
    <welcome-file>default.jsp</welcome-file>
  </welcome-file-list>
  <listener>
    <listener-class>
      org.springframework.web.context.ContextLoaderListener
    </listener-class>
  </listener>
  <listener>
    <listener-class>
      org.springframework.web.context.request.RequestContextListener
    </listener-class>
  </listener>
  <servlet>
    <servlet-name>Faces Servlet</servlet-name>
    <servlet-class>javax.faces.webapp.FacesServlet</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>Faces Servlet</servlet-name>
    <url-pattern>*.jsf</url-pattern>
  </servlet-mapping>
  <!-- Spring Security -->
  <context-param>
    <param-name>contextConfigLocation</param-name>
    <param-value>
      /WEB-INF/applicationContext.xml
      /WEB-INF/applicationContext-security.xml
    </param-value>
  </context-param>
  <filter>
    <filter-name>springSecurityFilterChain</filter-name>
    <filter-
class>org.springframework.web.filter.DelegatingFilterProxy</filter-class>
  </filter>
  <filter-mapping>
    <filter-name>springSecurityFilterChain</filter-name>
    <url-pattern>*/</url-pattern>
```

```

        </filter-mapping>

        <!-- File Upload -->
        <filter>
            <filter-name>PrimeFaces FileUpload Filter</filter-name>

            <filter-class> org.primefaces.webapp.filter.FileUploadFilter</filter-class>
        </filter>

        <filter-mapping>
            <filter-name>PrimeFaces FileUpload Filter</filter-name>
            <servlet-name>Faces Servlet</servlet-name>
        </filter-mapping>

    </web-app>

```

Arquivo de configuração do Spring Framework

```

<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xmlns:context="http://www.springframework.org/schema/context"
       xmlns:jdbc="http://www.springframework.org/schema/jdbc"
       xmlns:p="http://www.springframework.org/schema/p"
       xmlns:tx="http://www.springframework.org/schema/tx"
       xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans-3.0.xsd
http://www.springframework.org/schema/context
http://www.springframework.org/schema/context/spring-context-3.0.xsd
http://www.springframework.org/schema/jdbc
http://www.springframework.org/schema/jdbc/spring-jdbc-3.0.xsd
http://www.springframework.org/schema/tx
http://www.springframework.org/schema/tx/spring-tx-3.0.xsd">

    <context:component-scan base-package="br.socialeventos"/>

    <bean id="dataSource"
class="org.springframework.jdbc.datasource.DriverManagerDataSource">
        <property name="driverClassName"
value="org.postgresql.Driver"/>
        <property name="url"
value="jdbc:postgresql://localhost:5432/socialeventos"/>
        <property name="username" value="socialeventos"/>
        <property name="password" value="socialeventos"/>
    </bean>

    <bean id="jdbcTemplate"
class="org.springframework.jdbc.core.JdbcTemplate">
        <property name="dataSource" ref="dataSource"/>
    </bean>

    <bean id="sessionFactory"

class="org.springframework.orm.hibernate3.annotation.AnnotationSessi
onFactoryBean">

```

```

        <property name="dataSource" ref="dataSource"/>
        <property name="packagesToScan"
value="br.socialeventos.dominio"/>

        <property name="hibernateProperties">
            <props>
                <prop key="hibernate.show_sql">true</prop>
                <prop
key="hibernate.dialect">org.hibernate.dialect.PostgreSQLDialect</prop>

                <prop key="hibernate.hbm2ddl.auto">create-
drop</prop>
            </props>
        </property>
    </bean>

    <bean id="hibernateTemplate"
        class="org.springframework.orm.hibernate3.HibernateTemplate">
        <property name="sessionFactory" ref="sessionFactory"/>
    </bean>

    <bean id="transactionManager"

        class="org.springframework.orm.hibernate3.HibernateTransactionManage
r">
        <property name="sessionFactory" ref="sessionFactory"/>
    </bean>

    <tx:annotation-driven transaction-manager="transactionManager"/>

</beans>

```

Arquivo de configuração do JSF

```

<?xml version="1.0"?>
<faces-config xmlns="http://java.sun.com/xml/ns/javaee"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-facesconfig_2_0.xsd"
    version="2.0">

    <application>
        <el-
resolver>org.springframework.web.jsf.el.SpringBeanFacesELResolver</el-
resolver>
    </application>

</faces-config>

```

Arquivo de configuração do Spring Security

```

<?xml version="1.0" encoding="UTF-8"?>
<b:beans xmlns="http://www.springframework.org/schema/security"
    xmlns:b="http://www.springframework.org/schema/beans"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

```

```

        xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans-3.0.xsd
        http://www.springframework.org/schema/security
http://www.springframework.org/schema/security/spring-security-3.0.xsd">
        <http>

        <intercept-url pattern="/admin/**" access="ROLE_ADMINISTRADOR" />
        <intercept-url pattern="/restrito/**" access="ROLE_USUARIO"
/>

        <form-login login-page="/publico/logins.jsf"
        always-use-default-target="true" default-target-
url="/restrito/mypage.jsf"
        authentication-failure-
url="/publico/logins.jsf?login_error=1" />
        <logout logout-success-url="/publico/logins.jsf" invalidate-
session="true" />
        <remember-me />
        <access-denied-handler error-
page="/publico/acessonegado.jsf"/>
        </http>

        <authentication-manager>
        <authentication-provider>
        <jdbc-user-service data-source-ref="dataSource"
        authorities-by-username-query="SELECT u.login, p.permissao FROM
usuario u, permissao p WHERE u.id = p.id_usuario AND u.login = ?"
        users-by-username-query="SELECT u.login, u.senha, u.ativo FROM
usuario u WHERE u.login = ?" />

        </authentication-provider>
        </authentication-manager>
</b:beans>

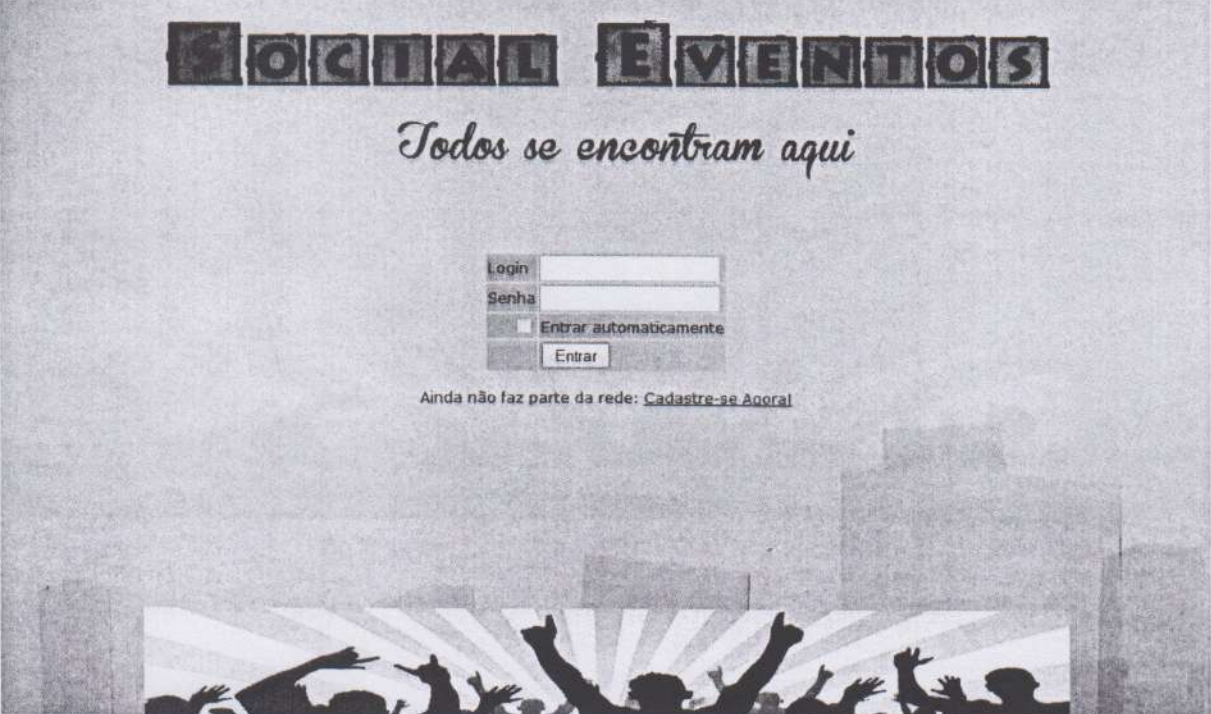
```

Tela Principal do Usuário



ANEXO B – TELAS DO SISTEMA

Tela de Login



SOCIAL EVENTOS

Todos se encontram aqui

Login

Senha

☐ Entrar automaticamente

Ainda não faz parte da rede: [Cadastre-se Agora!](#)

The login screen features a header with the site name 'SOCIAL EVENTOS' in a bold, blocky font, followed by the tagline 'Todos se encontram aqui' in a script font. Below this is a login form with fields for 'Login' and 'Senha', a checkbox for 'Entrar automaticamente', and an 'Entrar' button. A link for 'Cadastre-se Agora!' is positioned below the form. The background is a grayscale image of a city skyline with silhouettes of people raising their hands in the foreground.

Tela Principal do Usuário



BEM VINDO INACIO

SOCIAL EVENTOS
Todos se encontram aqui

Home Perfil * Amigos * Eventos  

PRÓXIMOS EVENTOS:

The user main screen displays a personalized welcome message 'BEM VINDO INACIO' on the left and the site logo and tagline on the right. A navigation bar contains links for 'Home', 'Perfil', 'Amigos', and 'Eventos', along with a user profile icon and a logout icon. Below the navigation bar, there is a section titled 'PRÓXIMOS EVENTOS:'. The background is the same grayscale city skyline image with silhouettes of people raising their hands.

Tela Principal do Administrador

