



Fundação Educacional do Município de Assis
Instituto Municipal de Ensino Superior de Assis
Campus "José Santilli Sobrinho"

FREDERICO MANOEL BERTOLUCI REIS

**Desenvolvimento de um sistema Web em Java para criação de
módulos baseados em conceitos de segurança da informação.**

Assis
2014

FREDERICO MANOEL BERTOLUCI REIS

Desenvolvimento de um sistema Web em Java para criação de módulos baseados em conceitos de segurança da informação.

Projeto de pesquisa apresentado ao curso de Bacharelado em Ciência da Computação do Instituto Municipal de Ensino Superior de Assis - IMESA e a Fundação Educacional do Município de Assis – FEMA, como requisito parcial à obtenção do Certificado de Conclusão.

Orientador: Me. Douglas Sanches da Cunha

Área de Concentração: Segurança da Informação

Assis
2014

FICHA CATALOGRÁFICA

REIS, FREDERICO MANOEL BERTOLUCI

Desenvolvimento de um sistema Web em Java para criação de módulos baseados em conceitos de segurança da informação/ Frederico Manoel Bertoluci Reis. Fundação Educacional do Município de Assis – FEMA - Assis, 2014.
81p.

Orientador: Me. Douglas Sanches da Cunha.
Trabalho de Conclusão de Curso – Instituto Municipal de Ensino Superior de Assis - IMESA

1. Segurança da informação. 2. Sistema web. 3. Java web.

CDD: 001.6

Biblioteca da FEMA

Desenvolvimento de um sistema Web em Java para criação de módulos baseados em conceitos de segurança da informação.

FREDERICO MANOEL BERTOLUCI REIS

Trabalho de Conclusão de Curso apresentado ao Instituto Municipal de Ensino Superior de Assis, como requisito do Curso de Bacharelado em Ciência da Computação, analisado pela seguinte comissão examinadora:

Orientador: Me. Douglas Sanches da Cunha

Analisador: Celio Desiro

Assis
2014

DEDICATÓRIA

Dedico este trabalho à Ana Maria Lange Gomes, pessoa com quem amo partilhar a vida. Com você tenho me sentido mais vivo de verdade. Obrigado pelo carinho, a paciência e por sua capacidade de me trazer paz na correria de cada semestre.

AGRADECIMENTOS

Ao Mestre Douglas Sanches da Cunha pela oportunidade e apoio na elaboração deste trabalho. À Instituição pelo ambiente criativo e amigável que me proporcionou. Agradeço a minha mãe Rosa Elizabeth Bertoluci, heroína que me deu apoio e incentivo nas horas difíceis, de desânimo e cansaço. Ao meu pai Nivaldo Bertozo Reis, que apesar de todas as dificuldades me fortaleceu e que pra mim foi muito importante.

A uma mulher muito especial que me incentivou desde o início do curso, Ana Maria Lange Gomes, que sem ela eu não teria conseguido chegar até aqui. E a todos que direta ou indiretamente fizeram parte da minha formação, o meu muito obrigado.

Talvez não tenha conseguido fazer o melhor, mas lutei para que o melhor fosse feito. Não sou o que deveria ser, mas Graças a Deus, não sou o que era antes.

Marthin Luther King
(1929-1968)

RESUMO

Anteriormente a questão segurança da informação não era muito complexa, pois a informação era guardada em inúmeros papéis que podiam ser trancados fisicamente, porém com a chegada da era tecnológica e da comunicação, a questão ficou bem mais complexa, hoje a maioria dos dispositivos se conecta a internet e conseqüentemente a internet se conecta a eles, além disto, sabemos que dados em formato digital são portáteis, este fato fez que a informação se tornassem atrativas para ladrões.

Infelizmente o que vemos na prática é que muitas empresas não dão o devido valor a esta questão e por muitas vezes o preço é muito alto, portanto o melhor caminho é reduzir ao máximo quaisquer riscos à informação, seguindo um trajeto no sentido único de manter a integridade e a disponibilidade dos sistemas de informação. Para se implantar uma eficaz segurança da informação dentro de uma organização devemos ficar atentos para algumas questões como uma boa análise de riscos, a definição da Política de Segurança e por fim um plano de contingência.

Com isto, este trabalho visa mostrar algumas regras que devemos adotar como padrões de projeto para se garantir uma boa segurança nos softwares web.

Palavras chave: Segurança da informação, Sistema web, Java web.

ABSTRACT

Previously the issue of information security was not very complicated, because the information was stored in many roles that could be locked physically, but with the arrival of the technological era and communication, the issue was much more complex, today most of the devices connects internet and consequently the internet connects to them, moreover, we know that digital data are portable, this fact did that information become attractive to thieves.

Unfortunately what we see in practice is that many companies do not give due weight to this issue and often the price is very high, so the best way is to reduce the maximum any risks to information, following a single path in order to maintain integrity and availability of information systems. To implement an effective information security within an organization should be alert to such issues as a good risk analysis, the definition of the Security Policy and lastly a contingency plan.

With this, this paper aims to show some rules we should adopt as design patterns to ensure good security software on the web.

Keywords: Information security, web system, Java web.

LISTA DE ILUSTRAÇÕES

| | |
|--|----|
| Figura 1 - Caso de Uso: Efetuar o Login | 35 |
| Figura 2 - Diagrama de Sequencia: Efetuar o Login | 36 |
| Figura 3- Caso de Uso: Favoritos | 39 |
| Figura 4- Diagrama de Sequencia: Favoritos | 41 |
| Figura 5- Caso de Uso: Eventos | 44 |
| Figura 6- Diagrama de Sequencia: Evento | 45 |
| Figura 7- Caso de Uso: Perfil | 47 |
| Figura 8- Diagrama de Sequencia: Perfil | 48 |
| Figura 9- Caso de Uso: Cadastrar Usuário | 50 |
| Figura 10- Diagrama de Sequencia: Cadastrar Usuário | 51 |
| Figura 11- Caso de Uso: Cadastrar Grupo | 52 |
| Figura 12- Diagrama de Sequencia: Cadastrar Grupo | 53 |
| Figura 13- Caso de Uso: Cadastrar Sistema | 54 |
| Figura 14- Diagrama de Sequencia: Cadastrar Sistemas | 55 |
| Figura 15- Caso de Uso: Cadastrar Eventos | 56 |
| Figura 16- Diagrama de Sequencia: Cadastrar Eventos | 57 |
| Figura 17- Caso de Uso: Cadastrar Permissões | 58 |
| Figura 18- Diagrama de Sequencia: Cadastrar Permissões | 60 |
| Figura 19- Caso de Uso: Cadastrar Usuários/Grupos | 61 |
| Figura 20- Diagrama de Sequencia: Cadastrar Usuários/Grupos | 62 |
| Figura 21- Diagrama de Caso de Uso Global | 63 |
| Figura 22- Diagrama de Pacotes | 64 |
| Figura 23- Diagrama de Classes do Pacote útil | 65 |
| Figura 24- Diagrama de Classes do Pacote dao | 66 |
| Figura 25- Diagrama de Classes do Pacote model | 67 |
| Figura 26- Diagrama de Classes do Pacote manageBean | 68 |
| Figura 27- Diagrama de Atividade | 69 |
| Figura 28- Diagrama WBS | 70 |
| Figura 29- Sequenciamento das Atividades Definidas | 71 |
| Figura 30 – LOGIN | 80 |
| Figura 31 – MENU | 80 |
| Figura 32 - CADASTRO DE USUARIO | 81 |
| Figura 33 - CADASTRO DE PERMISSÕES | 81 |

LISTA DE TABELAS

| | |
|--|-----------|
| Tabela 1- Orçamento do Projeto..... | 75 |
|--|-----------|

SUMÁRIO

| | | |
|---------------|---|------------|
| 1. | INTRODUÇÃO | 175 |
| 1.1. | OBJETIVO..... | 17 |
| 1.1.1. | OBJETIVO GERAL | 17 |
| 1.1.2 | OBJETIVOS ESPECÍFICOS..... | 17 |
| 1.2 | JUSTIFICATIVA..... | 17 |
| 1.3 | PROBLEMA..... | 18 |
| 1.4 | ESTRUTURA DE DESENVOLVIMENTO | 18 |
| 2. | REVISÃO DA LITERATURA | 20 |
| 3. | TECNOLOGIAS, FERRAMENTAS E METODOLOGIAS UTILIZADAS. | 23 |
| 3.1. | TECNOLOGIAS E FERRAMENTAS | 23 |
| 3.1.1. | NETBEANS..... | 23 |
| 3.1.2. | JAVA | 24 |
| 3.1.3. | POSTGRESQL..... | 25 |
| 3.1.4. | PGADMIN 3 | 26 |
| 3.1.5. | JAVA SERVER FACES (JSF) | 27 |
| 3.1.6. | PRIMEFACES..... | 28 |
| 3.1.7. | ASTAH COMMUNITY | 28 |
| 3.2. | METODOLOGIAS DE ANÁLISE | 28 |
| 3.2.1. | CASOS DE USO..... | 29 |
| 3.2.2. | DIAGRAMA DE CLASSES..... | 29 |
| 3.2.3. | DIAGRAMA DE ATIVIDADES..... | 29 |
| 3.2.4. | SEQUENCIAMENTO DE ATIVIDADES | 29 |
| 4. | DOCUMENTAÇÃO | 30 |
| 4.1. | LEVANTAMENTO DOS REQUISITOS..... | 30 |
| 4.1.1. | FORMA ADOTADA | 30 |
| 4.1.2. | ANÁLISE DOS REQUISITOS | 31 |

| | | |
|-----------|--|----|
| 4.1.2.1. | PROBLEMAS POTENCIAIS | 31 |
| 4.1.2.2. | RESTRICÇÕES DO SISTEMA | 31 |
| 4.1.3. | VALIDAÇÃO DOS REQUISITOS | 32 |
| 4.1.4. | ESPECIFICAÇÃO DOS REQUISITOS | 32 |
| 4.1.4.1. | ESCOPO..... | 33 |
| 4.1.4.2. | LISTA DE EVENTOS | 33 |
| 4.2. | ANÁLISE ORIENTADA A OBJETOS..... | 34 |
| 4.2.1. | ESPECIFICAÇÃO DOS CASOS DE USOS E DIAGRAMAS DE SEQUENCIA | 34 |
| 4.2.1.1. | EFETUAR O LOGIN | 35 |
| 4.2.1.2. | DIAGRAMA DE SEQUENCIA: REALIZAR LOGIN..... | 36 |
| 4.2.1.3. | VISUALIZAR MENSAGENS | 37 |
| 4.2.1.4. | VISUALIZAR EVENTOS FAVORITOS | 37 |
| 4.2.1.5. | MENU DO SOFTWARE | 38 |
| 4.2.1.6. | FAVORITOS | 39 |
| 4.2.1.7. | DIAGRAMA DE SEQUENCIA: FAVORITOS | 41 |
| 4.2.1.8. | ADICIONAR EVENTOS AOS FAVORITOS | 41 |
| 4.2.1.9. | LISTAR EVENTOS FAVORITOS | 42 |
| 4.2.1.10. | REMOVER EVENTOS AOS FAVORITOS | 43 |
| 4.2.1.11. | EVENTOS | 44 |
| 4.2.1.12. | DIAGRAMA DE SEQUENCIA: EVENTO..... | 45 |
| 4.2.1.13. | LISTAR EVENTOS | 45 |
| 4.2.1.14. | EXIBIR EVENTO SELECIONADO | 46 |
| 4.2.1.15. | PERFIL..... | 47 |
| 4.2.1.16. | DIAGRAMA DE SEQUENCIA: PERFIL | 48 |
| 4.2.1.17. | ALTERAR DADOS DO USUÁRIO..... | 49 |
| 4.2.1.18. | CADASTRAR USUÁRIO..... | 50 |
| 4.2.1.19. | DIAGRAMA DE SEQUENCIA: CADASTRAR USUÁRIO | 51 |
| 4.2.1.20. | CADASTRAR GRUPO..... | 52 |

| | | |
|-----------|--|----|
| 4.2.1.21. | DIAGRAMA DE SEQUENCIA: CADASTRAR GRUPO | 53 |
| 4.2.1.22. | CADASTRAR SISTEMAS | 54 |
| 4.2.1.23. | DIAGRAMA DE SEQUENCIA: CADASTRAR SISTEMAS..... | 55 |
| 4.2.1.24. | CADASTRAR EVENTO | 56 |
| 4.2.1.25. | DIAGRAMA DE SEQUENCIA: CADASTRAR EVENTOS..... | 57 |
| 4.2.1.26. | CADASTRAR PERMISSÕES | 58 |
| 4.2.1.27. | DIAGRAMA DE SEQUENCIA: CADASTRAR PERMISSÕES | 60 |
| 4.2.1.28. | CADASTRAR USUÁRIOS/GRUPOS | 61 |
| 4.2.1.29. | DIAGRAMA DE SEQUENCIA: CADASTRAR USUÁRIOS/GRUPOS.. | 62 |
| 4.2.2. | DIAGRAMA DE CASO DE USO GLOBAL | 63 |
| 4.2.3. | DIAGRAMA DE PACOTES | 64 |
| 4.2.4. | DIAGRAMA DE CLASSE..... | 65 |
| 4.2.4.1. | DIAGRAMA DE CLASSES DO PACOTE UTIL..... | 65 |
| 4.2.4.2. | DIAGRAMA DE CLASSES DO PACOTE DAO..... | 66 |
| 4.2.4.3. | DIAGRAMA DE CLASSES DO PACOTE MODEL..... | 67 |
| 4.2.4.4. | DIAGRAMA DE CLASSES DO PACOTE MANAGEBEAN | 68 |
| 4.2.5. | DIAGRAMA DE ATIVIDADES..... | 69 |
| 4.3. | PLANEJAMENTO DO PROJETO | 70 |
| 4.3.1. | WBS – WORK BREAKDOWN STRUCTURE | 70 |
| 4.3.2. | DEFINIÇÃO DAS ATIVIDADES..... | 71 |
| 4.3.3. | SEQUENCIAMENTO DAS ATIVIDADES DEFINIDAS..... | 71 |
| 4.3.4. | RECURSOS NECESSÁRIOS PARA O DESENVOLVIMENTO DO PROJETO | 72 |
| 4.3.5. | ESTIMATIVA DE CUSTOS..... | 72 |
| 4.3.6. | ORÇAMENTO DO PROJETO | 74 |
| 5. | CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS | 76 |
| | REFERÊNCIAS | 77 |
| | APÊNDICE A | 80 |

1. INTRODUÇÃO

Cada vez mais empresas aderem aos sistemas web. A web está deixando de ser usada somente para redes sociais e outras particularidades, e está começando a entrar na era do “cloud computing” (Computação em nuvem). Desta forma, várias empresas estão incorporando os sistemas web.

Assim diante de toda esta tecnologia, é comum as pessoas experimentarem a frustração de não conseguir utilizar os serviços de um caixa eletrônico, pois o mesmo estava fora do ar. Estes acontecimentos se devem à manutenção de alguma parte do sistema que, muitas vezes, não tem nenhuma relação com o serviço solicitado ou com a parte que é necessária para o uso que se deseja.

Com a popularidade dos sistemas web, e sua adesão tão frequente, torna-se inviável, e por muitas vezes ainda configura-se em prejuízos, o fato das empresas terem que ficar algumas horas ou até mesmo alguns dias sem seu sistema, para uma eventual atualização ou manutenção. Quando tal fato ocorre, tanto a empresa quanto o cliente acabam estressados, ocasionando, em muitos casos um quadro de perdas econômicas.

Na atualidade, a maioria dos sistemas web utiliza um login acoplado diretamente ao sistema, o que causa algumas dificuldades para implantação de novos módulos ou nas manutenções do sistema. Estas dificuldades ocorrem, geralmente, pela necessidade de ter que se retirar todo o sistema do ar para a implantação de novos módulos ou para pequenas manutenções com irrisórias alterações, como, por exemplo, novas regras exigidas pela lei ou pelo chefe da empresa. Outro incômodo encontrado pelos clientes é a personalização dos sistemas web, visto que, em todos os softwares web, os menus são fixos ou não oferecem a opção de alterar a cor ou a fonte das letras.

Em vários livros e apostilas de desenvolvimento, sempre se foi mostrado como programar os sistemas como um único módulo. Gonçalves (2007) mostra como desenvolver uma aplicação inteira para uma biblioteca, porém, o login fica acoplado

ao sistema. Ullman (2008), explica um modo de como criar um web site com o sistema de login, também inteiramente acoplado. Em praticamente todos os livros que se encontra hoje, o método de desenvolvimento é com o login acoplado ao sistema web.

O presente projeto apresenta a proposta de um login desacoplado do sistema web, no qual podem ser instalados novos módulos do sistema sem ter que se retirá-lo do ar. Outro acréscimo a ser obtido com este novo login desacoplado, é o fato de se poder remover um módulo do sistema sem afetar todo o resto do sistema web, favorecendo, com isso, as empresas que não precisarão parar quando for necessária a manutenção de seu sistema web. Junto com este login, será implantada uma interface dinâmica, onde o cliente poderá escolher a posição dos menus, tamanho da fonte das letras e ainda, as cores de fundo e das letras.

Desta forma, poderá ser realizada uma programação em módulos respectivos, como por exemplo, no sistema de um banco: a funcionalidade que gerencia o saldo ficará separada da funcionalidade de extratos a qual será independente da funcionalidade de abertura de contas e assim por diante. Com isso, se um cliente acessar o sistema do banco para ver seu saldo, e a funcionalidade de abertura de contas estiver em manutenção, este não precisará retornar mais tarde, já que a função do sistema que irá utilizar ainda estará disponível.

Com a geração deste módulo de login com interface dinâmica, a empresa de desenvolvimento conseguirá mais tempo para o desenvolvimento de novos sistemas web, pois não precisará se ocupar com a criação de um login para cada sistema. Outro acréscimo que a utilização deste módulo trará está no setor financeiro, visto que as empresas estarão economizando com a manutenção ou atualização isolada do sistema web no momento em que, somente uma pequena parte ficará fora do ar ao invés do sistema inteiro.

1.1. OBJETIVO

1.1.1. OBJETIVO GERAL

Como Objetivo geral deste projeto pretende-se obter um módulo de login desacoplado do sistema web em si, e com isso procura-se permitir a programação em módulos respectivos.

Com este módulo de login busca-se a implementação de uma interface dinâmica, que poderá ser adaptada para qualquer outra linguagem e igualmente para outras partes do sistema e outros sistemas, tanto web quanto desktop.

Desta forma, acredita-se que as empresas evitarão perdas financeiras e/ou transtornos com a manutenção dos sistemas web, favorecendo conjuntamente para a satisfação dos clientes.

1.1.2 OBJETIVOS ESPECÍFICOS

Visando atender o objetivo geral, os seguintes itens foram planejados:

- Levantar requisitos de segurança em sistemas web para garantir integridade e segurança dos dados manipulados, assim modelar o módulo proposto com os aspectos obtidos pela pesquisa.
- A partir dos resultados da pesquisa, foram desenvolvidos pequenos módulos para demonstração do módulo de login proposto.

1.2 JUSTIFICATIVA

Na atualidade, vários órgãos do governo e empresas particulares estão adotando sistemas web junto com a computação em nuvem (cloudcomputing).

Apesar do uso de tecnologias recentes com o html5, xhtml, Banco de dados NOSQL, computação distribuída entre outras, muitas instituições ainda encontram dificuldades na manutenção de seus sistemas. Um exemplo de dificuldade na

implantação e manutenção é o sistema do Ministério da Educação de Rio Pomba por FERNANDES et al. (2008).

Portanto, o projeto de criação deste módulo de login desacoplado justifica-se pela inovação de introduzir no mercado este produto, e pela contribuição na manutenção dos sistemas web, que poderá ser feita em etapas separadas. Desta forma, utilizando este novo login desacoplado, pode-se retirar do ar somente uma pequena parte de o sistema, parando somente um setor de cada vez ou até mesmo um único serviço do setor, diminuindo, com isso, o tempo em que o setor ou o serviço ficará indisponível.

1.3 PROBLEMA

A ideia para este projeto surgiu junto com um trabalho proposto em sala de aula, quando houve a necessidade de manutenção de um sistema já em funcionamento. A partir deste trabalho se observou a necessidade de retirar um sistema inteiro do ar para uma atualização de uma única página.

Para solucionar este problema, este projeto propõe a criação de um módulo de login onde todos os outros se conectam a ele, assim evitando que o sistema pare por completo por causa de pequenas atualizações ou manutenções.

1.4 ESTRUTURA DE DESENVOLVIMENTO

Este trabalho está estruturado em cinco capítulos:

O Capítulo 1, esta Introdução, onde são apresentados os objetivos e a justificativa para sua realização. Neste tópico é apresentado o tema do projeto a ser desenvolvido; o Capítulo 2 apresenta a revisão da literatura onde são apresentados outros trabalhos que apresentam tópicos importantes para o desenvolvimento deste projeto, onde serão mostrados vários tópicos para este projeto ser seguro e confiável. O capítulo 3, onde são descritas as ferramentas, tecnologias utilizadas no desenvolvimento do trabalho, além da metodologia de análise. O Capítulo 4

apresenta a documentação, junto com os diagramas do projeto. Por fim, no Capítulo 5, são apresentadas as considerações obtidas na elaboração do trabalho e o direcionamento para os trabalhos futuros.

2. REVISÃO DA LITERATURA

A informação é uma arma importante para qualquer empresa nos tempos modernos e também um recurso que pode alavancar ou falir a empresa. A segurança da informação é de vital importância, além de ser uma forma de gestão para muitas organizações. Porém, muitas organizações não prestam atenção no valor da informação até que as perdem. Ainda se encontra uma grande dificuldade por parte das empresas para se entender a importância da segurança da informação. Muitas empresas descartam a importância da segurança da informação apenas por acharem que não passa de um custo desnecessário para investir em sua empresa. Porém quando seus dados são roubados ou perdidos por falta de manutenção é que as empresas percebem o benefício do investimento que aparente não tem retorno, JUKEMURA (2006).

Spanceski (2004) diz que o maior desafio é de fazer o cliente entender que a segurança da informação não resulta em melhorias nas quais todos perceberam que algo foi feito, e sim que uma solução de segurança é geralmente imensurável.

O que ocorre sempre, é que ninguém nota a presença de um sistema de segurança, porém, sempre é notada a sua falta.

É preciso acentuar que o advento das novas tecnologias digitais, as quais contribuem para o processo da gestão da informação e o crescente fluxo da informação vem provocando mudanças radicais no âmbito das comunicações, principalmente com o surgimento da Internet, que possibilitou maior disseminação da informação. Nos meios de comunicação, houveram mudanças de paradigmas, tornando o fluxo de informação muito mais rápido e eficaz. Numa empresa jornalística, onde a informação é a principal matéria-prima, é necessário que seja aplicado os princípios da comunicação sob o aspecto da gestão da informação em meio digital, atuando de maneira diferenciada dos meios de comunicação tradicionais.

Com o crescimento da informação, muitas empresas estão aderindo aos sistemas web para poder ter maior controle sob a informação. Com esse crescimento de

sistemas, foi notado que quando necessária a manutenção do sistema, a empresa praticamente para toda a sua movimentação para poder efetuar a manutenção de seu sistema.

Segundo Ricardo Albuquerque e Bruno Ribeiro (2005), um software web é considerado seguro se ele tiver três aspectos importantes: Confidencialidade, integridade e disponibilidade. Porém, Kleiton Gomes e Newton Santos (2006) já comentam que, além destes três aspectos, existe outros tão importantes quanto estes, que são: autenticação, não repúdio, legalidade, privacidade, entre outros. Para fins de execução deste projeto se adotará a visão de Gomes, a qual expõe a importância de mais de três aspectos.

Assim, cada aspecto tem uma característica específica, como segue:

Confidencialidade: capacidade de um sistema de impedir que usuários não autorizados vejam determinada informação que foi delegada a somente usuários autorizados a vê-la.

Integridade: atributo de segurança que indica se uma informação pode ser alterada somente de forma autorizada.

Disponibilidade: indica a quantidade de vezes que o sistema cumpriu uma tarefa solicitada sem falhas internas, para um número de vezes em que foi solicitado a fazer a tarefa.

Autenticação: capacidade de garantir que um usuário, sistema ou informação é mesmo quem alega ser.

Não repúdio: capacidade de o sistema provar que um usuário executou determinada ação no sistema.

Legalidade: aderência de um sistema à legislação.

Privacidade: Capacidade de manter incógnito um usuário, impossibilitando a ligação direta da identidade do usuário com as ações por este realizadas.

Contudo, Fabrício Alves e André Alves (2009) comentam que os aspectos citados acima são importantes sim, porém as ferramentas de desenvolvimento são tão

importantes quanto os aspectos, já que sem elas não se têm garantias de que o dado é manipulado ou apresentado de forma coerente.

Além disso, existem algumas normas e padrões que regulam o desenvolvimento de sistemas web. Uma delas é o *ABNT NBR ISO/IEC 17.799*, que tem como objetivo assegurar a continuidade e minimizar o dano empresarial, prevenindo e minimizando o impacto de incidentes de segurança com a utilização de algoritmos de criptografia e de gerências de senhas para controle de acesso seguro dos dados disponibilizados.

Este módulo de login proposto seguirá os aspectos e normas citados acima, além de um levantamento minucioso das tecnologias empregadas para seu desenvolvimento, para que se possa garantir a segurança da informação que estará trafegando pela internet.

Assim pretende-se garantir um produto confiável e de qualidade para a empresa que o utiliza, e ainda, agradar o cliente que não precisará ficar preocupado com a localização do armazenamento dos seus dados.

3. TECNOLOGIAS, FERRAMENTAS E METODOLOGIAS UTILIZADAS.

No desenvolvimento deste projeto foi utilizada a linguagem Java por ser uma linguagem de programação totalmente orientada a objetos. Também foi utilizado o framework JSF (Java Server Faces) para desenvolver as telas do sistema, junto com o JSF, foi utilizado o Prime faces, que é um componente baseado em JSF. A ferramenta de desenvolvimento foi escolhido o NetBeans 7.2 por sua facilidade e popularidade com a linguagem.

O banco de dados foi o PostgreSQL, que foi escolhido por facilidade de integração ao Java e por desempenho estável quando utilizado em sistema web, para se trabalhar com o PostgreSQL foi escolhido o PgAdim 3, por ser uma ferramenta gráfica para utilização direta ao banco de dados e por ter uma interface gráfica simples. E por final, mas não menos importante a ferramenta utilizada para produção da UML e documentação do sistema foi o Astah Community.

3.1. TECNOLOGIAS E FERRAMENTAS

Aqui estão citadas todas as ferramentas utilizadas durante o desenvolvimento do projeto.

3.1.1. NETBEANS

O NetBeans é um projeto open source com grande aceitação pelos usuários, o projeto foi fundado pela Sun Microsystems em junho de 2000. É possível baixar o NetBeans em seu site, netbeans.org, onde também se encontra a comunidade de desenvolvedores do projeto, o objetivo da comunidade é aperfeiçoar o NetBeans, além de permitir aos utilizadores de mais de cento e sessenta países a terem acesso sem o menor problema de compatibilidade com o sistema operacional, no site do NetBeans é possível fazer o download das últimas versões da plataforma, além de

poder utilizar a documentação disponível para melhor utilizá-lo, é possível utilizar os fóruns disponíveis no site para obter ajuda em relação a dúvidas sobre o NetBeans ou até mesmo para tirar dúvidas de projetos próprios.

O NetBeans, é uma ferramenta de desenvolvimento que permite ser utilizada em várias linguagens de desenvolvimento. Ele possui um editor multilinguagem além de um Debugger (Depurador, programa para verificação de erros). Por ele ser multilinguagem é possível desenvolver vários projetos em linguagens diferentes, sem nenhum conflito ou atraso no desenvolvimento, assim podendo desenvolver uma aplicação Java ou uma aplicação em PHP. As linguagens aceitas por ele são: Java, HTML, PHP, C/C++ e Groovy.

Dentro do próprio NetBeans, é possível gerenciar banco de dados, servidores web para teste além de rastreadores de problemas nos projetos em criação.

É possível obter mais informações sobre o NetBeans no site www.netbeans.org.

3.1.2. JAVA

Java é uma linguagem de programação 100% orientada a objetos, que foi lançada em 1995 pela *Sun Microsystems*, porém segundo o DEITEL (2003), a criação do Java se iniciou em 1991 inicialmente com o nome de Green. Esta linguagem é base para praticamente todo tipo de aplicativos de rede.

O foco inicial era criar uma tecnologia de desenvolvimento de software para aparelhos domésticos (Atualmente podemos ver Java em vários eletrodomésticos, sendo também base para a TV digital brasileira).

Seu criador, James Gosling, o batizou com o nome de Oak, que era uma árvore que ele sempre observava, porém, depois de muito estudo, foi descoberto que já existia uma linguagem chamada de Oak.

Segundo Pacievitch, em 2006, o Java já tinha sido quase todo convertido para a licença de Software livre, criando assim uma comunidade de desenvolvimento global. Sua utilização é relativamente simples, por isso a quantidade de

programadores que o adotam aumenta a cada dia (PACIEVITCH, 2011).

Com a popularização da internet, a equipe de desenvolvedores, começou a ter a ideia de implantá-lo na internet. Em seu lançamento em 1995, foi liberada a primeira versão do java, a qual já suportava desenvolvimento para a internet.

Com sua enorme aceitação, grandes empresas como a IBM, anunciaram que ajudariam no suporte ao Java. Com isso, as empresas fariam seus produtos aceitarem aplicativos feitos em Java. Segundo estudos, Java foi a tecnologia adotada mais rapidamente na história da informática.

Esta linguagem passou a ser utilizada em vários dispositivos como os computadores, palmtops, celulares e em grande parte dos produtos da Apple.

A linguagem Java oferece mecanismos de encapsulamento, abstração e hereditariedade; é independente de plataforma, possui uma máquina virtual a JVM (Java Virtual Machine), que transforma o código do programa em bytecodes, linguagem própria da JVM; é uma linguagem compilada e interpretada não vinculada com ambiente de desenvolvimento, a sua restrição é apenas ter o JDK que é composto pelo compilador, máquina virtual e as bibliotecas de desenvolvimento instalado no computador (JUNIOR, 1999).

3.1.3. POSTGRESQL

O PostgreSQL é o resultado de um projeto que se iniciou com o nome de Projeto Ingres, desenvolvido pela Universidade de Berkeley na Califórnia.

O líder do Projeto, Michael Stonebraker, foi um dos pioneiros em Banco de Dados Relacional, em 1982 ele deixou a universidade para comercializar o Ingres, porém não obteve grande sucesso, causando o seu retorno para a universidade.

Com o seu retorno, a DARPA (Defense Advanced Research Projects agency, Agência de pesquisas de Projetos Avançados) e ARO (Army Reserch Office, Escritório de Pesquisa Exército) patrocinaram o pós-Ingres, com o objetivo de resolver problemas encontrados no Banco de Dados Relacional.

Um dos principais problemas encontrados era a incapacidade de se trabalhar com tipos (atualmente conhecidos com “objetos”). O projeto final foi chamado de Postgres, que incluía a funcionalidade de incluir tipos.

Em agosto de 1996 o projeto foi renomeado para PostgreSQL, para representar a nova linguagem de consulta aos dados. Ainda em 1996, foi lançada a primeira versão fora da universidade. A primeira versão do PostgreSQL foi a 6.0, e desde sua liberação, desenvolvedores voluntários do mundo todo tem desenvolvido novas funcionalidades até os dias atuais.

Hoje o PostgreSQL é um projeto open source, que é coordenado pelo PostgreSQL Global Development Group, a versão mais atual é a 9.3.3, onde se encontra centenas de melhorias desde a primeira versão liberada.

Atualmente o projeto do PostgreSQL possui compatibilidade com diversos Sistemas Operacionais, tais quais: GNU/Linux, Unix (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64), e MS Windows.

O PostgreSQL é conhecido pelo fácil suporte e integração a chaves estrangeiras, junções, visões, gatilhos e procedimentos armazenados. Nele, faz-se presente grande quantidade de tipos de dados do ISO SQL:1999, como INTEGER, NUMERIC, BOOLEAN, CHAR, VARCHAR, DATA, INTERVAL, e TIMESTAMP. Além de admitir a alocação de objetos binários, abrangendo figuras, sons ou vídeos (Sobre PostgreSQL, 2013).

3.1.4. PGADMIN 3

O software PgAdmin é o mais popular software Open Source para administração do Banco de Dados PostgreSQL. Ele pode ser usado tanto no Linux como no Windows, serve para administrar versões do PostgreSQL a partir da 7.3.

Ele é projetado para atender às necessidades de todos os usuários, desde escrever consultas SQL simples, até para o desenvolvimento de bancos de dados complexos. A interface gráfica suporta todas as funcionalidades do PostgreSQL e torna a administração mais fácil. O aplicativo também inclui um editor com destaque de

sintaxe SQL, um editor de código já criado, um SQL agente de agendamento de trabalho, o apoio a mecanismo de replicação e muito mais. Conexão com o servidor pode ser feita utilizando soquetes de domínio Unix TCP / IP, ou pode ser criptografado SSL para a segurança.

É desenvolvido por uma comunidade de especialistas do PostgreSQL em todo o mundo e está disponível em vários idiomas. O PgAdmin é um projeto de software livre liberado sob a **PostgreSQL/Artistic Licence** (Disponível em <http://www.pgadmin.org/licence.php>). O software está disponível no site em formato binário.

PgAdmin vem na instalação padrão na maioria das distribuições Linux, como Red Hat, Fedora, Debian e Ubuntu. Se necessário uma versão diferente do incluído na distribuição, poderá ser feita uma atualização pelos repositórios de cada versão do Linux.

3.1.5. JAVA SERVER FACES (JSF)

Na atualidade, existem muitos frameworks para desenvolvimento de interfaces gráficas para software web. O JSF é um framework baseado em componente que permite a criação de interface. Por exemplo, se quiséssemos exibir uma tabela com linhas e colunas, não seria necessário gerar tags de HTML para linhas e células como em um loop, basta adicionar um componente de tabela na página. Podemos dizer que JSF é baseado nas seguintes regras:

- uma série de componentes pré-fabricados para criação de UIs (*User Interfaces*).
- um modelo de programação baseado em eventos
- um modelo de componentes que permite qualquer usuário desenvolver seus próprios elementos adicionais.

JSF não é o único framework web baseado em componentes, mas é a camada de visão no padrão Java EE. JSF está incluído em cada servidor de aplicativos Java EE, e pode ser facilmente adicionado a um web container como o *Tomcat*.

3.1.6. PRIMEFACES

Para atender a demanda dos desenvolvedores por componentes mais sofisticados, há várias extensões do JSF que seguem o mesmo ciclo e modelo da especificação. Exemplos dessas bibliotecas são **PrimeFaces**, **RichFaces** e **IceFaces**. Todas elas definem componentes JSF que vão muito além da especificação.

Cada biblioteca oferece *ShowCases* na web para mostrar seus componentes e suas funcionalidades. Onde se pode ver o *showcase* do **PrimeFaces** no endereço <http://www.primefaces.org>.

No seu demo *online*, se pode ver uma lista de componentes disponíveis, como inputs, painéis, botões diversos, menus, gráficos e componentes *drag & drop*, que vão muito além das especificações, ainda mantendo a facilidade de uso:

3.1.7. ASTAH COMMUNITY

O Astah Community é o sucessor do Jude Community, que se tornou o editor gratuito mais utilizado pela sua grandiosidade de recursos oferecidos e por sua praticidade em elaborar diagramas completos.

3.2. METODOLOGIAS DE ANÁLISE

A análise orientada a objetos surgiu na década de 1990. Após algum tempo surge a UML (Unified Modeling Language, Unified Modeling Language), aprovada como padrão de pelo OMG (*Object Management Group*), que é uma organização internacional que aprova padrões abertos para aplicações orientadas a objetos (BEZERRA, 2002).

“A UML é uma linguagem visual para modelar sistemas orientados a objetos. Isso quer dizer que a UML é uma linguagem constituída de elementos gráficos (visuais)

utilizados na modelagem que permitem representar os conceitos do paradigma da orientação objetos” (BEZERRA, 2002, p. 14).

3.2.1. CASOS DE USO

O caso de uso é utilizado para facilitar a comunicação entre o analista de sistema e o cliente, mostrando melhor o cenário do software a ser desenvolvido, junto com algumas funcionalidades a partir da visão do usuário.

3.2.2. DIAGRAMA DE CLASSES

O diagrama de classes é definido para exibir a estrutura do software, demonstrando o conceito de classes desenvolvidas no software. Classe é a representação de um objeto, tendo os seus atributos e métodos.

3.2.3. DIAGRAMA DE ATIVIDADES

No diagrama de atividades, são demonstradas algumas atividades do software antes ou depois de alguma ação.

3.2.4. SEQUENCIAMENTO DE ATIVIDADES

O sequenciamento de atividades tem por objetivo identificar e documentar os relacionamentos lógicos nas atividades presentes no cronograma, para que os prazos sejam atingidos.

4. DOCUMENTAÇÃO

Na documentação é onde se encontra todos os dados, pesquisa e diagramas feitos para este projeto.

4.1. LEVANTAMENTO DOS REQUISITOS

Aqui está toda a especificação necessária para o desenvolvimento deste projeto.

4.1.1. FORMA ADOTADA

Uma das partes mais importantes no desenvolvimento de um software é o levantamento de requisitos. Onde se tem o primeiro contato com o cliente para se falar do software a ser desenvolvido. Neste ponto se obtêm as especificações e exigências do cliente para a criação do software. Um bom levantamento dos requisitos facilita em muito o desenvolvimento.

Abaixo segue o questionário.

Qual o foco principal da empresa?

R:

Qual o objetivo do software que será desenvolvido?

R:

Existe algum software similar ao que será desenvolvido?

R:

Se sim, qual a dificuldade em se trabalhar com ele hoje?

R:

Qual o resultado esperado do software a ser Desenvolvido?

R:

4.1.2. ANÁLISE DOS REQUISITOS

Na análise dos requisitos, é citados todos os itens necessários para o desenvolvimento do projeto, além das restrições do mesmo.

4.1.2.1. PROBLEMAS POTENCIAIS

Cadastrar usuários;

Cadastrar grupos

Cadastrar usuários nos grupos

Cadastrar permissões para usuários/grupos;

Cadastrar Sistemas;

Cadastrar Eventos;

Manter a segurança de cada evento cadastrado;

Manter o acesso aos eventos restritos ao sistema;

Visualizar eventos disponíveis para cada usuário;

Visualizar eventos favoritos;

Visualizar/Alterar perfil de cada usuário;

Visualizar mensagens de cada usuário;

4.1.2.2. RESTRIÇÕES DO SISTEMA

Sempre necessitará de acesso à internet por ser um sistema web, porém não será necessário muitos recursos de hardware, já que o sistema será desenvolvido com compatibilidade para quase todos os navegadores disponíveis.

4.1.3. VALIDAÇÃO DOS REQUISITOS

Faltou algum requisito?

Sim () Não ()

As informações foram compreendidas corretamente?

Sim () Não ()

Os objetivos estão de acordo?

Sim () Não ()

Os dados levantados atendem as expectativas?

Sim () Não ()

O desenvolvimento foi satisfatório?

Sim () Não ()

Observações sobre dúvidas, sugestões ou reclamações:

Informações adicionais:

Responsável pela Validação

4.1.4. ESPECIFICAÇÃO DOS REQUISITOS

O software a ser desenvolvido controlará e gerenciará o acesso de usuários aos eventos disponíveis dentro da empresa, esses eventos podem ser apenas simples consultas e relatórios, até movimentações e controle financeiro.

O responsável pela administração poderá criar níveis de acesso diferentes para

grupos/usuários, melhorando a segurança da empresa.

4.1.4.1. ESCOPO

Os usuários, grupos, sistemas, eventos e permissões de acesso devem ser cadastrados pelo administrador do sistema.

O administrador poderá dar permissão de acesso individualmente para cada usuário, ou se preferir, poderá criar um grupo para usuários e dar permissão de acesso ao grupo inteiro.

As permissões de acesso serão divididas em quatro níveis, que serão ler (somente a visualização do evento), inserir (poderá gravar informações no evento), alterar (poderá alterar informações do evento) e excluir (permitirá remover informações do evento).

Será permitido a cada usuário a marcação de seus eventos favoritos. Esses eventos ficarão disponíveis em uma lista de eventos na tela principal de acesso para facilitar sua utilização.

Cada usuário terá acesso para alterar informações em seu perfil. Essas alterações serão restringidas a alteração de telefone, endereço, e-mail, login ou senha de acesso.

4.1.4.2. LISTA DE EVENTOS

Manter Usuário

Manter Grupo

Manter evento

Manter sistema

Manter permissões

Manter favoritos

Manter perfil do usuário

Manter mensagens

Visualizar eventos por usuário

Visualizar eventos favoritos do usuário

Visualizar mensagens do usuário

Emitir relatórios de usuário

Emitir relatórios de grupos

Emitir relatórios de sistemas

Emitir relatórios de eventos

4.2. ANÁLISE ORIENTADA A OBJETOS

Na análise orientada a objetos, estão todos os diagramas e especificações necessários para o desenvolvimento do projeto.

4.2.1. ESPECIFICAÇÃO DOS CASOS DE USOS E DIAGRAMAS DE SEQUENCIA

Estão reunidos neste tópico, todos os diagramas de caso de uso junto com os diagramas de sequencia de cada parte do projeto desenvolvido.

4.2.1.1. EFETUAR O LOGIN

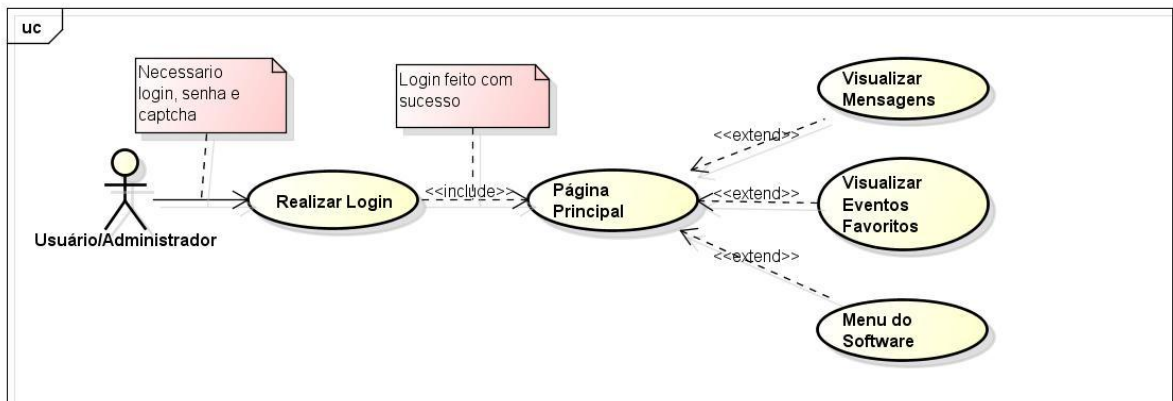


Figura 1 - Caso de Uso: Efetuar o Login

Finalidade/Objetivo:

Permitir que o usuário/administrador acesse ao software

Atores:

Usuário e Administrador.

Evento Inicial:

O usuário fornece seu nome de usuário, senha e captcha.

Fluxo Principal:

O usuário deve informar todos os dados necessários na tela de login;

Será feita a autenticação dos dados informados pelo usuário/administrador na base de dados do software;

Apos autenticação o usuário/administrador será redirecionado para a página principal do sistema, onde terá acesso aos eventos favoritos, menu do software e as mensagens enviadas para ele.

Fluxos Alternativos:

Se for um novo usuário é necessário que o administrador faça seu cadastro;

Caso o usuário/administrador esqueça a senha, é possível utilizar a ajuda disponível

na tela de login para alteração da mesma.

Fluxos de Exceção:

Caso seja informado dados inválidos no login, o software pedirá os dados novamente informando uma mensagem de dados inválidos.

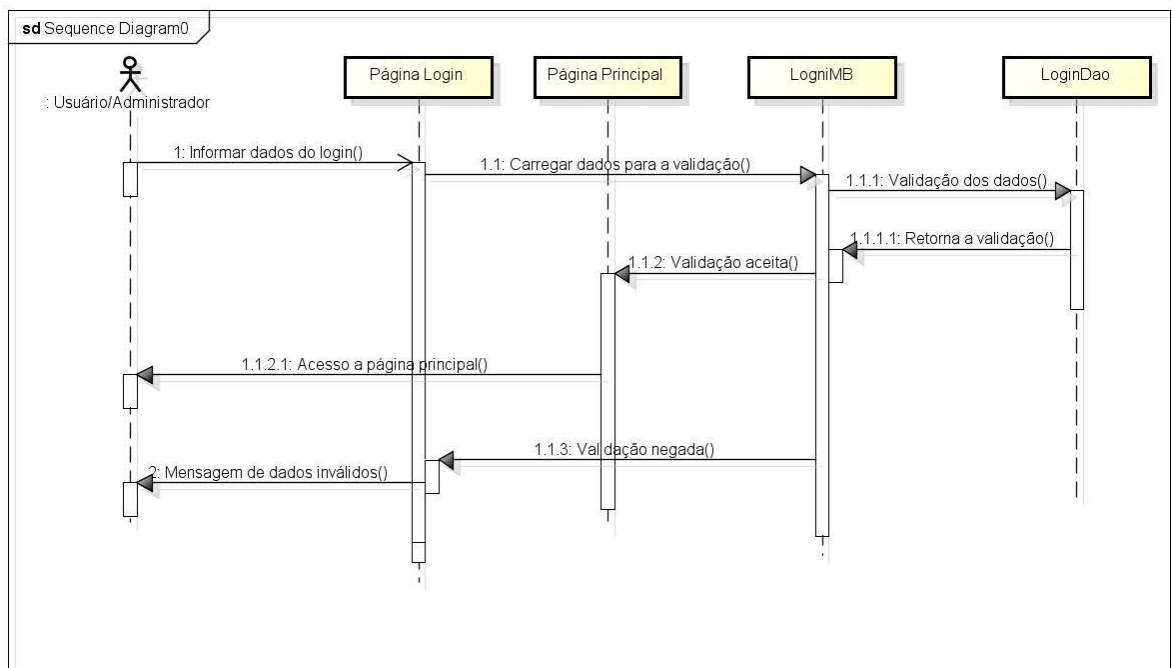
Pós-Condições:

O usuário/administrador tem acesso total a sua página principal, podendo realizar suas tarefas normalmente.

Casos de Testes:

Validar os dados do usuário e verificar se o mesmo não se encontra inativo/bloqueado no software.

4.2.1.2. DIAGRAMA DE SEQUENCIA: REALIZAR LOGIN



powered by Astah

Figura 2 - Diagrama de Sequencia: Efetuar o Login

4.2.1.3. VISUALIZAR MENSAGENS

Finalidade/Objetivo:

Permitir o acesso às mensagens enviadas ao usuário/administrador. Essas mensagens serão enviadas a partir de um modulo externo que será acrescentado posteriormente para demonstração.

Atores:

Usuário e administrador.

Evento Inicial:

Listar todas as mensagens não lida pelo usuário/administrador

Fluxo Principal:

O usuário/administrador seleciona a mensagem a ser lida.

Visualização do corpo da mensagem e de qual sistema foi enviada.

Fechamento da mensagem selecionada, alterando o estado dela para lida.

Pós-Condições:

O usuário/administrador efetuar tarefas pedidas pelo sistema.

Casos de Testes:

Mostrar somente mensagens não lidas.

4.2.1.4. VISUALIZAR EVENTOS FAVORITOS

Finalidade/Objetivo:

Permitir o acesso rápido pelo usuário/administrador aos eventos de maior utilização/necessidade.

Atores:

Usuário e Administrador.

Pré-condições:

Cadastrar os eventos favoritos na página de favoritos.

Fluxo Principal:

O usuário/administrador seleciona o evento para utilizá-lo/consultar dependendo da necessidade ou permissão atribuídas ao usuário (Administrador tem acesso total a todos os eventos cadastrados).

Fluxos Alternativos:

O usuário pode solicitar ao administrador para cadastrar níveis de permissão acima do disponível a ele naquele momento.

Fluxos de Exceção:

O evento não será mais exibido caso as permissões seja removida daquele usuário.

Casos de Testes:

Validação das permissões que o usuário tem para cada evento cadastrado em seus favoritos.

4.2.1.5. MENU DO SOFTWARE

Finalidade/Objetivo:

Permitir acesso a todas as áreas disponíveis ao usuário/administrador.

Atores:

Usuário e Administrador.

Pré-condições:

Ter feito o login.

Evento Inicial:

Exibir opções disponíveis no software.

Fluxo Principal:

Conduzir o usuário/administrador pelo software.

4.2.1.6. FAVORITOS

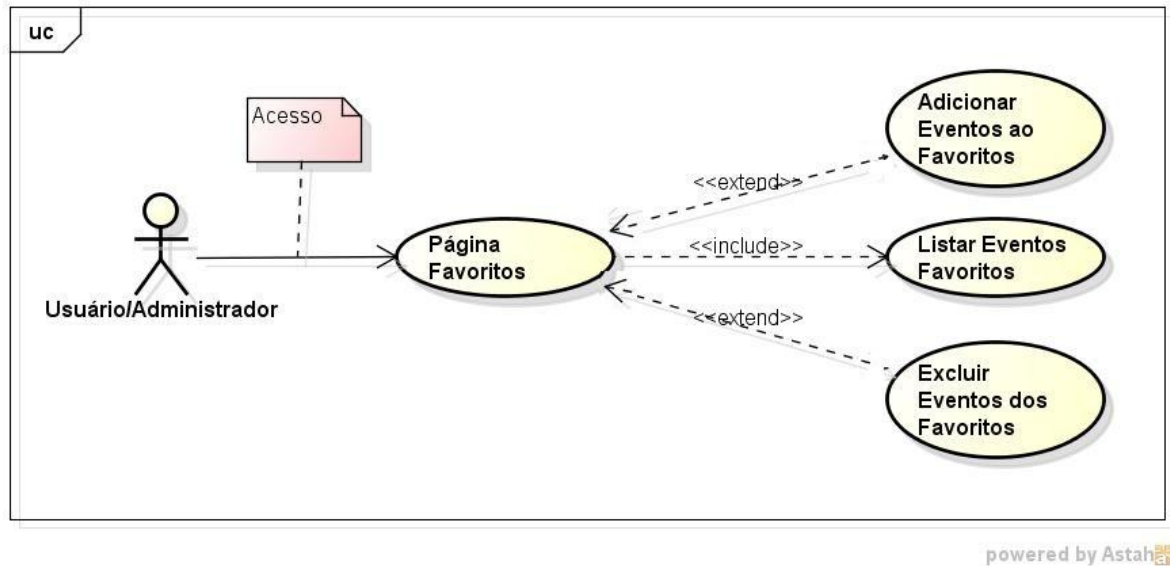


Figura 3- Caso de Uso: Favoritos

Finalidade/Objetivo:

Permitir que o usuário/administrador gere os eventos favoritos que ele utiliza.

Atores:

Usuário e Administrador.

Pré-condições:

É necessário fazer a autenticação no login.

Evento Inicial:

Exibição de todos os eventos disponíveis para o usuário/administrador, e todos os que já estão cadastrados como favoritos.

Fluxo Principal:

O usuário/administrador deve fazer a autenticação no login e em seguida selecionar o botão de favoritos no menu principal;

Fluxos Alternativos:

Visualizar a lista de favoritos na tela principal do software

Pós-Condições:

O usuário/administrador pode cadastrar eventos a lista de favoritos.

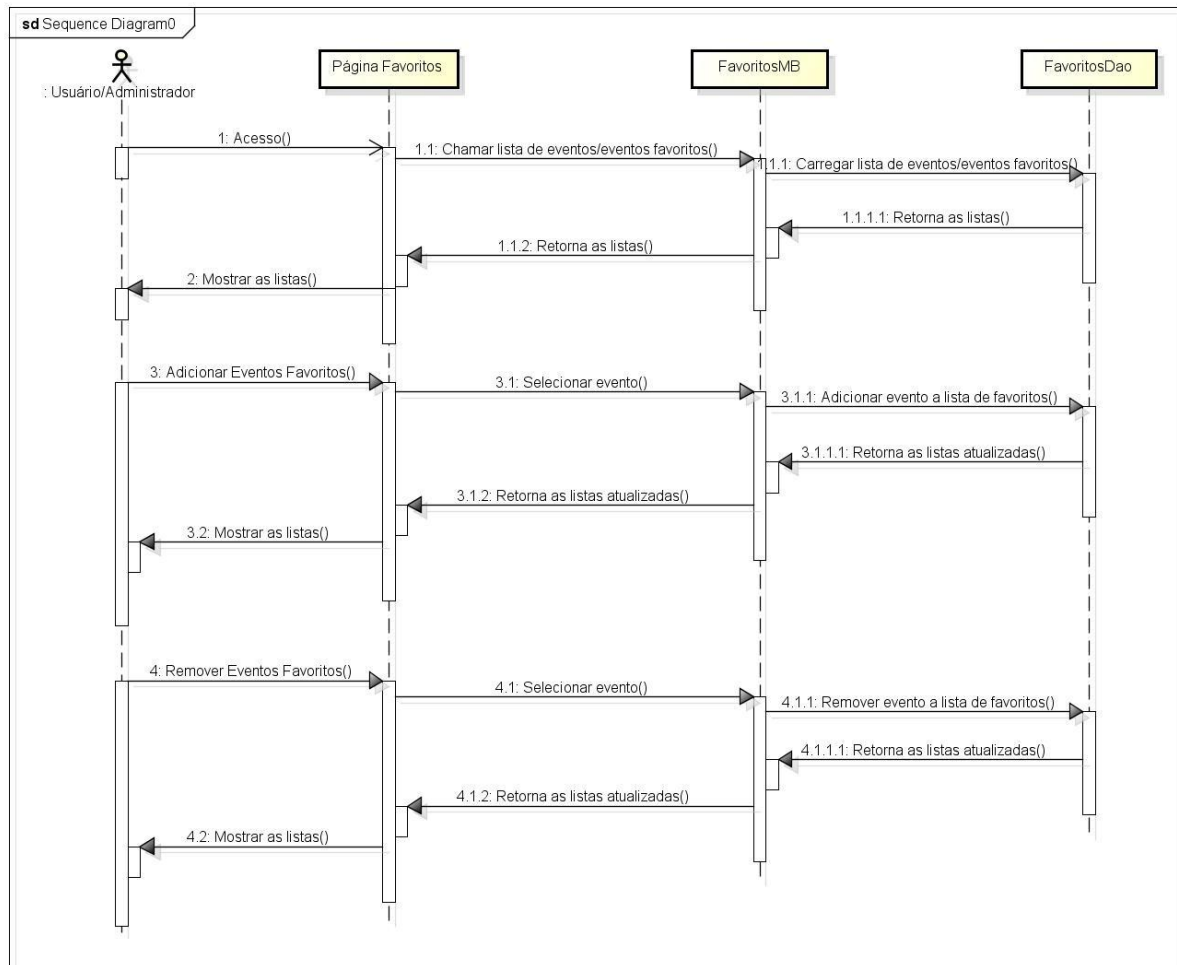
Fluxos de Exceção:

O evento será eliminado das duas listas caso a permissão seja retirada do usuário.

Casos de Testes:

Teste de permissões dos eventos do usuário.

4.2.1.7. DIAGRAMA DE SEQUENCIA: FAVORITOS



powered by Astah

Figura 4- Diagrama de Sequencia: Favoritos

4.2.1.8. ADICIONAR EVENTOS AOS FAVORITOS

Finalidade/Objetivo:

Permitir adicionar eventos de uso frequente a sua lista de favoritos que é mostrada na página principal.

Atores:

Usuário/Administrador

Pré-condições:

É necessário fazer a autenticação no login e acessar a página de favoritos.

Evento Inicial:

Acessar a página de favoritos no menu principal

Fluxo Principal:

Selecionar o evento da lista de eventos disponíveis. Em seguida o evento será mostrado na lista de eventos favoritos, onde as alterações podem ser confirmadas.

Pós-Condições:

Será atualizada a lista de favoritos da página principal.

Fluxos de Exceção:

O evento será eliminado das duas listas caso a permissão seja retirada do usuário.

Casos de Testes:

Teste de permissões dos eventos do usuário.

4.2.1.9. LISTAR EVENTOS FAVORITOS

Finalidade/Objetivo:

Listar todos os eventos disponíveis para cadastrar nos favoritos e também listar todos os eventos favoritos.

Atores:

Usuário/Administrador

Pré-condições:

É necessário fazer a autenticação no login e acessar a página de favoritos.

Evento Inicial:

Acessar a página de favoritos no menu principal

Fluxos de Exceção:

O evento será eliminado das duas listas caso a permissão seja retirada do usuário.

Casos de Testes:

Teste de permissões dos eventos do usuário.

4.2.1.10. REMOVER EVENTOS AOS FAVORITOS

Finalidade/Objetivo:

Permitir remover eventos da lista de favoritos que é mostrada na página principal.

Atores:

Usuário/Administrador

Pré-condições:

É necessário fazer a autenticação no login e acessar a página de favoritos.

Evento Inicial:

Acessar a página de favoritos no menu principal

Fluxo Principal:

Selecionar o evento da lista de eventos favoritos. Em seguida o evento será mostrado na lista de eventos disponíveis.

Pós-Condições:

Será atualizada a lista de favoritos da página principal.

Fluxos de Exceção:

O evento será eliminado das duas listas caso a permissão seja retirada do usuário.

Casos de Testes:

Teste de permissões dos eventos do usuário.

4.2.1.11. EVENTOS

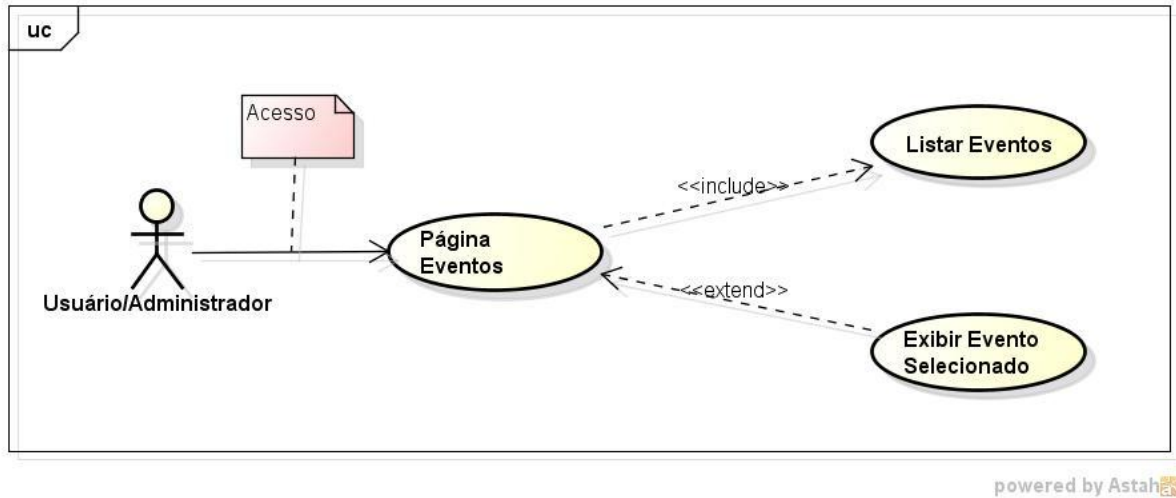


Figura 5- Caso de Uso: Eventos

Finalidade/Objetivo:

Mostrar todos os eventos disponíveis para o usuário/administrador de acordo com as permissões do usuário

Atores:

Usuário/Administrador

Pré-condições:

É necessário fazer a autenticação no login e acessar a página de favoritos.

Evento Inicial:

Acessar a página de eventos no menu principal

Fluxo Principal:

O usuário/administrador deve fazer a autenticação no login e em seguida selecionar o botão de eventos no menu principal;

Pós-Condições:

Visualização do evento selecionado.

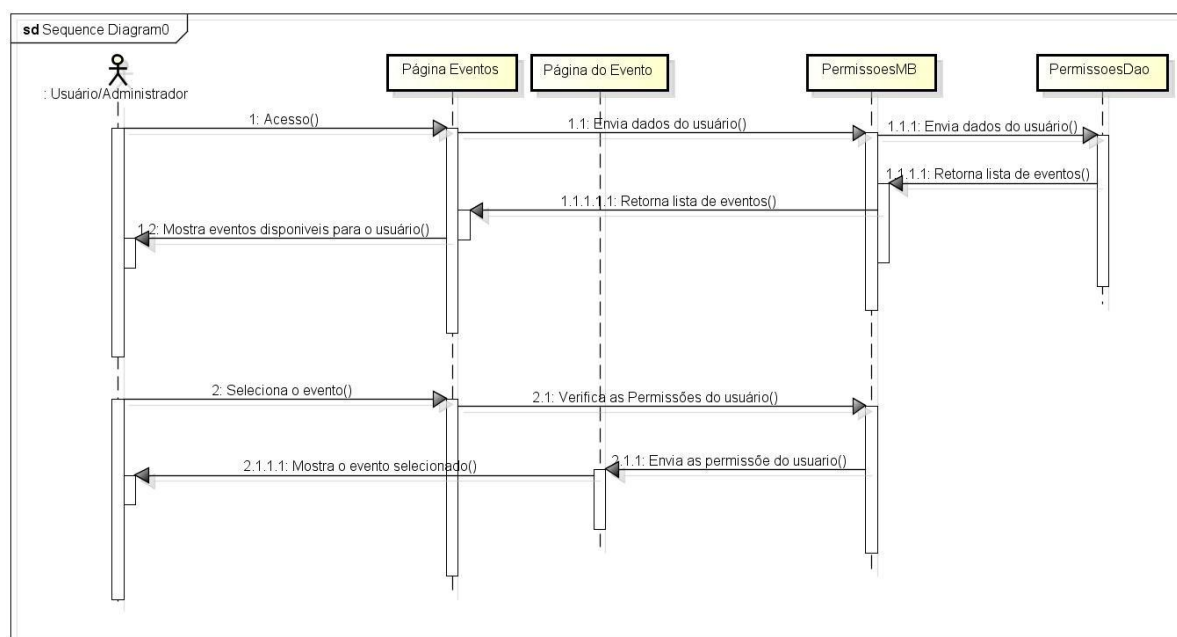
Fluxos de Exceção:

O evento será eliminado da lista caso a permissão seja retirada do usuário.

Casos de Testes:

Teste de permissões dos eventos do usuário.

4.2.1.12. DIAGRAMA DE SEQUENCIA: EVENTO



powered by Astah

Figura 6- Diagrama de Sequencia: Evento

4.2.1.13. LISTAR EVENTOS

Finalidade/Objetivo:

Mostrar todos os eventos disponíveis para o usuário/administrador de acordo com as permissões do usuário

Atores:

Usuário e Administrador

Pré-condições:

É necessário fazer a autenticação no login e acessar a página de eventos.

Evento Inicial:

Acessar a página de eventos no menu principal

Fluxo Principal:

O usuário/administrador deve fazer a autenticação no login e em seguida selecionar o botão de eventos no menu principal;

Pós-Condições:

Visualização do evento selecionado.

Fluxos de Exceção:

O evento será eliminado da lista caso a permissão seja retirada do usuário.

Casos de Testes:

Teste de permissões dos eventos do usuário.

4.2.1.14. EXIBIR EVENTO SELECIONADO

Finalidade/Objetivo:

Exibir o evento que o usuário/administrador selecionou com todas as permissões de acesso disponível para aquele usuário.

Atores:

Usuário e Administrador

Pré-condições:

É necessário fazer a autenticação no login, acessar a página de eventos e selecionar um evento disponível.

Evento Inicial:

Selecionar evento na página de eventos.

Fluxo Principal:

O usuário/administrador deve fazer a autenticação no login, em seguida selecionar o botão de eventos no menu principal e selecionar o evento desejado;

Fluxos de Exceção:

O evento será eliminado da lista caso a permissão seja retirada do usuário.

Casos de Testes:

Teste de permissões dos eventos do usuário.

4.2.1.15. PERFIL

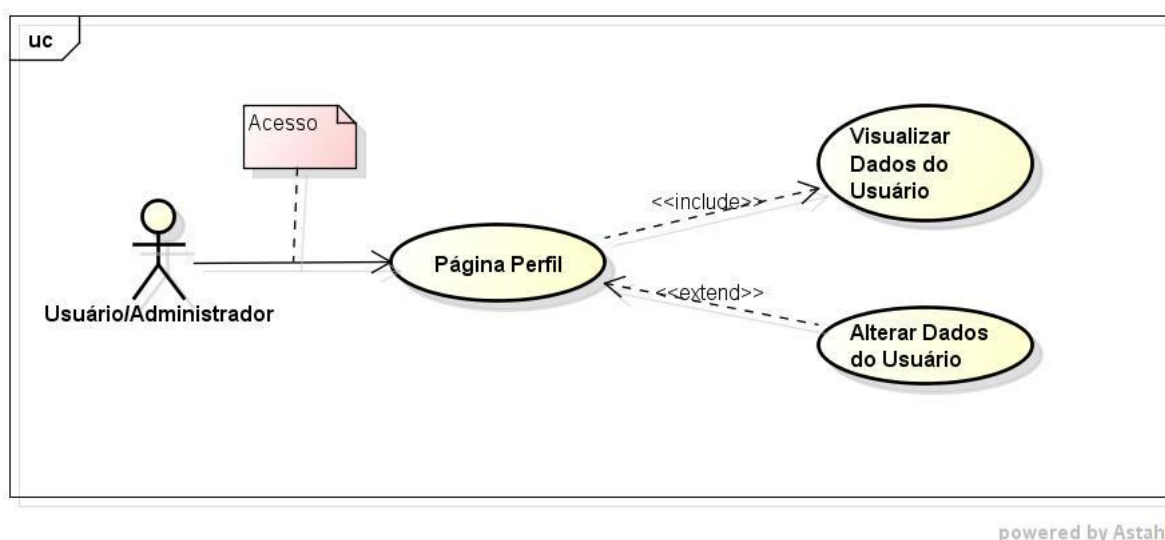


Figura 7- Caso de Uso: Perfil

Finalidade/Objetivo:

Permitir a visualização dos dados cadastrais do usuário/administrador

Atores:

Usuários e Administrador.

Pré-condições:

É necessário fazer a autenticação no login e acessar a página de perfil.

Evento Inicial:

Acessar a página de perfil no menu principal

Fluxo Principal:

O usuário/administrador deve fazer a autenticação no login, em seguida selecionar o botão de perfil no menu principal.

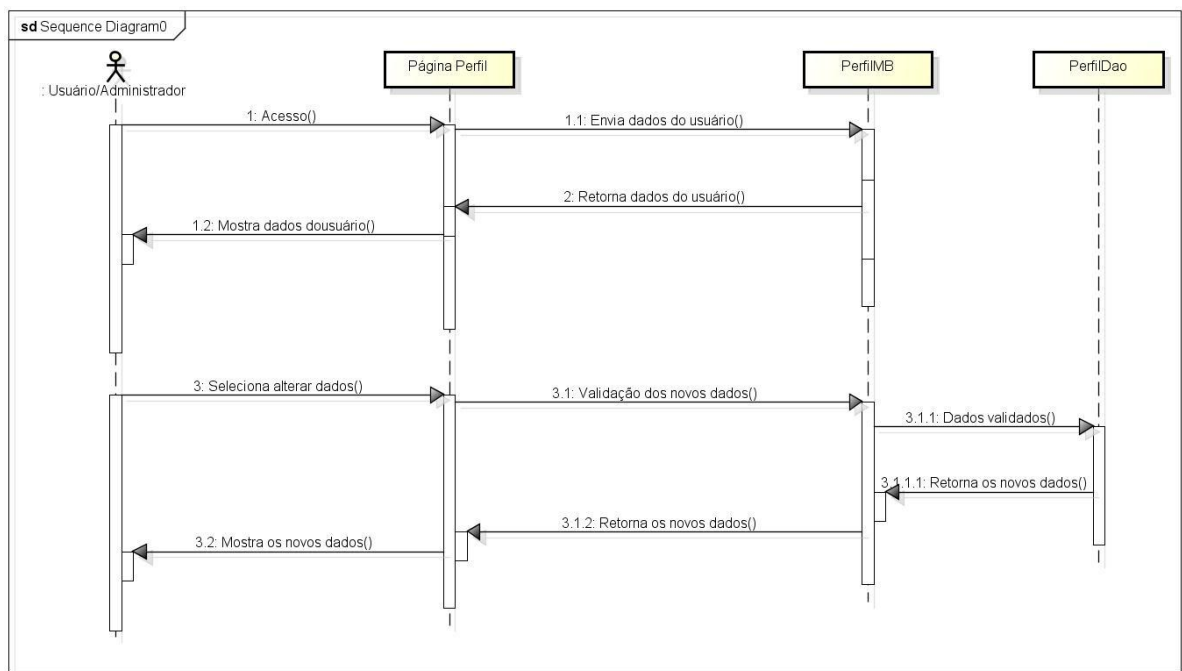
Pós-Condições:

Carregar informações do usuário guardadas na sessão do software.

Fluxos de Exceção:

Validar usuário na sessão logo após o login ser efetuado.

4.2.1.16. DIAGRAMA DE SEQUENCIA: PERFIL



powered by Astah

Figura 8- Diagrama de Sequencia: Perfil

4.2.1.17. ALTERAR DADOS DO USUÁRIO

Finalidade/Objetivo:

Permitir alteração dados cadastrais do usuário/administrador

Atores:

Usuários e Administrador.

Pré-condições:

É necessário fazer a autenticação no login, acessar a página de perfil e selecionar o botão de alteração.

Evento Inicial:

Acessar a página de perfil no menu principal

Fluxo Principal:

O usuário/administrador deve fazer a autenticação no login, em seguida selecionar o botão de perfil no menu principal, logo após selecionar o botão alterar para que seja feita a liberação dos campos para poder editar as informações, após isto, basta salvar as alterações.

Pós-Condições:

Atualização dos dados cadastrais

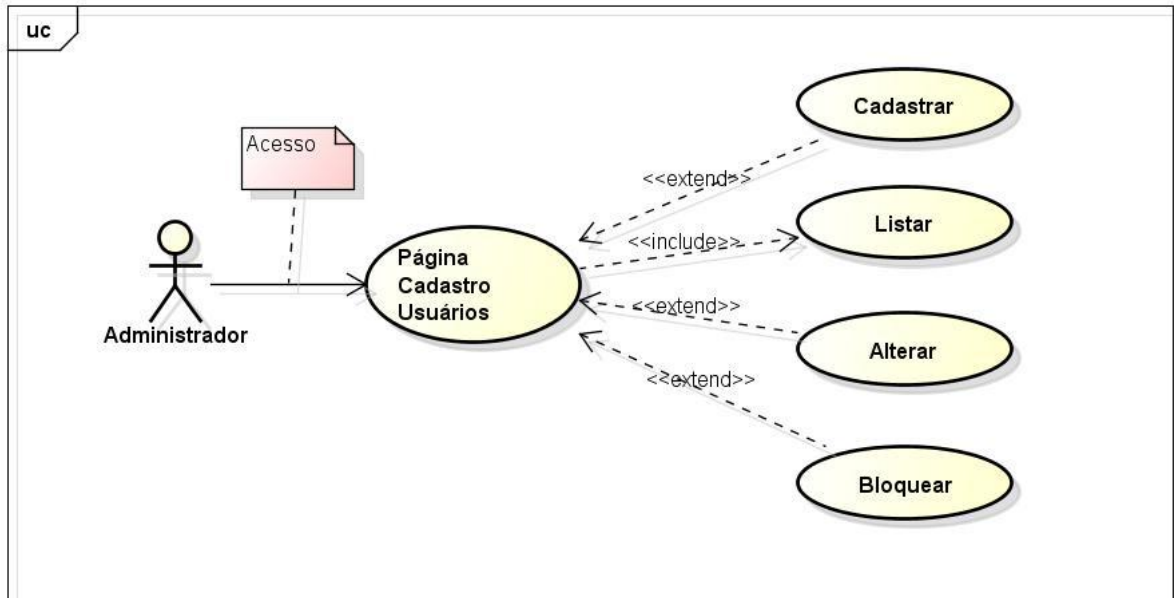
Fluxos de Exceção:

Caso os dados não passem pela validação, a atualização não será permitida.

Casos de Testes:

Validar usuário na sessão logo após o login ser efetuado, e validação dos novos dados inseridos.

4.2.1.18. CADASTRAR USUÁRIO



powered by Astah

Figura 9- Caso de Uso: Cadastrar Usuário

Finalidade/Objetivo:

Permitir ao administrador do software visualizar/alterar/cadastrar/bloquear usuários

Atores:

Administrador

Pré-condições:

É necessário fazer a autenticação no login, acessar a página de administrador e selecionar o botão cadastro de usuários.

Evento Inicial:

Visualização da lista de todos os usuários existentes no software

Fluxo Principal:

Visualização da lista de todos os usuários existentes no software

Pós-Condições:

Atualização da lista de usuários do software

Casos de Testes:

Validação dos dados novos/alterados

4.2.1.19. DIAGRAMA DE SEQUENCIA: CADASTRAR USUÁRIO

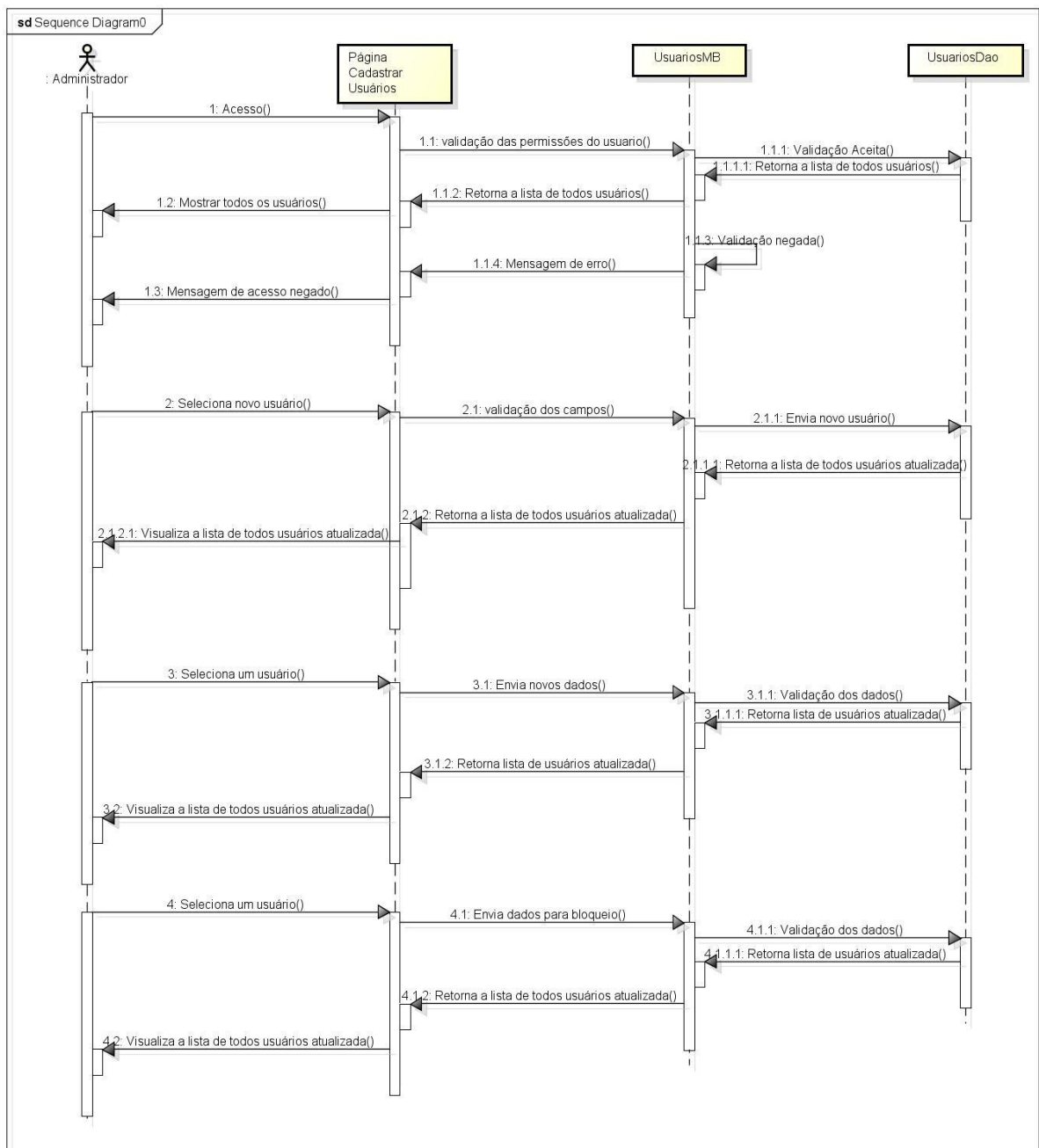


Figura 10- Diagrama de Sequencia: Cadastrar Usuário

4.2.1.20. CADASTRAR GRUPO

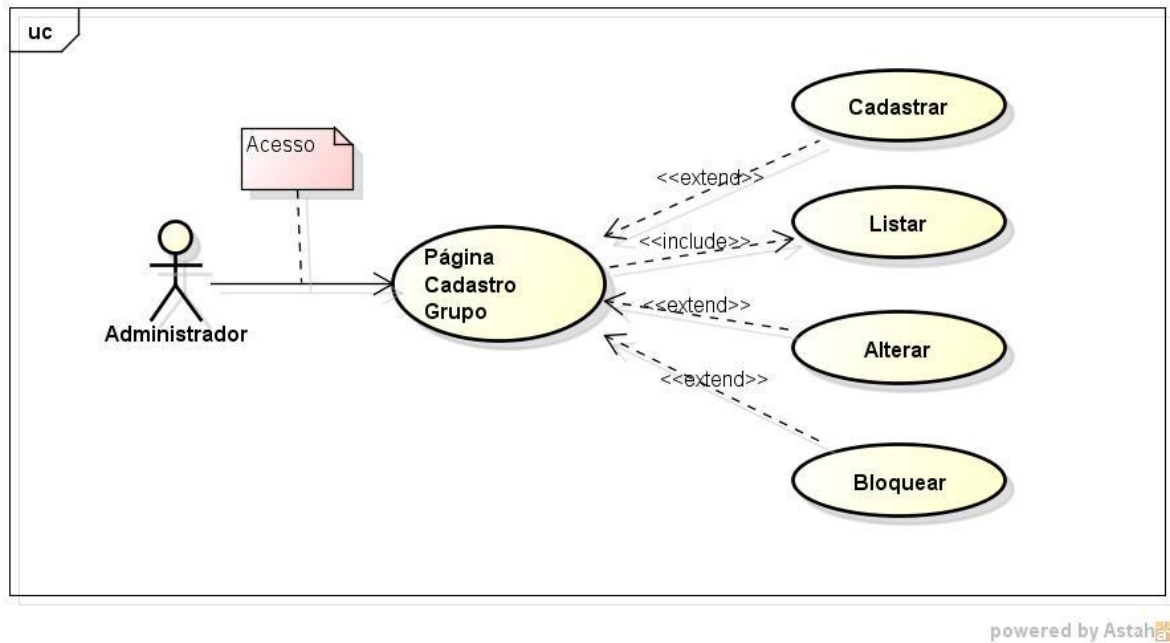


Figura 11- Caso de Uso: Cadastrar Grupo

Finalidade/Objetivo:

Permitir ao administrador do software visualizar/alterar/cadastrar/bloquear grupos

Atores:

Administrador

Pré-condições:

É necessário fazer a autenticação no login, acessar a página de administrador e selecionar o botão cadastro de grupos.

Evento Inicial:

Visualização da lista de todos os grupos existentes no software

Fluxo Principal:

Visualização da lista de todos os grupos existentes no software

Pós-Condições:

Atualização da lista de grupos do software

Casos de Testes:

Validação dos dados novos/alterados

4.2.1.21. DIAGRAMA DE SEQUENCIA: CADASTRAR GRUPO

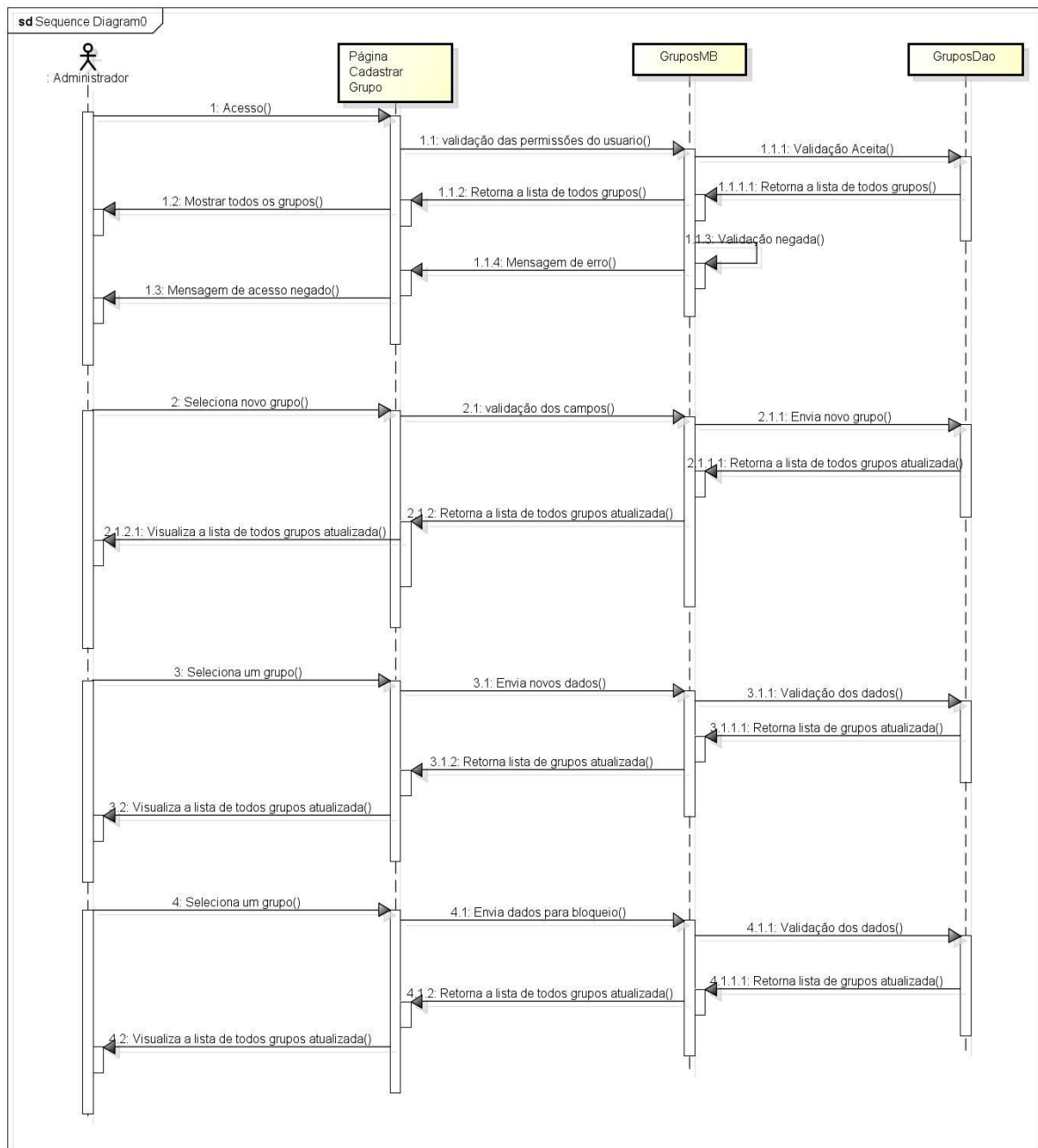
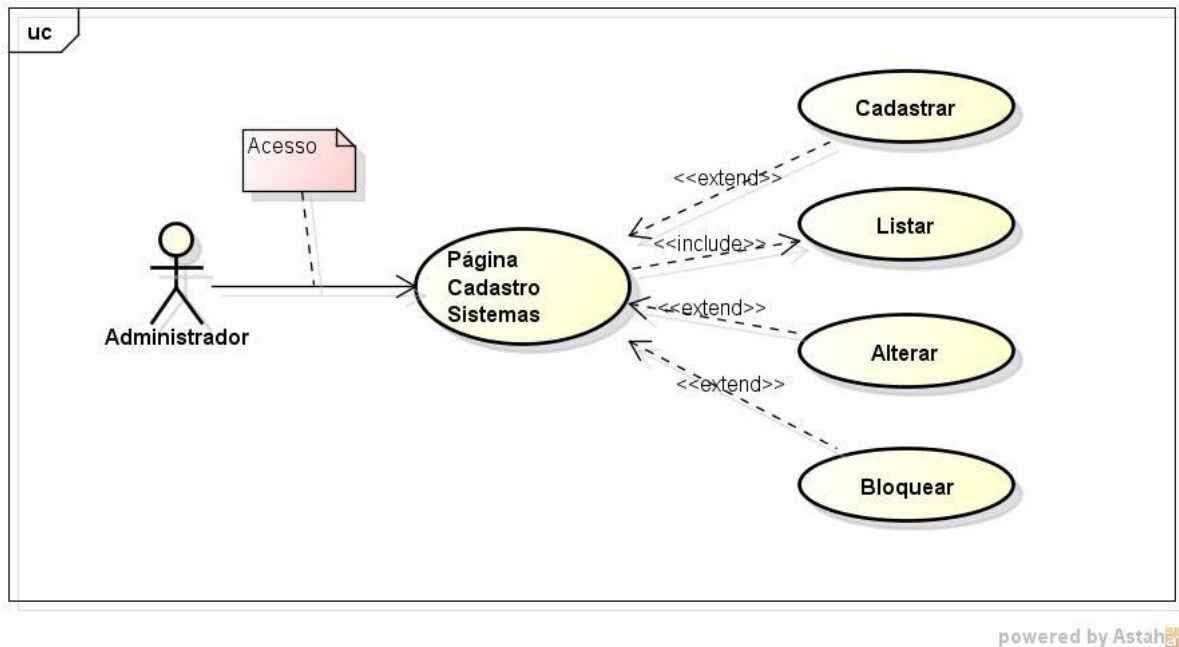


Figura 12- Diagrama de Sequencia: Cadastrar Grupo

4.2.1.22. CADASTRAR SISTEMAS



powered by Astah

Figura 13- Caso de Uso: Cadastrar Sistema

Finalidade/Objetivo:

Permitir ao administrador do software visualizar/alterar/cadastrar/bloquear sistemas

Atores:

Administrador

Pré-condições:

É necessário fazer a autenticação no login, acessar a página de administrador e selecionar o botão cadastro de sistemas.

Evento Inicial:

Visualização da lista de todos os sistemas existentes no software

Fluxo Principal:

Visualização da lista de todos os sistemas existentes no software

Pós-Condições:

Atualização da lista de sistemas do software

Casos de Testes:

Validação dos dados novos/alterados

4.2.1.23. DIAGRAMA DE SEQUENCIA: CADASTRAR SISTEMAS

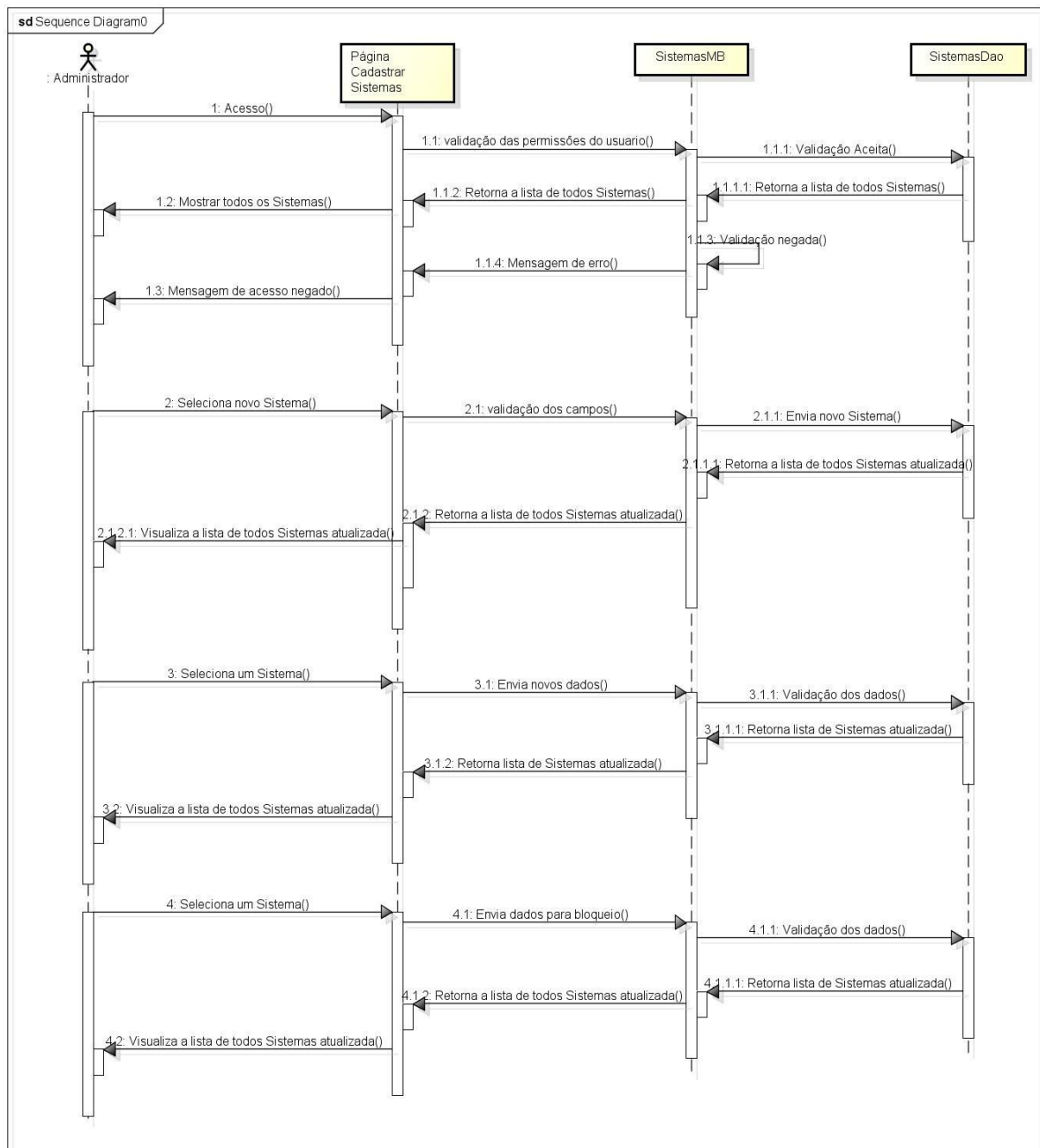
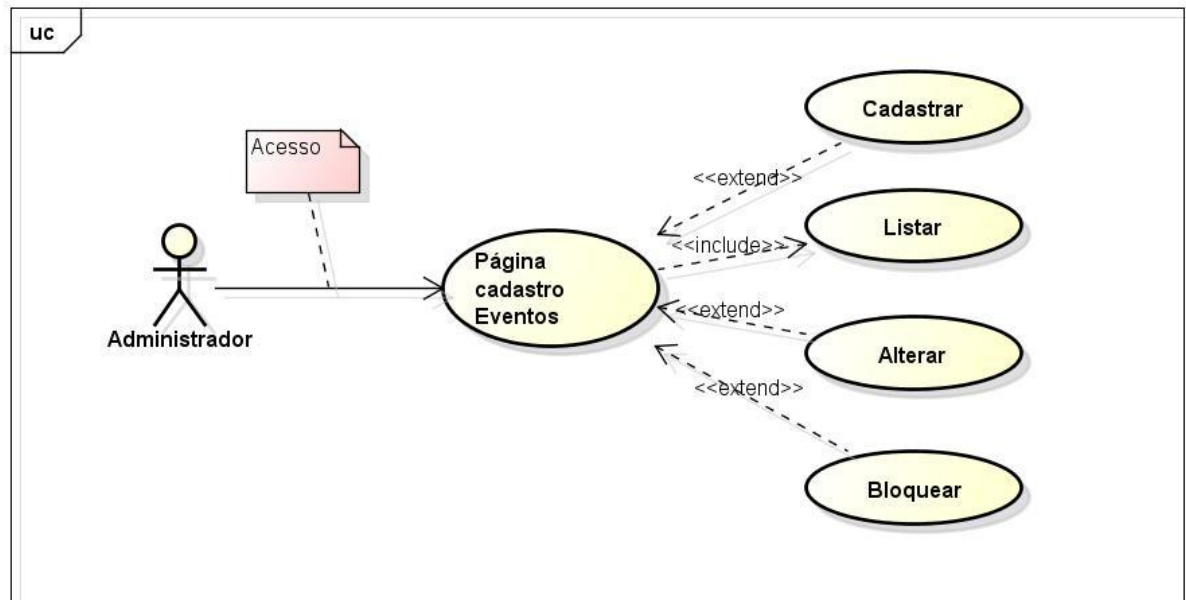


Figura 14- Diagrama de Sequencia: Cadastrar Sistemas

4.2.1.24. CADASTRAR EVENTO



powered by Astah

Figura 15- Caso de Uso: Cadastrar Eventos

Finalidade/Objetivo:

Permitir ao administrador do software visualizar/alterar/cadastrar/bloquear eventos

Atores:

Administrador

Pré-condições:

É necessário fazer a autenticação no login, acessar a página de administrador e selecionar o botão cadastro de eventos.

Evento Inicial:

Visualização da lista de todos os eventos existentes no software

Fluxo Principal:

Visualização da lista de todos os eventos existentes no software

Pós-Condições:

Atualização da lista de eventos do software

Casos de Testes:

Validação dos dados novos/alterados

4.2.1.25. DIAGRAMA DE SEQUENCIA: CADASTRAR EVENTOS

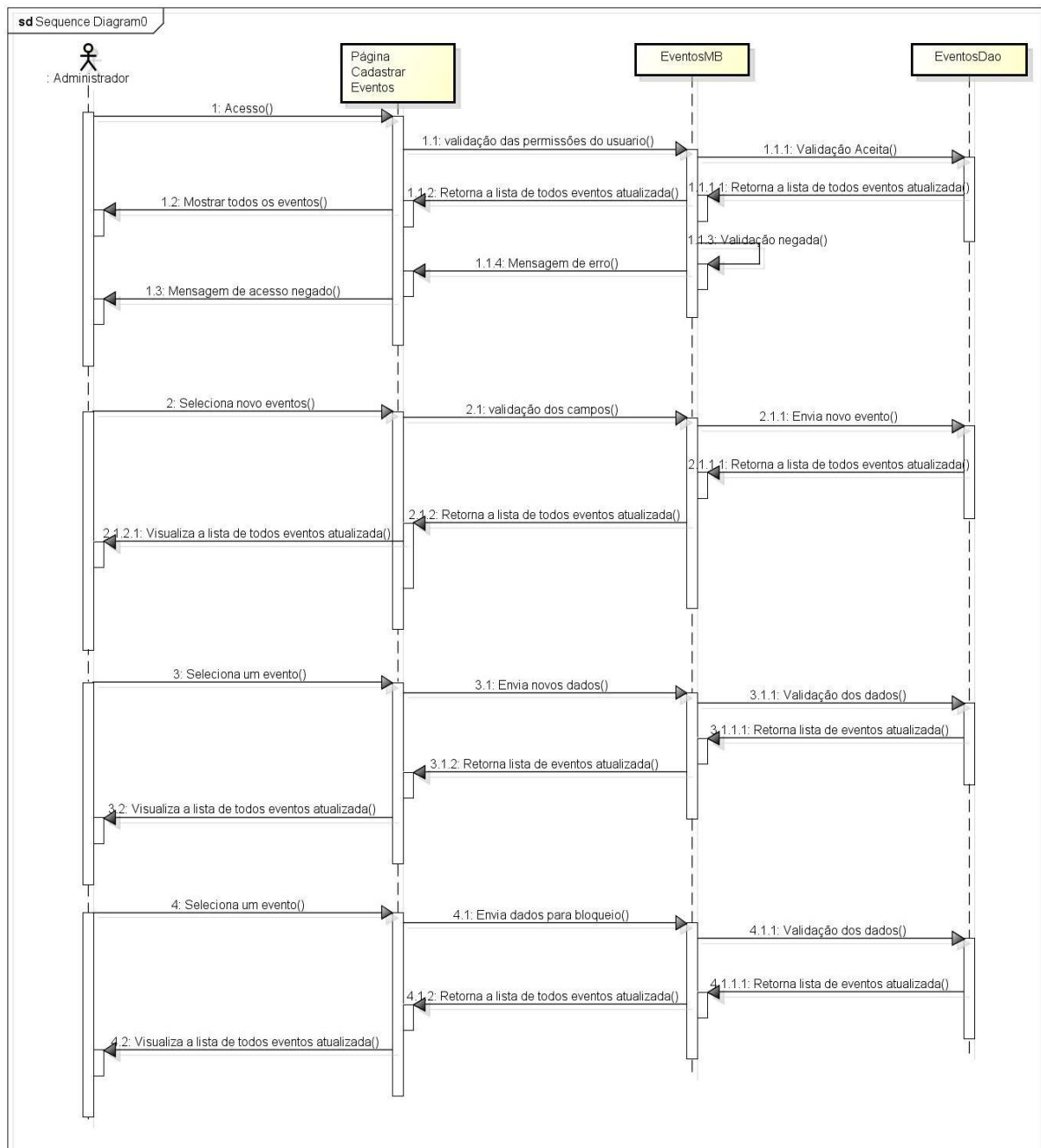


Figura 16- Diagrama de Sequencia: Cadastrar Eventos

4.2.1.26. CADASTRAR PERMISSÕES

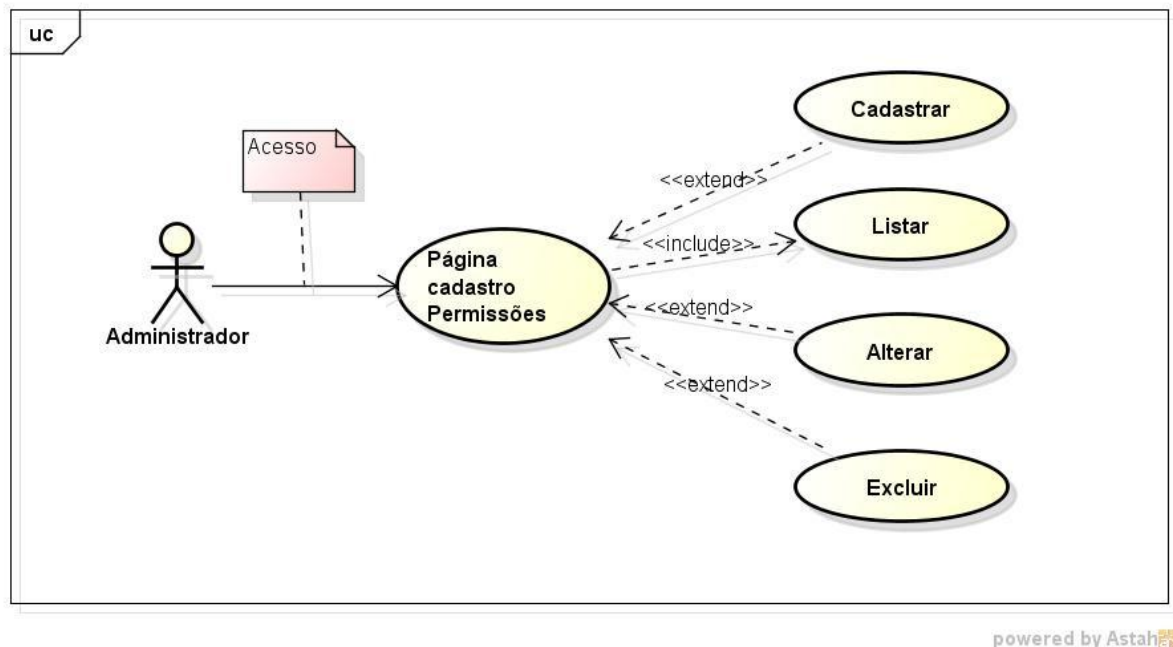


Figura 17- Caso de Uso: Cadastrar Permissões

Finalidade/Objetivo:

Permitir ao administrador do software visualizar/alterar/cadastrar/excluir permissões para usuários/grupos

Atores:

Administrador

Pré-condições:

É necessário fazer a autenticação no login, acessar a página de administrador e selecionar o botão cadastro de permissões.

Evento Inicial:

Visualização da lista de todas as permissões existentes no software

Fluxo Principal:

Visualização da lista de todas as permissões existentes no software

Pós-Condições:

Atualização da lista de permissões do software

Casos de Testes:

Validação dos dados novos/alterados/excluídos e para ver se o evento que pertence aquela permissão está disponível

4.2.1.27. DIAGRAMA DE SEQUENCIA: CADASTRAR PERMISSÕES

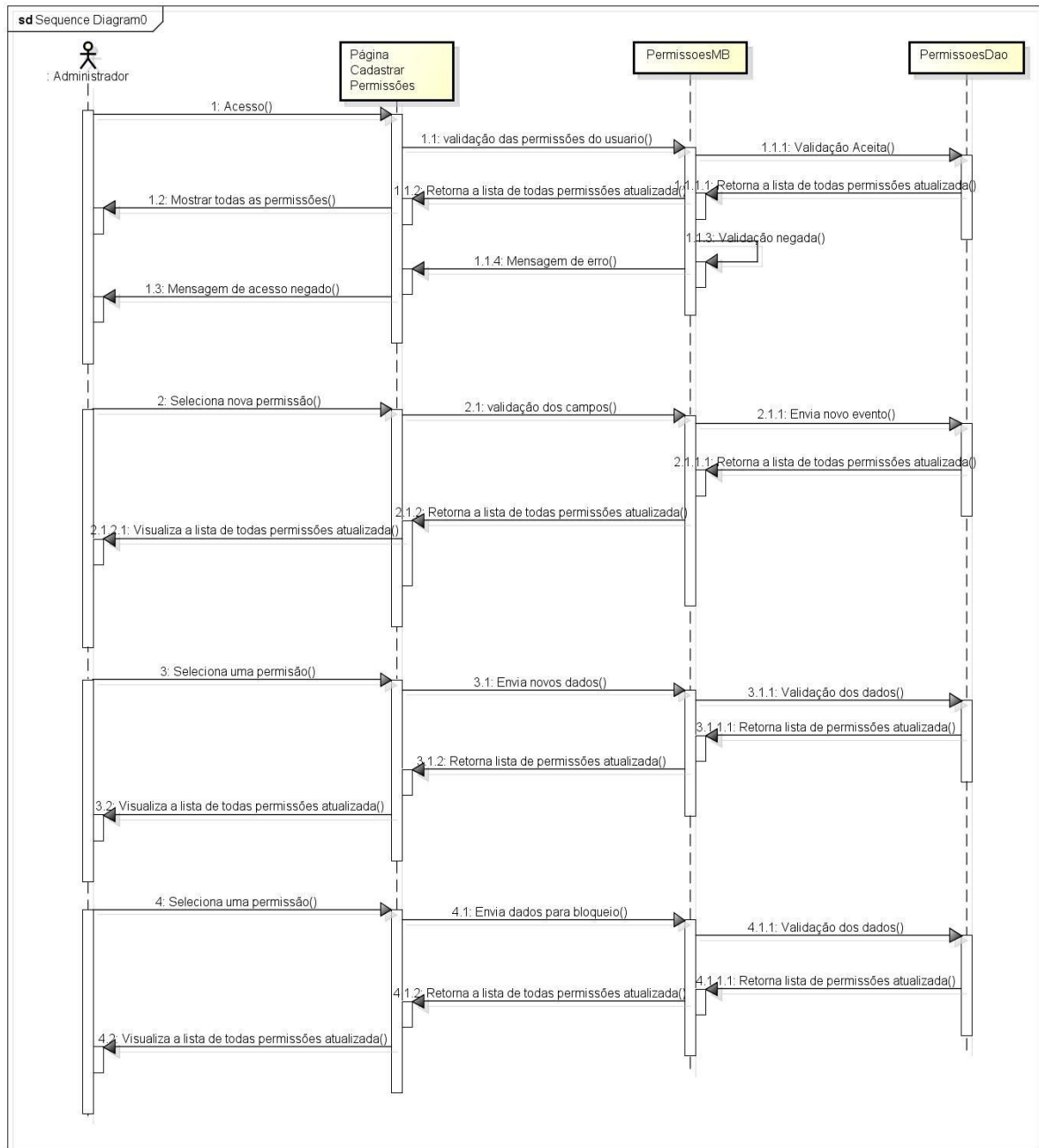


Figura 18- Diagrama de Sequencia: Cadastrar Permissões

4.2.1.28. CADASTRAR USUÁRIOS/GRUPOS

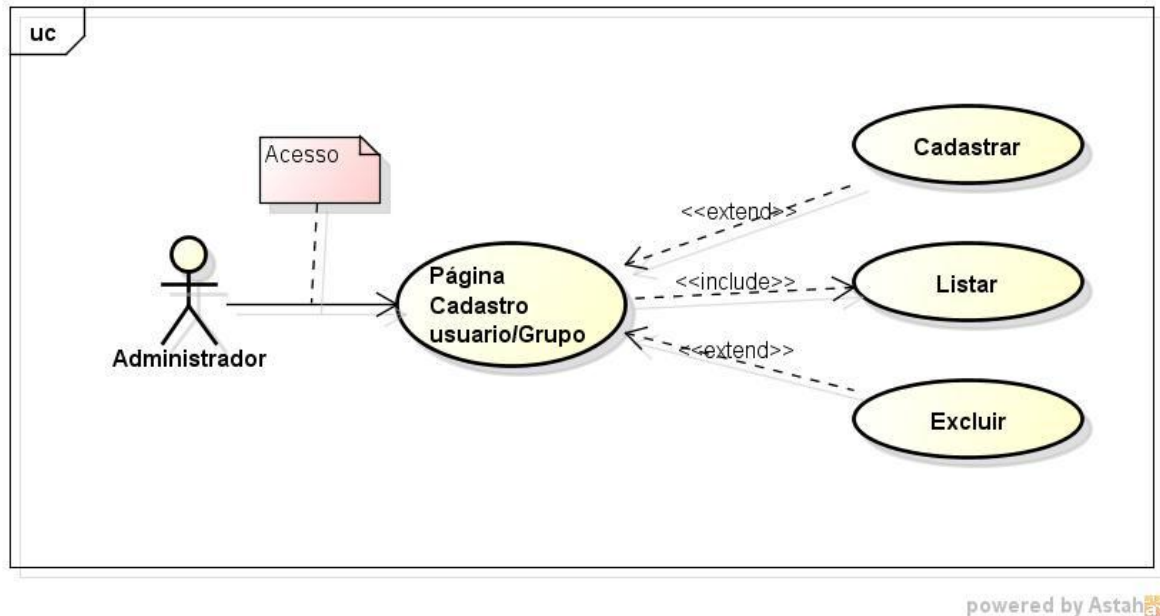


Figura 19- Caso de Uso: Cadastrar Usuários/Grupos

Finalidade/Objetivo:

Permitir ao administrador do software visualizar/cadastrar/excluir Usuários nos grupos

Atores:

Administrador

Pré-condições:

É necessário fazer a autenticação no login, acessar a página de administrador e selecionar o botão cadastro de Usuários/Grupos

Evento Inicial:

Visualização da lista de todos os grupos com seus respectivos usuários existentes no software

Fluxo Principal:

Visualização da lista de todos os grupos com seus respectivos usuários existentes no software

Pós-Condições:

Atualização da lista de Usuários/Grupos do software

Casos de Testes:

Validação dos dados novos/excluídos

4.2.1.29. DIAGRAMA DE SEQUENCIA: CADASTRAR USUÁRIOS/GRUPOS

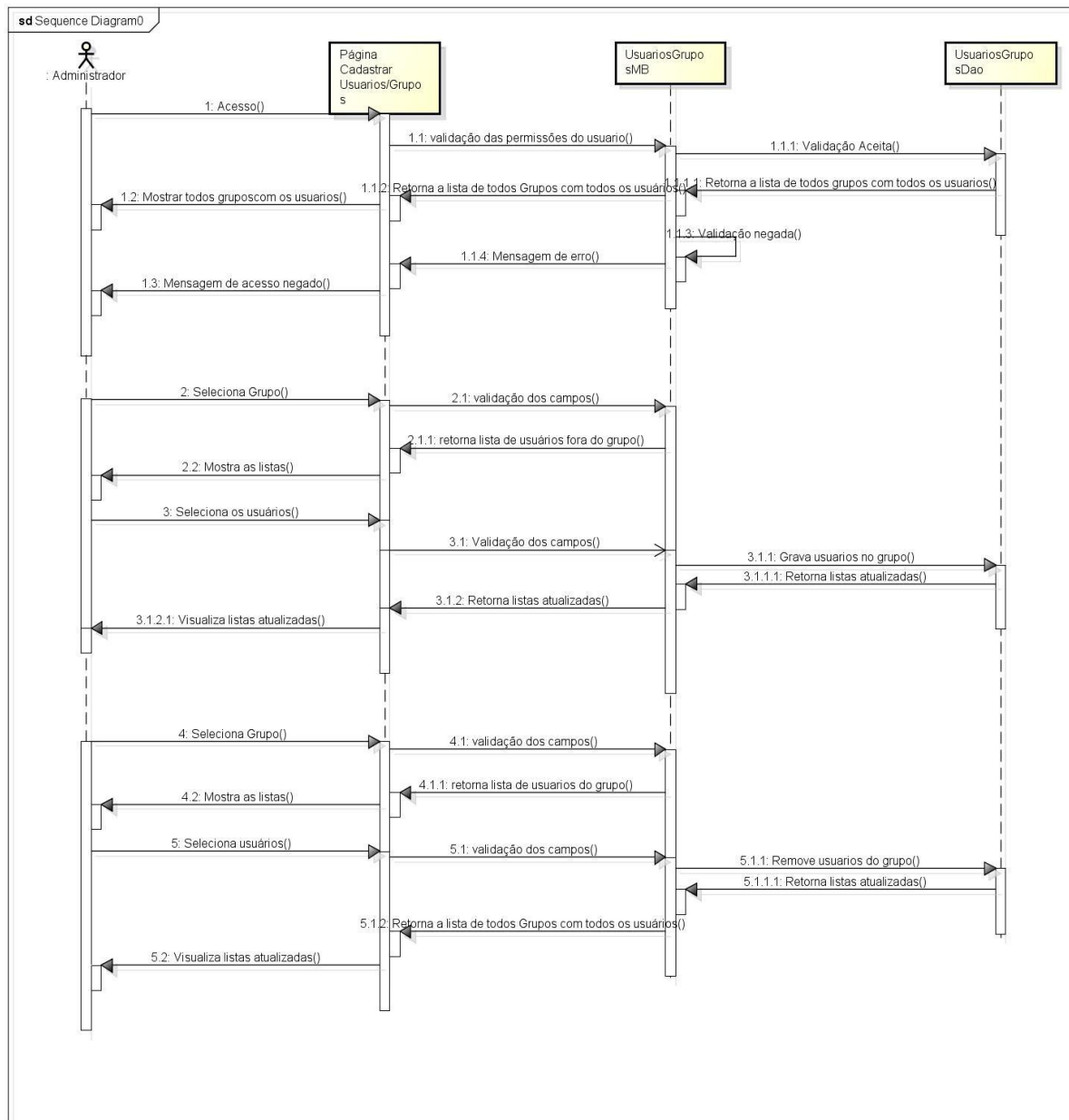


Figura 20- Diagrama de Sequencia: Cadastrar Usuários/Grupos

4.2.2. DIAGRAMA DE CASO DE USO GLOBAL

A figura a seguir apresenta o caso de uso global, onde se tem uma visão geral de todos casos de uso citados anteriormente.

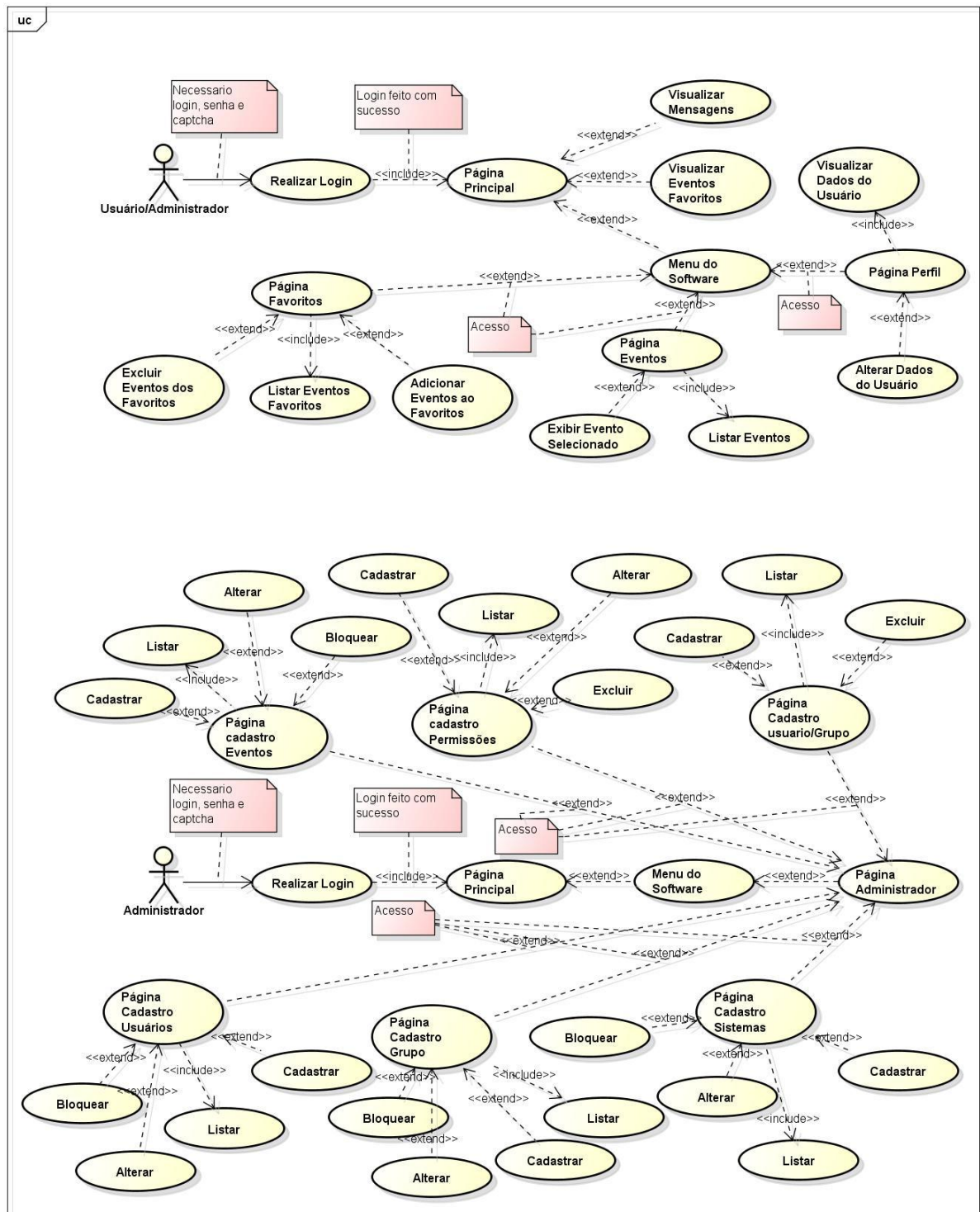


Figura 21- Diagrama de Caso de Uso Global

4.2.3. DIAGRAMA DE PACOTES

No diagrama de pacotes estão representados todos os pacotes utilizados no desenvolvimento do projeto.

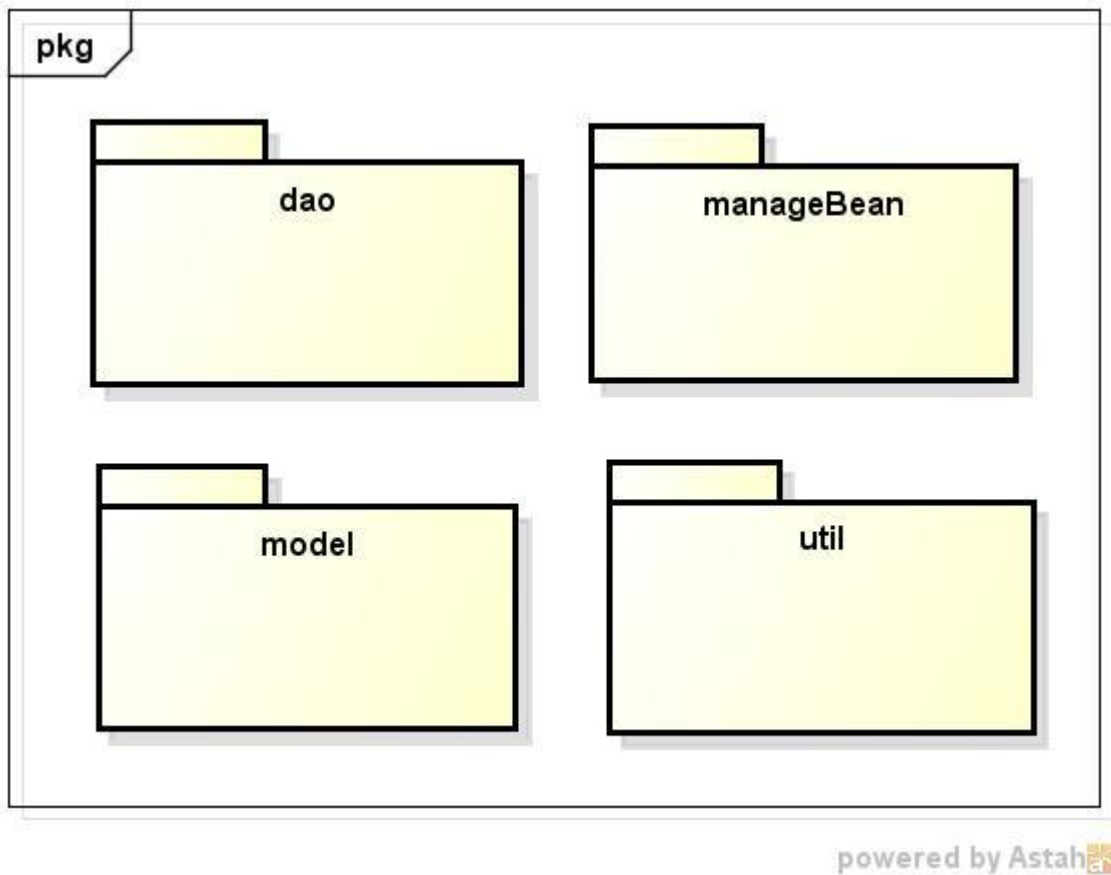


Figura 22- Diagrama de Pacotes

4.2.4. DIAGRAMA DE CLASSE

Nos diagramas de classes, será mostrada as classes utilizadas para o desenvolvimento do projeto.

4.2.4.1. DIAGRAMA DE CLASSES DO PACOTE UTIL

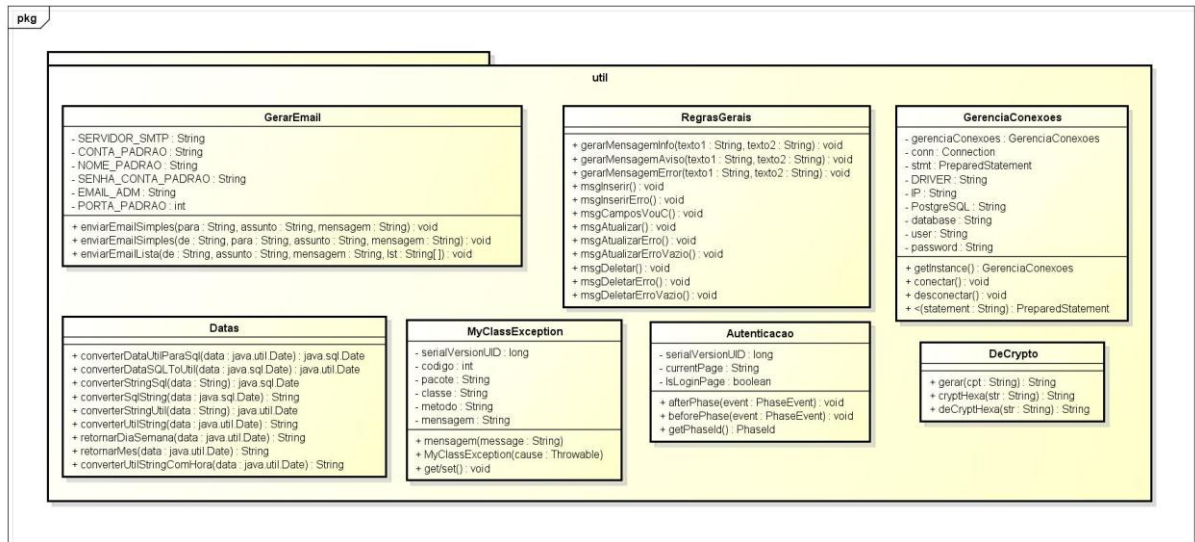
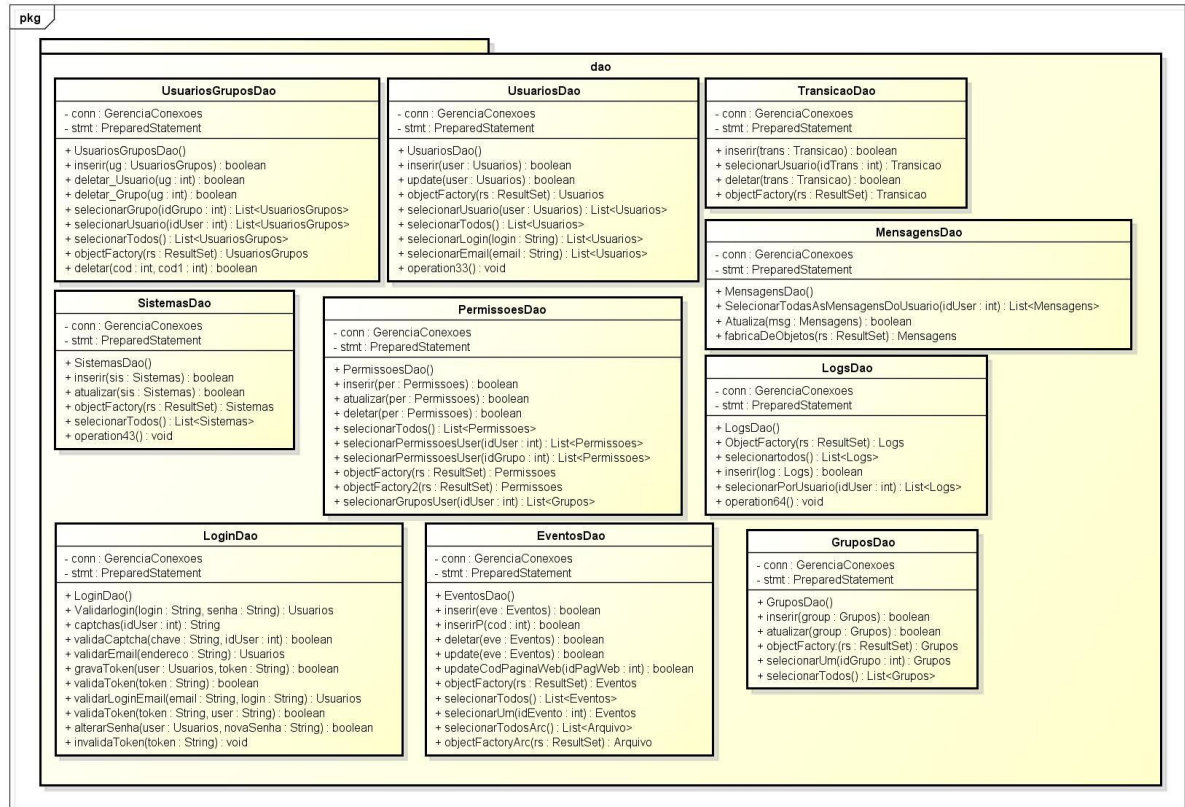


Figura 23- Diagrama de Classes do Pacote útil

4.2.4.2. DIAGRAMA DE CLASSES DO PACOTE DAO

Dentro do pacote dao, está todas as classes que servem para a persistência dos dados no banco de dados.

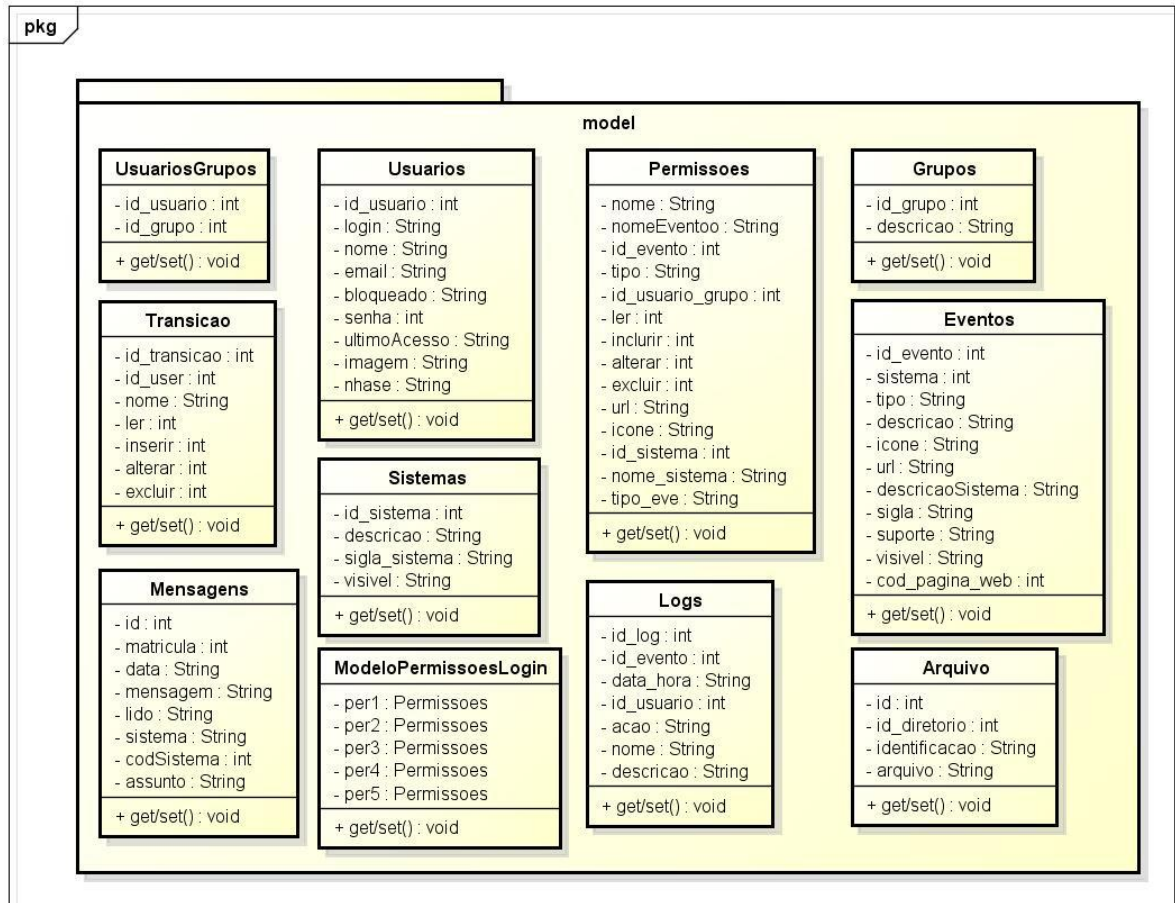


powered by Astah

Figura 24- Diagrama de Classes do Pacote dao

4.2.4.3. DIAGRAMA DE CLASSES DO PACOTE MODEL

Dentro do pacote model, esta todo o mapeamento das tabelas criadas especificamente para o projeto desenvolvido.



powered by Astah

Figura 25- Diagrama de Classes do Pacote model

4.2.4.4. DIAGRAMA DE CLASSES DO PACOTE MANAGEBEAN

Dentro do pacote managebean, estão localizadas todas as classes que se encontram as regras do sistema.

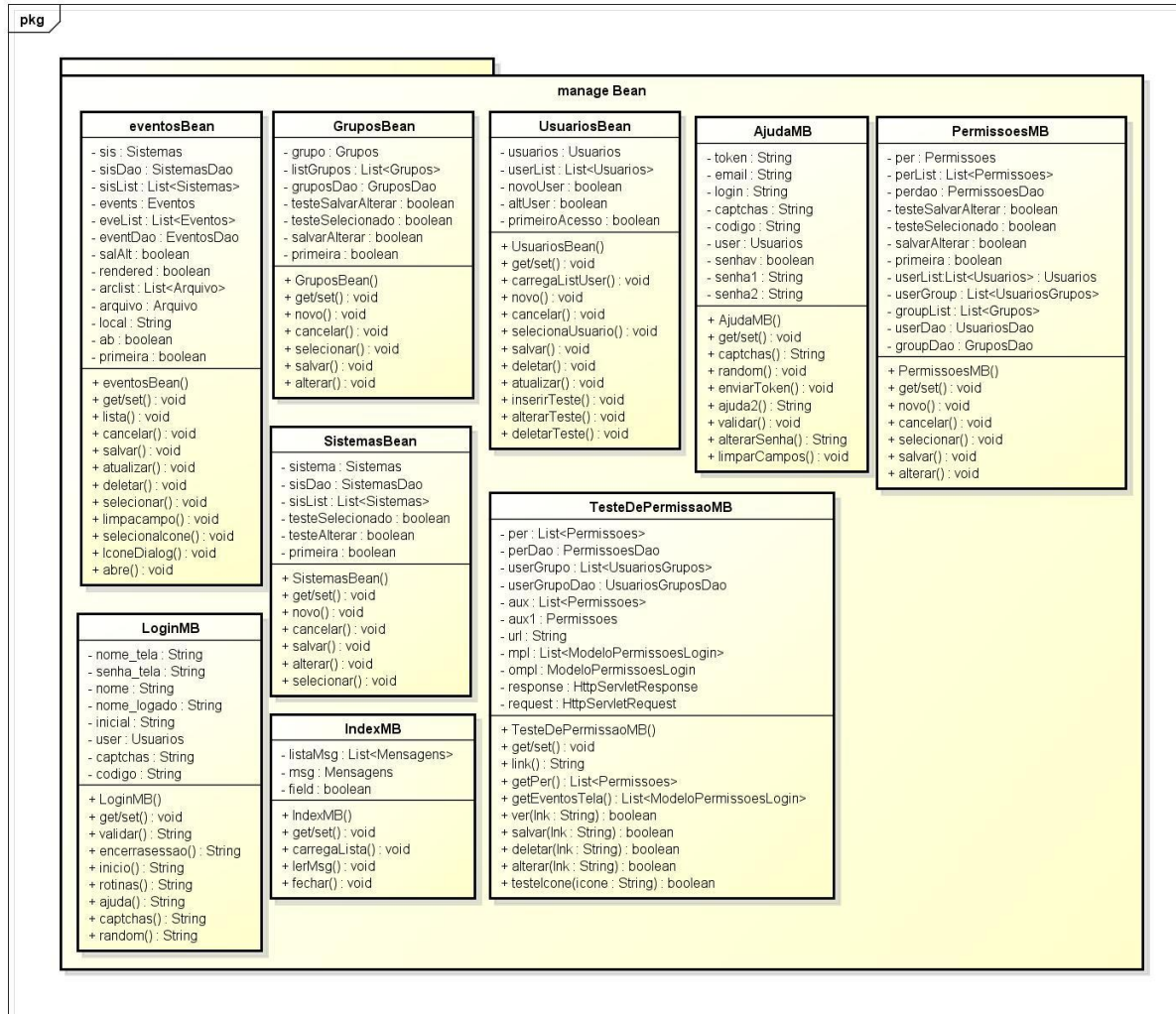
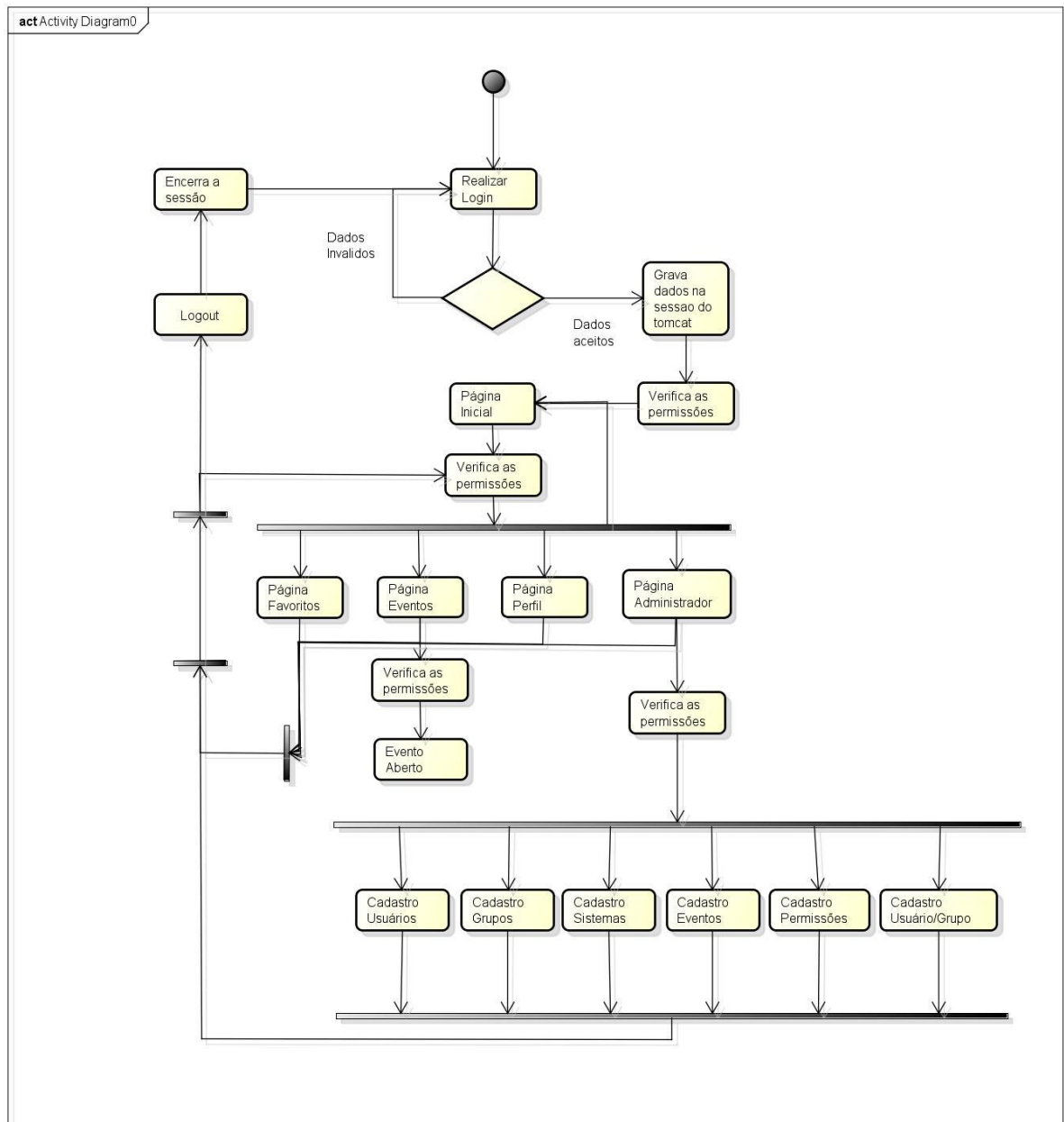


Figura 26- Diagrama de Classes do Pacote manageBean

4.2.5. DIAGRAMA DE ATIVIDADES

O diagrama de atividades demonstra todas as sequencias possíveis dentro do projeto.



powered by Astah

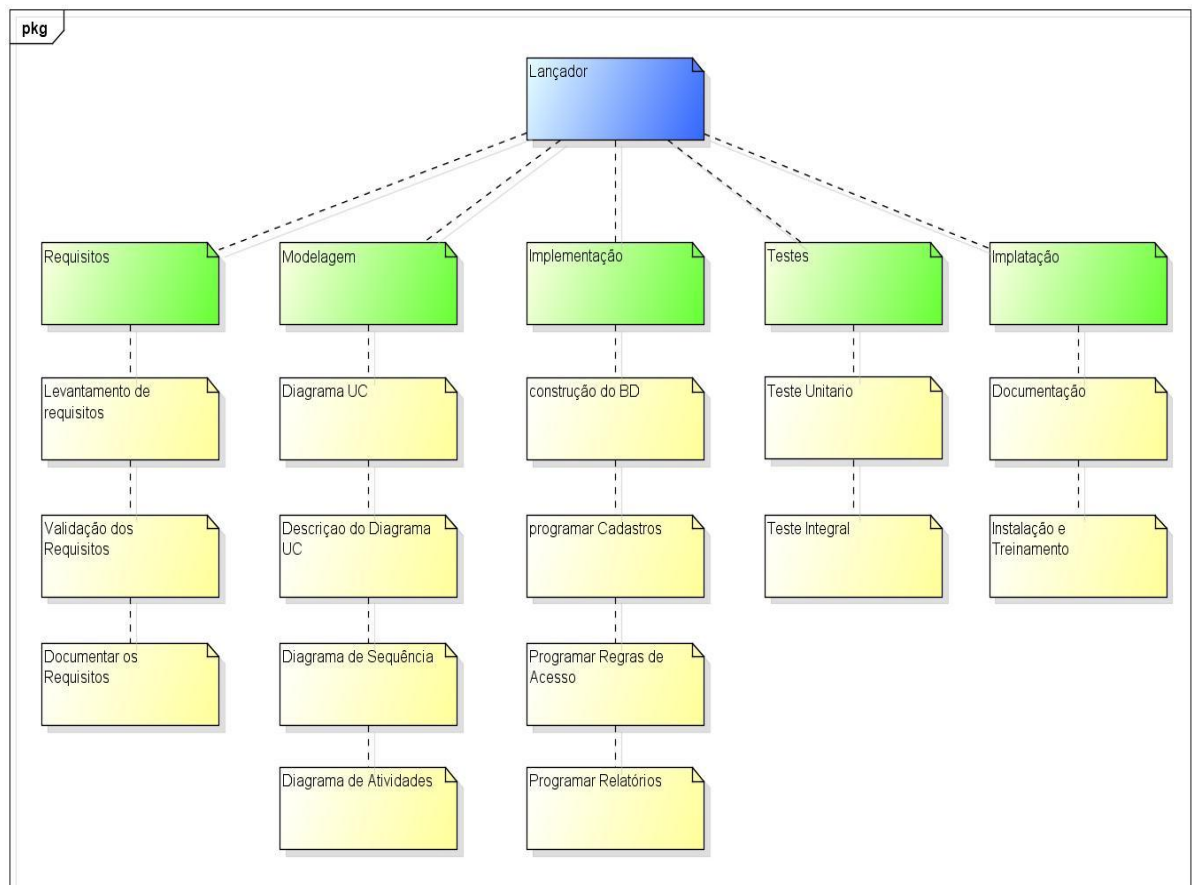
Figura 27- Diagrama de Atividade

4.3. PLANEJAMENTO DO PROJETO

No planejamento do projeto, está especificado todos os requisitos para o desenvolvimento.

4.3.1. WBS – WORK BREAKDOWN STRUCTURE

WBS é um agrupamento de elementos do projeto que organiza e define total de abrangência do projeto.



powered by Astah

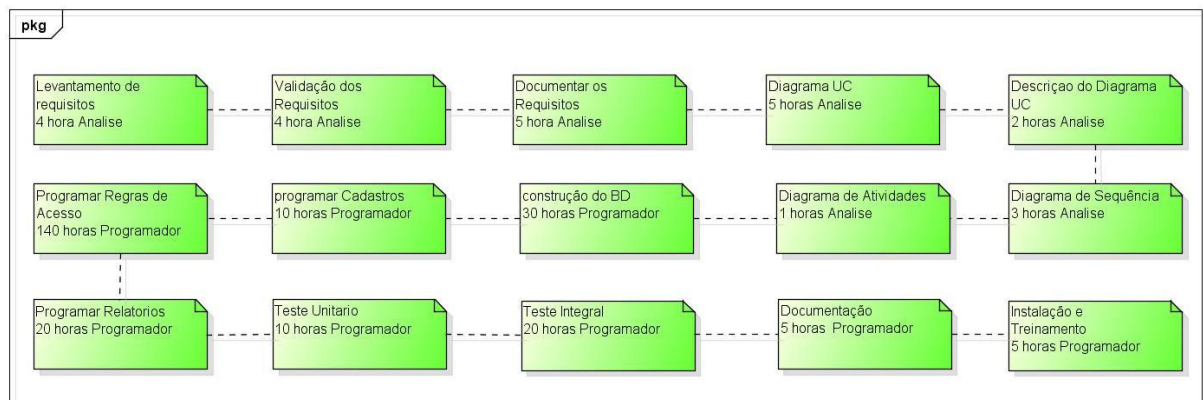
Figura 28- Diagrama WBS

4.3.2. DEFINIÇÃO DAS ATIVIDADES

- Levantamento dos requisitos;
- Análise dos requisitos;
- Validação dos requisitos;
- Especificação dos requisitos;
- Caso de Uso;
- Especificação dos casos de uso e Diagramas de Sequencia;
- Diagrama de classes;
- Diagrama de atividades;
- Programação;
- Testes;
- Treinamento.

4.3.3. SEQUENCIAMENTO DAS ATIVIDADES DEFINIDAS

Aqui está representada a sequencia em que ocorreu o desenvolvimento do projeto.



powered by Astah

Figura 29- Sequenciamento das Atividades Definidas

4.3.4. RECURSOS NECESSÁRIOS PARA O DESENVOLVIMENTO DO PROJETO

Estão listados a baixo os itens necessários para o desenvolvimento deste projeto.

1 (um) Analista de Sistema;

1 (um) Programador Java;

1 (um) Notebook;

1 (uma) Impressora;

Sistema Operacional Windows – Distribuição 7;

IDE para o desenvolvimento: *Netbeans*;

IDE para modelagem de dados: *Astah*;

Linguagem para o desenvolvimento: Java;

Servidor de Aplicação: *Tomcat*;

Bando de dados: *PostgreSQL*;

Aplicativos: LibreOffice.

4.3.5. ESTIMATIVA DE CUSTOS

Esta estimativa foi baseada em valores para o estado de São Paulo.

Analista de Sistema

Quantidade de horas: 24

Custo da hora: R\$ 60,00 (sessenta reais)

Custo total: (Total de dias * Custo diário) = R\$ 1.400,00 (mil e quatrocentos reais)

Programador

Quantidade de horas: 240

Custo da hora: R\$ 40,00 (quarenta reais)

Custo total: (Total de dias * Custo diário) = R\$ 9.600,00 (nove mil e seiscentos reais)

Notebook

Valor Unitário: R\$ 1.650,00 (mil e seiscentos e cinquenta)

Horas de uso: 264 horas

Depreciação: (R\$ 1.650,00 / 24 meses) = R\$ 68,75 (sessenta e oito reais e Setenta e cinco centavos) por mês

Custo dia: (R\$ 68,75 / 30 dias) = R\$ 2,30 (dois reais e trinta centavos) por dia

Custo do Notebook: (Total de dias * Custo dia) = R\$ 75,90 (setenta e cinco reais e noventa centavos)

Impressora

Valor Unitário: R\$ 450,00 (quatrocentos e cinquenta reais)

Horas de uso: 80 horas

Depreciação: (R\$ 450,00 / 24 meses) = R\$ 18,75(dezoito reais e setenta e cinco centavos) por mês

Custo dia: (R\$ 18,75 / 30 dias) = R\$ 0,63 (sessenta e três centavos) por dia

Custo da Impressora: (Total de dias * Custo dia) = R\$ 6,30 (seis reais e trinta centavos)

Sistema Operacional Windows – Distribuição 7

Valor Unitário: R\$ 655,00 (seiscentos e cinquenta e cinco reais)

Horas de uso: 264 horas

Depreciação: (R\$ 655,00 / 24 meses) = R\$ 27,30(vinte e sete reais e trinta centavos) por mês

Custo dia: (R\$ 27,30 / 30 dias) = R\$ 0,91 (noventa e um centavos) por dia

Custo do Sistema Operacional: (Total de dias * Custo dia) = R\$ 30,03(trinta reais e três centavos)

IDE para o desenvolvimento – *Netbeans*

Custo: R\$ 0,00 (*Freeware*)

IDE para a modelagem de dados – Astah

Custo: R\$ 0,00 (*Freeware*)

Linguagem para o desenvolvimento – Java;

Custo: R\$ 0,00 (*Freeware*)

Servidor de Aplicação – *Tomcat*;

Custo: R\$ 0,00 (*Freeware*)

Bando de dados – PostgreSQL;

Custo: R\$ 0,00 (*Freeware*)

Aplicativos – *LibreOffice*.

Custo: R\$ 0,00 (*Freeware*)

4.3.6. ORÇAMENTO DO PROJETO

Este orçamento foi baseado nos valores citados no tópico anterior.

Recursos Valor

Analista de Sistema R\$ 1.400,00

Programador Java R\$ 9.600,00

Notebook R\$ 75,90

Impressora R\$ 6,30

Sistema Operacional Windows – Distribuição 7; R\$ 30,03

IDE para o desenvolvimento: *Netbeans*; R\$ 0,00

IDE para modelagem de dados: *Astah*; R\$ 0,00

Linguagem para o desenvolvimento: *Java*; R\$ 0,00

Servidor de Aplicação: *Tomcat*; R\$ 0,00

Bando de dados: *PostgreSQL*; R\$ 0,00

Aplicativos: *LibreOffice*. R\$ 0,00

Custo Total do Projeto: R\$ 11.112,23

| Recursos | Valor |
|--|----------------------|
| Analista de Sistema | R\$ 1.400,00 |
| Programador Java | R\$ 9.600,00 |
| Notebook | R\$ 75,90 |
| Impressora | R\$ 6,30 |
| Sistema Operacional Windows – Distribuição 7 | R\$ 30,03 |
| IDE para o desenvolvimento: <i>Netbeans</i> | R\$ 0,00 |
| IDE para modelagem de dados: <i>Astsh</i> ; | R\$ 0,00 |
| Linguagem para o desenvolvimento: Java | R\$ 0,00 |
| Servidor de Aplicação: <i>Tomcat</i> | R\$ 0,00 |
| Bando de dados: <i>PostgreSQL</i> | R\$ 0,00 |
| Aplicativos: <i>LibreOffice</i> | R\$ 0,00 |
| Custo Total do Projeto: | R\$ 11.112,23 |

Tabela 1- Orçamento do Projeto

5. CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS

Com o estudo realizado neste projeto, conclui-se que segurança da informação compreende uma série de medidas que visam proteger e preservar a informação em seus sistemas, garantindo os itens mencionados no trabalho: integridade, disponibilidade, não repúdio, autenticidade e confidencialidade.

Os componentes de criptografia, confidencialidade, integridade, não repúdio e autenticidade, garantem um bom software web. Com isto torna-se necessário uma boa estratégia com o intuito de criar uma segurança que seja aceitável para se manter um software web em uso.

Contudo, não basta somente o software utilizar os itens, mas também a empresa usar de incentivo a educação em questões de segurança, pois o maior risco nem sempre vem por ataques externos, mas sim de falha humana com o manuseio do sistema.

Atualmente, numa era onde conhecimento e informação são fatores de grande importância para qualquer organização, segurança da informação é um pré-requisito para todo e qualquer sistema de informações online. Nesse sentido, há uma relação de dependência entre a segurança da informação e seus itens de desenvolvimento.

Alguns trabalhos futuros a ser desenvolvido para este projeto são:

- Melhorar a validação com REST.
- Utilizar internacionalização.
- Aperfeiçoar a parte de upload de imagens

REFERÊNCIAS

ALBUQUERQUE, Ricardo; RIBEIRO, Bruno. **Segurança no Desenvolvimento de Software: Como desenvolver sistemas seguros e avaliar a segurança de aplicações desenvolvidas com base na ISO 15.408**, Rio de Janeiro: Editora Campus, 2002.

ALVES, Fabrício Silva, André Luiz Alves. **APLICABILIDADE DOS PRINCIPAIS CONCEITOS DE SEGURANÇA NO CICLO DE VIDA DE SISTEMAS EM SOFTWARE HOUSES**, Pontifícia Universidade Católica de Goiás (PUC-Goiás) – Goiânia – Brasil Especialização em Qualidade e Gestão de *Softwares* – Dezembro de 2009.

BEZERRA, Eduardo. **Princípios de Análise e Projeto de Sistemas com UML**. Rio de Janeiro: Editora Campus, 2002.

Caelun Ensino e Inovação. Disponível em <<https://www.caelum.com.br/apostila-java-testes-jsf-web-services-design-patterns/introducao-ao-jsf-e-primefaces/#7-2-caracteristicas-do-jsf>> Acesso em: 07 de março de 2014.

Deitel, H. M.; Deitel P. J. *Java como programar*. Trad. Lisbôa, Carlos Arthur Lang,. 4.ed. Porto Alegre, RS : BOOKMAN Companhia Editora, 2003.

FERNANDES, Alexandre Furtado; GARCIA, Carla Patrícia; CAMPOS, Eloim de Oliveira; FARIA, Fabrício Tavares de; REIS, Gustavo Henrique da Rocha; CAMPOS, Samira Fávero; OLIVEIRA, Wellington Moreira de. **Plano Diretor de Tecnologia da Informação e Comunicação**. Rio Pomba, 2008.

Freitas, Welbson (2007). História do Java. Disponível em: <<http://infoblogs.com.br/frame/goframe.action?contentId=13593>>. Acesso em: 13 de Fevereiro de 2014.

GOMES, Kleiton da Silva; SANTOS, Newton. **Segurança no Desenvolvimento de Aplicações Web**. 2006. Trabalho de Conclusão de Curso do Curso de Bacharel em Ciência da Computação - Centro de Ciências Exatas e Tecnológicas – CCET – Universidade da Amazônia – UNAMA, Amazonas, Manaus, 2006.

GONÇALVES, Edson. **Desenvolvendo Aplicações Web com JSP Servelets JSF Hibernate EJB 3 Persistence e Ajax**. 1. ed. Brasil: Editora Ciência Moderna, 2007.

Historia do PostgreSQL. Disponível em<https://wiki.postgresql.org/wiki/Main_Page>. Acesso em 25 de fevereiro de 2014.

JUKEMURA, Anibal Santos. **Administração da Segurança e Auditoria em Sistemas de Informação**, Universidade Católica de Goiás – UCG – Goiânia, GO – Brasil – Material didático para Especialização em Qualidade e Gestão de *Softwares*, 2009. de Bacharel em Ciência da Computação, 2006.

JUNIOR, Peter Jandl. **Introdução ao Java**. Núcleo de Educação a Distância. Universidade de São Francisco, 1999. Disponível em <<http://markmail.org/download.xqy?id=zdem6rhmgwp5sefm&number=1>>. Acesso em 14 out. 2013.

MELLO,Emerson Ribeiro de;WANGHAM, Michelle S. et. al., Joni da Silva Fraga, Edson Camargo. **Segurança em Serviços Web**, Minicurso SBSeg 2006, Departamento de Automação e Sistemas, Universidade Federal de Santa Catarina

Pacievitch Y. *História do Java*. Disponível em: <<http://www.infoescola.com/informatica/historia-do-java/>>. Acesso em: 12 de dezembro de 2013.

Plantão nerd.com. Disponível em: <<http://www.plantaonerd.com/blog/2011/04/18/astah-community-um-software-para-trabalha-com-umls/#more-411>> Acesso em: 31 de janeiro de 2014.

Sobre o NetBeans. **O que é NetBeans**. Disponível em <<http://pgdoctbr.sourceforge.net/pg82/intro-what-is.html> >. Acesso em: 11 jan. 2014.

Sobre o PostgreSQL. **O que é PostgreSQL**. Disponível em <<http://pgdoctbr.sourceforge.net/pg82/intro-what-is.html> >. Acesso em: 14 out. 2013.

Spanceski, Francini Reitz. **Política de Segurança da Informação: Desenvolvimento de um modelo voltado para Instituições de Ensino**, Instituto Superior de Tupy – Joinville, SC – Brasil – Trabalho de Conclusão de Curso em Sistemas de Informação, 2004.

ULLMAN, Larry. **PHP 6 and MySQL 5 for Dynamic Web Sites**. 1. ed. Brasil: Editora Peachpit Press, 2008.

APÊNDICE A

Abaixo esta representada a tela inicial do projeto, onde nota-se o login juntamente com o captcha desenvolvido para este projeto.



Figura 30 – LOGIN

A figura a seguir mostra o menu principal, onde são visíveis as mensagens e ícones de usuário.



Figura 31 – MENU

O cadastro de usuários está ilustrado abaixo, onde o administrador cadastra todos os usuários do sistema.

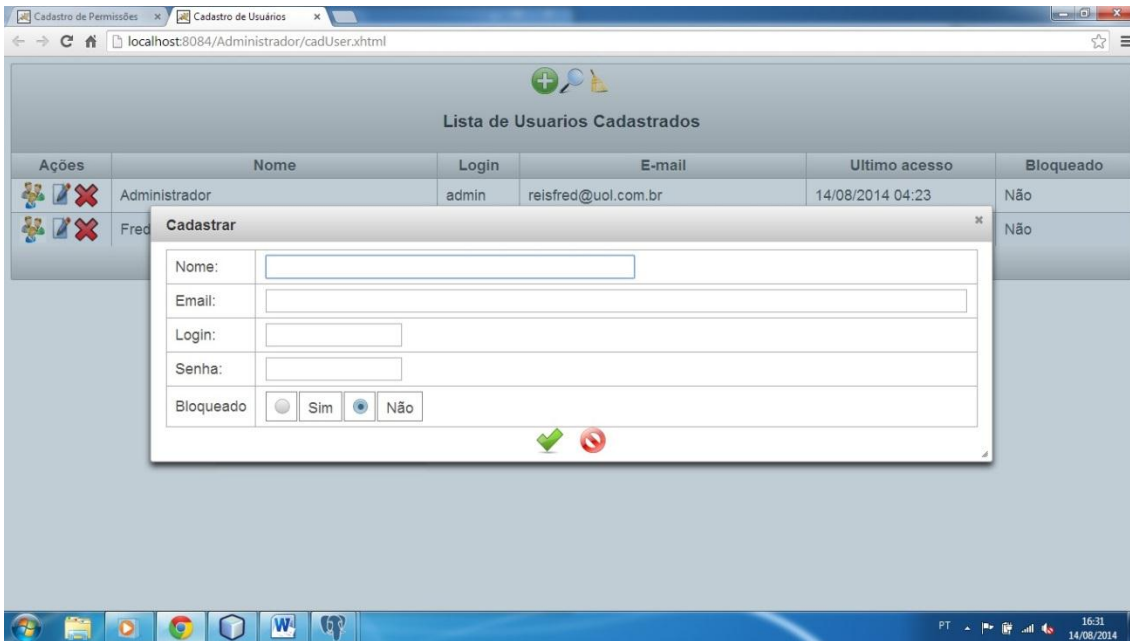


Figura 32 - CADASTRO DE USUARIO

Na figura abaixo é representada a área de cadastro de permissões do sistema, onde o administrador pode conceder ou retirar permissão de um usuário ou grupo.

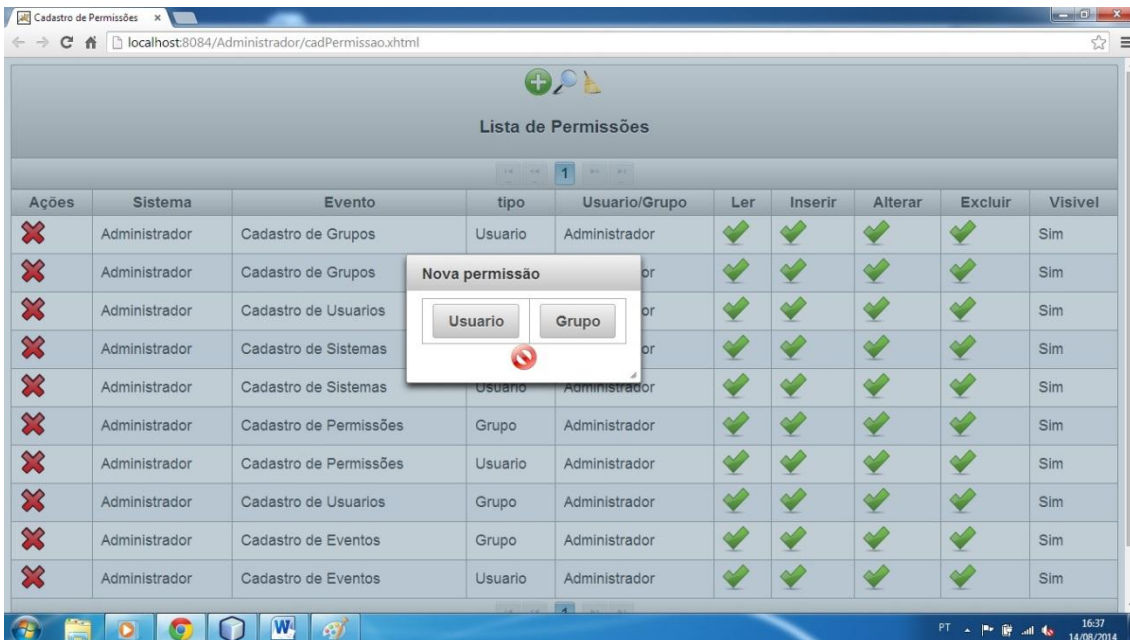


Figura 33 - CADASTRO DE PERMISSÕES