

FELIPE DE LIMA RODRIGUES

SISTEMA DE GERENCIAMENTO DE EQUIPE

ASSIS

2014

FELIPE DE LIMA RODRIGUES

SISTEMA DE GERENCIAMENTO DE EQUIPE

Trabalho de Conclusão de Curso
apresentado ao Instituto Municipal de Ensino
Superior de Assis, como requisito do Curso
de Graduação.

Orientando: Felipe de Lima Rodrigues
Orientador: Prof. Dr. Alex Sandro Romeo de Souza Poletto
Analisador: Prof. Esp. Guilherme de Cleve Farto

ASSIS

2014

FICHA CATALOGRÁFICA

RODRIGUES, Felipe de Lima

Sistema de Gerenciamento de Equipes / Felipe de Lima Rodrigues. Fundação Educacional do Município de Assis - FEMA, Assis, 2014.

68 Páginas.

Orientador: Prof. Dr. Alex Sandro Romeo de Souza Poletto.

Trabalho de Conclusão de Curso. Instituto Municipal de Ensino Superior de Assis – IMESA.

1. Gerenciamento. 2. Desenvolvimento WEB.

CDD: 001.61
Biblioteca da FEMA

SISTEMA DE GERENCIAMENTO DE EQUIPES

FELIPE DE LIMA RODRIGUES

Trabalho de Conclusão de Curso apresentado ao Instituto Municipal de Ensino Superior de Assis – IMESA, como requisito do Curso de Graduação, analisado pela seguinte comissão examinadora:

Orientador: Prof. Dr. Alex Sandro Romeo de Souza Poletto.

Analisador: Prof. Esp. Guilherme de Cleve Farto.

AGRADECIMENTOS

Aos familiares, meus pais Julia e Valer e meu irmão Rafael que me inspiraram a buscar sempre o melhor.

A namorada, Flávia que sempre acreditou e me incentivou a realizar este trabalho da melhor forma.

Ao professor, Alex pela orientação e sabedoria que contribuiu no desenvolvimento de um projeto muito bom para conclusão deste curso.

EPÍGRAFE

Minha energia é o desafio, minha motivação é o impossível, e é por isso que eu
preciso ser, à força e a esmo, inabalável.

Augusto Branco

RESUMO

O software foi planejado para gerenciar de forma eficiente equipes de trabalho de empresas de pequeno e médio porte. O sistema oferece gestão de equipes, projetos, tarefas, colaboradores entre outras informações e fornece *feedback* dos resultados analisados, auxiliando em tomadas de decisões importante para qualquer organização. Dentre as funcionalidades estão o acompanhamento de duração de tarefas e projetos, relatórios e gráficos sobre as informações geradas, controle sobre a produtividade dos colaboradores.

Como principais tecnologias o software utiliza a linguagem de programação Java o desenvolvimento do software, com o auxílio dos frameworks Java Server Faces 2.2 para construção da interface WEB e o Hibernate para persistência de Dados.

Palavras Chave: 1. Desenvolvimento WEB; 2. Java; 3. Java Server Faces; 4. Hibernate; 5. Orientação a Objetos.

ABSTRACT

The software was planned to efficiently manage work teams of small and medium-sized businesses. The system offers team management, projects, tasks, employees, among other information and yet gives feedback from the analysed results, helping taking decisions that are important to any organization. Amongst the functions are the monitoring of tasks and projects length, reports and graphics about the information and control of the employees productivity.

As main technologies, the software uses the programming language Java the software development, with assistance of frameworks Java Server Faces 2.2 to build the interface WEB and the Hibernate to data persistence.

Keywords: 1. WEB Development; 2. Java; 3. Java Server Faces; 4. Hibernate; 5. Object Orientation.

LISTA DE FIGURAS

Figura 1. Diagrama de Caso de Uso Administrador	33
Figura 2. Diagrama de Caso de Uso Líder de Equipe.....	34
Figura 3. Diagrama de Caso de Uso Colaborador	35
Figura 4. Diagrama de Caso de Uso Manter Cargos	36
Figura 5. Diagrama de Caso de Uso Manter Colaboradores	37
Figura 6. Diagrama de Caso de Uso Manter Projetos.....	38
Figura 7. Diagrama de Caso de Uso Manter Tarefas.....	39
Figura 8. Diagrama Caso de Uso Manter Tópicos	40
Figura 9. Diagrama Caso de Uso Manter Equipes	41
Figura 10. Diagrama de Caso de Uso Emitir Relatórios.....	42
Figura 11. Diagrama de Caso de Uso Postar no Mural.....	43
Figura 12. Diagrama de Caso de Uso Consultar Horas Trabalhadas	44
Figura 13. Diagrama de Caso de Uso Iniciar Tarefa	45
Figura 14. Caso de Uso Concluir Tarefa	46
Figura 15. Caso de Uso Concluir Projeto	47
Figura 16. Diagrama de Atividade Iniciar Tarefa	48
Figura 17. Diagrama de Atividade Concluir Tarefa.....	49
Figura 18. Diagrama de Classe.....	50
Figura 19. Diagrama Entidade-Relacionamento.....	51
Figura 20. Diagrama WBS.....	52
Figura 21. Diagrama Sequência de Atividades	53
Figura 22. Acesso ao Sistema.....	54
Figura 23. Criptografia MD5	55
Figura 24. Interface iniciar tarefa.....	56
Figura 25. Interface concluir tarefa.....	57
Figura 26. Interface concluir projeto	58
Figura 27. Interface consultar horas trabalhadas	59
Figura 28. Interface do gráfico de situação das tarefas por equipe.....	60
Figura 29. Interface relatório gerado pelo sistema	61

LISTA DE TABELAS

Tabela 1. Especificação Caso de Uso Manter Cargos	36
Tabela 2. Especificação Caso de Uso Manter Colaboradores	37
Tabela 3. Especificação Caso de Uso Manter Projetos	38
Tabela 4. Especificação Caso de Uso Manter Tarefas	39
Tabela 5. Especificação Caso de Uso Manter Tópicos	40
Tabela 6. Especificação Caso de Uso Manter Equipes.....	41
Tabela 7. Especificação Caso de Uso Emitir Relatórios	42
Tabela 8. Especificação Caso de Uso Postar no Mural	43
Tabela 9. Especificação Caso de Uso Consultar Horas Trabalhadas.....	44
Tabela 10. Especificação Caso de Uso Iniciar Tarefa.....	45
Tabela 11. Especificação Caso de Uso Concluir Tarefa	46
Tabela 12. Especificação Caso de Uso Concluir Projeto	47

SUMÁRIO

1. INTRODUÇÃO	15
1.1. OBJETIVOS.....	15
1.2. PÚBLICO ALVO	16
1.3. JUSTIFICATIVA.....	16
2. GERENCIAMENTO DE EQUIPES	17
2.1. INTRODUÇÃO.....	17
2.2. POR QUE GERENCIAR EQUIPES DE TRABALHO	17
2.3. GERENCIAR TEMPO DE ATIVIDADES	18
2.4. MONITORAMENTO	18
2.5. DESEMPENHO DA EQUIPE	19
2.6. RELATÓRIOS.....	19
3. MÉTODO DE DESENVOLVIMENTO	20
3.1. ANÁLISE.....	20
3.2. APLICAÇÃO	20
3.3. BANCO DE DADOS	21
3.4. RELATÓRIOS.....	21
4. ORIENTAÇÃO A OBJETOS	22
4.1. CLASSES	22
4.2. OBJETOS	22
4.3. ENCAPSULAMENTO	22
4.4. HERANÇA	23
4.5. POLIMORFISMO.....	23

5. BANCO DE DADOS	24
5.1. ENTIDADES	24
5.2. RELACIONAMENTOS	25
5.2.1. UM-PARA-UM	25
5.2.2. UM-PARA-MUITOS.....	25
5.2.3. MUITOS-PARA-MUITOS	26
6. DESCRIÇÃO DO AMBIENTE DE DESENVOLVIMENTO.....	27
6.1. LINGUAGENS UTILIZADAS	27
6.1.1. JAVA	27
6.1.2. HTML.....	27
6.1.3. CSS	28
6.1.4. UML.....	28
6.2. FERRAMENTAS DE DESENVOLVIMENTO	29
6.2.1. ECLIPSE	29
6.2.2. MYSQL.....	29
6.2.3. NAVICAT	29
6.3. FRAMEWORKS.....	30
6.3.1. HIBERNATE.....	30
6.3.2. JSF (JAVA SERVER FACES)	30
7. ANÁLISE.....	31
7.1. FERRAMENTAS DE ANÁLISE	31
7.1.1. DBDESIGNER 4.....	31
7.1.2. ASTAH COMMUNITY	31
7.1.3. MICROSOFT VISIO 2013.....	32
7.2. LEVANTAMENTO E ESPECIFICAÇÃO DOS REQUISITOS	32

7.2.1. DESCRIÇÃO DOS OBJETIVOS	32
7.2.2. RESULTADOS ESPERADOS.....	32
7.3. DIAGRAMAS UML	33
7.3.1. CASOS DE USO ADMINISTRADOR.....	33
7.3.2. CASOS DE USO LÍDER DE EQUIPE	34
7.3.3. CASOS DE USO COLABORADOR.....	35
7.3.4. ESPECIFICAÇÃO CASO DE USO MANTER CARGOS	36
7.3.5. ESPECIFICAÇÃO CASO DE USO MANTER COLABORADORES.....	37
7.3.6. ESPECIFICAÇÃO CASO DE USO MANTER PROJETOS.....	38
7.3.7. ESPECIFICAÇÃO CASO DE USO MANTER TAREFAS	39
7.3.8. ESPECIFICAÇÃO CASO DE USO MANTER TÓPICOS.....	40
7.3.9. ESPECIFICAÇÃO CASO DE USO MANTER EQUIPE.....	41
7.3.10. ESPECIFICAÇÃO CASO DE USO EMITIR RELATÓRIOS.....	42
7.3.11. ESPECIFICAÇÃO CASO DE USO POSTAR NO MURAL	43
7.3.12. ESPECIFICAÇÃO CASO DE USO CONSULTAR HORAS TRABALHADAS	44
7.3.13. ESPECIFICAÇÃO CASO DE USO INICIAR TAREFA.....	45
7.3.14. ESPECIFICAÇÃO CASO DE USO CONCLUIR TAREFA.....	46
7.3.15. ESPECIFICAÇÃO CASO DE USO CONCLUIR PROJETO	47
7.3.16. DIAGRAMA ATIVIDADE INICIAR TAREFA	48
7.3.17. DIAGRAMA ATIVIDADE CONCLUIR TAREFA.....	49
7.3.18. DIAGRAMA DE CLASSES	50
7.3.19. DIAGRAMA ER	51
7.3.20. WBS	52
7.3.21. SEQUÊNCIA DAS ATIVIDADES	53
7.4. ACESSO AO SISTEMA.....	54

7.4.1. FILTROS (FILTER).....	54
7.4.2. CRIPTOGRAFIA MD5.....	55
8. INTERFACES.....	56
8.1. INTERFACE INICIAR TAREFA.....	56
8.2. INTERFACE CONCLUIR TAREFA.....	57
8.3. INTERFACE CONCLUIR PROJETO.....	58
8.4. INTERFACE CONSULTAR HORAS TRABALHADAS.....	59
8.5. INTERFACE DO GRÁFICO DE SITUAÇÃO DAS TAREFAS POR EQUIPE.....	60
8.6. INTERFACE RELATÓRIO GERADO PELO SISTEMA.....	61
9. CONCLUSÃO.....	62
REFERÊNCIAS.....	63
APÊNDICE A – Trigger insert_tarefa.....	65
APÊNDICE B – Trigger update_tarefa.....	65
APÊNDICE C – Trigger delete_tarefa.....	66
APÊNDICE D – Trigger update_topico.....	66
APÊNDICE E – Trigger update_banco_hora.....	67
APÊNDICE F – Trigger insert_equipe.....	67
APÊNDICE G – Trigger delete_equipe.....	68

1. INTRODUÇÃO

O Sistema de Gerenciamento de Equipes tem por objetivo facilitar o acompanhamento das tarefas pelo gerente do projeto, oferecendo assim, melhor controle dos prazos, do andamento do projeto, do controle das tarefas realizadas, em andamento ou que ainda não foram designadas.

Com o sistema implantado, o gerente de projetos poderá designar tarefas com mais agilidade para a equipe, podendo visualizar simultaneamente as tarefas que estão sendo realizadas por cada membro. Além disso, poderá analisar se o ritmo de andamento é aceitável, através de tópicos concluídos pelo funcionário designado, avaliando assim se a tarefa foi concluída dentro do prazo desejado bem como também priorizar tarefas que sejam importantes do momento.

Ter controle sobre as atividades realizadas pode ser de suma importância para o gerente avaliar se os requisitos levantados estão sendo realizados da forma esperada e se não há necessidade de adaptações do mesmo, melhorando a qualidade e custo do projeto.

1.1. OBJETIVOS

O Sistema de Gerenciamento de Equipe tem por objetivo servir de auxiliar na gerencia de equipes, oferecendo os recursos de distribuição de tarefas aos membros da equipe, priorizando atividades que estejam atrasadas ou com urgência. Oferecer dados informativos que possibilite ao gerente do projeto mensurar a qualidade da equipe e o andamento das atividades, avaliando a existência de dificuldades no desenvolvimento do projeto ou se há necessidade de realizar alterações.

A implantação do sistema oferecerá também, maior segurança, confiabilidade na equipe, melhor avaliação das capacidades dos envolvidos, obtendo assim rendimento dos projetos em que a equipe possa estar envolvida.

1.2. PÚBLICO ALVO

O público alvo deste software envolve o gerente de projeto, que poderá fazer uma melhor análise do projeto por ele gerenciado, como os funcionários envolvidos no projeto, que terão visão de suas tarefas a serem concluídas e metas a serem alcançadas.

1.3. JUSTIFICATIVA

A necessidade de se obter maior controle sobre as atividades realizadas em equipe e a mensuração do andamento do projeto, obtendo assim melhores resultados por toda a equipe, resultando no sucesso do projeto dentro dos padrões pré-estabelecidos, levou o desenvolvimento deste software.

2. GERENCIAMENTO DE EQUIPES

Nesta seção serão abordados fatores envolvidos no processo de gerenciamento de equipes de trabalho baseados no Guia PMBOK – Quarta Edição.

2.1. INTRODUÇÃO

O gerenciamento de equipes de trabalho é um processo que envolve muitos fatores, como estabelecer um plano de projeto, avaliar o desempenho dos membros, fornecer relatórios, resolver conflitos de interesses, entre outras particularidades de uma tarefa complexa como gerenciar equipes de trabalho. Como resultado deste processo muitas são fornecidos *feedback* da empresa auxiliando em tomadas de decisões, melhor ambiente de trabalho, dentre outros fatores (Guia PMBOK – Quarta Edição, 2009).

A necessidade de gerenciar um projeto é uma importante diferença entre projetos de nível profissional de um projeto de nível amador. Examinar o processo de desenvolvimento é um ponto muito importante para manter os prazos dos projetos em andamento, bem como a elaboração de relatórios de acompanhamento e organizar equipes qualificadas a realizar tarefas específicas (Guia PMBOK – Quarta Edição, 2009).

2.2. POR QUE GERENCIAR EQUIPES DE TRABALHO

Quando se trata de equipes de trabalho, reunimos diferentes pessoas em um ambiente e definimos os grupos de trabalho. O que representa pouco e traz resultados insatisfatórios. Grupos de trabalho existem muitos em diversas empresas, porém não significa trabalho sincronizado e eficiente. É necessário gerenciar os

projetos e as atividades que irão ser realizadas pela equipe de forma eficaz de maneira a favorecer a equipe, o projeto e os colaboradores envolvidos.

2.3. GERENCIAR TEMPO DE ATIVIDADES

Responsabilidades e atividades bem definidas são pontos muito importantes dentro de uma equipe de trabalho. É função do líder da equipe fazê-lo e gerenciar o alinhamento das atividades entre todos. Avaliações periódicas e a aplicação de metodologias também garantem que o tempo de trabalho seja bem utilizado. Porém também é necessário ter na equipe, colaboradores auto gerenciáveis quanto a suas atividades e tempo.

2.4. MONITORAMENTO

O monitoramento é processo que envolve o acompanhamento, avaliação, regulação, elaboração de relatórios de status, medições de progresso e previsões, com o objetivo de atender ao plano de gerenciamento proposto (Guia PMBOK – Quarta Edição, 2009).

O monitoramento de um projeto deve ser feito de forma contínua a fim de comparar os progressos e avaliar os reais custos envolvidos com aquelas que foram inicialmente planejados, avaliar os riscos envolvidos e as dificuldades encontradas pelos colaboradores no decorrer do projeto (Sommerville, 2003).

2.5. DESEMPENHO DA EQUIPE

Controlar o desempenho da equipe traz resultados significativos para o projeto, visando um constante aperfeiçoamento dos membros envolvidos (Guia PMBOK – Quarta Edição, 2009).

A avaliação de desempenho em longo prazo pode trazer informações importantes sobre o esclarecimento de papéis dos envolvidos do projeto, identificar problemas desconhecidos e ou não resolvidos e auxiliar na elaboração de planos e metas para o futuro.

2.6. RELATÓRIOS

Os relatórios fornecem dados preciosos para auxiliar na avaliação do projeto, da equipe, no reconhecimento do esforço de cada membro, no controle do cronograma entre outras funcionalidades (Guia PMBOK – Quarta Edição, 2009).

A elaboração dos relatórios de acompanhamento são geralmente realizados pelo gerente do projeto, com o objeto de obter dados atualizados do andamento do projeto, desempenho dos colaboradores, custo envolvidos, atividades que necessitem de esforço inesperado, etc. (Sommerville, 2003).

3. MÉTODO DE DESENVOLVIMENTO

Nesta seção serão citadas as ferramentas e utilizadas e o seu propósito na elaboração do trabalho.

3.1. ANÁLISE

A análise do sistema envolve levantamento e validação de requisitos, elaboração de diagramas referentes à estrutura e funcionamento do software. Para a análise e documentação do projeto, será utilizada a metodologia de análise Orientada a Objetos e a linguagem modelagem de dados UML (Unified Modeling Language). As ferramentas Astah Community e DBDesigner 4Fork serão utilizadas para a elaboração dos diagramas.

3.2. APLICAÇÃO

Para o desenvolvimento da aplicação será utilizada a IDE *Eclipse Kepler for Java Enterprise Edition*, juntamente com a linguagem de programação Java na versão 7 na construção de regras de negócio, acesso a dados seguindo o padrão Orientação a Objetos e *Design Patterns*. O framework Java Server Faces 2.2 e a tecnologia XHTML serão utilizados na construção da interface do usuário por facilitar o gerenciamento de protocolos WEB. O framework Hibernate na versão 4.2.6 auxiliará na persistência e recuperação de dados armazenados em Banco de Dados.

3.3. BANCO DE DADOS

O Banco de Dados responsável por armazenar de forma segura e eficiente os dados gerados pela aplicação será o MySQL 5.1.48, por ser um Banco de Dados Relacional e oferecer suporte à SQL e PLSQL e para manutenção será utilizada a ferramenta de Manutenção de Banco de Dados NaviCat 9.0.10.

3.4. RELATÓRIOS

Para o desenvolvimento dos relatórios da aplicação será utilizada a ferramenta Ireport 5.0 da JasperReports, por oferecer suporte a linguagem e de programação Java, e elaboração de relatórios de forma eficiente no contexto WEB.

4. ORIENTAÇÃO A OBJETOS

Orientação a Objetos é um padrão de programação presente em muitas das linguagens utilizadas atualmente, com o objetivo de oferecer mais simplicidade, organização, reaproveitamento de código entre outros benefícios. Este padrão busca representar na programação objetos do mundo real, e possuem alguns conceitos que devem ser utilizados, como Classes, Objetos, Encapsulamento, Herança, Polimorfismo dentre outros.

4.1. CLASSES

Classes são estruturas das linguagens de programação orientada a objetos para definir o modelo que se deseja programar de forma abstrata, ou seja, é onde são especificadas todas as características e ações que o modelo desejado pode conter (Rafael Santos, 2003).

4.2. OBJETOS

Objetos são instancias das Classes programadas, ou seja, eles são a materialização das Classes tornando assim possível a manipulação dos atributos (características) e métodos (ações) de uma determinada Classe (Rafael Santos, 2003).

4.3. ENCAPSULAMENTO

Encapsulamento é um conceito que tem objeto de proteger a integridade dos dados de uma Classe, ocultando seus atributos e tornando possível a manipulação destes dados somente através dos métodos especializados para cada atributo, evitando

assim a ocorrência de muitos erros oriundos da má manipulação dos dados (Rafael Santos, 2003).

4.4. HERANÇA

Herança na programação orientada a objetos é uma forma bem clara de reaproveitamento de código. Este conceito possibilita a criação uma Classe que utiliza uma Classe mãe como modelo, reaproveitando assim todos os atributos e métodos desta Classe mãe. Desta maneira é necessário apenas programar as características mais específicas da Classe filha (Rafael Santos, 2003).

4.5. POLIMORFISMO

Polimorfismo é a capacidade de manipulação de instancias de Classes que herdam de uma mesma Classe mãe de forma unificada, podendo programar métodos que recebam instancias da Classe mãe e estes métodos são capazes de processar instancias de qualquer classe que herde desta Classe mãe (Rafael Santos, 2003).

5. BANCO DE DADOS

Banco de dados é um conjunto de informações armazenadas que são inter-relacionadas, geralmente sobre uma área ou empreendimento específico, segundo Banco de Dados I (2014, p. 1):

Na realidade um Banco de Dados (BD) consiste em uma coleção ou conjunto de dados armazenados manualmente ou automatizado, que por sua vez são inter-relacionados, e que na maioria das vezes contém informações sobre um empreendimento particular (empresa específica).

O banco de dados possui grande importância no setor empresarial, devido ao recurso de armazenamento e tratamento dos dados da organização, oferecendo um *feedback* de grande importância para tomadas de decisões de qualquer organização (Poletto, 2014).

5.1. ENTIDADES

Entidades em Banco de Dados são objetos que existem no mundo real com uma identificação distinta e com um significado próprio. Estes dados são normalizados no Banco de Dados através de tabelas onde as colunas representam as características do objeto (Felipe Nery Rodrigues, Maurício Pereira, 2002).

5.2. RELACIONAMENTOS

Um relacionamento em Banco de Dados define o fato, acontecimento que liga dois objetos do mundo real. Estes relacionamentos podem ocorrer de três formas: Um-para-Um, Um-para-Muitos ou Muitos-para-Muitos.

5.2.1. UM-PARA-UM

Neste tipo de relacionamento cada elemento de uma entidade de Banco de Dados se relaciona com apenas um elemento de outra entidade. Exemplo deste relacionamento é que se existe uma entidade mulher e outra entidade homem, para o relacionamento entre estas entidades representar um casamento, uma mulher pode se relacionar com apenas um homem e vice-versa (Felipe Nery Rodrigues, Maurício Pereira, 2002).

5.2.2. UM-PARA-MUITOS

É o mais comum, denominado relacionamento básico entre duas entidades, onde um elemento da entidade A pode se relacionar com muitos elementos da entidade B. Este tipo de relacionamento possui características específicas em relação ao sentido de sua interpretação, pois é necessário analisar a cardinalidade da ocorrência do fato entre as duas entidades, ou seja, para que este relacionamento ocorra é necessário que a cardinalidade da entidade A para B seja um-para-muitos e a de B para A seja um-para-um (Felipe Nery Rodrigues, Maurício Pereira, 2002).

5.2.3. MUITOS-PARA-MUITOS

Este relacionamento ocorre quando a uma entidade A se relaciona com muitos elementos da entidade B, e a entidade B também se relaciona com muitos elementos da entidade A, ou seja, a cardinalidade de A para B é de um-para-muitos e a cardinalidade de B é também de um-para-muitos. Este relacionamento possui um aspecto peculiar, pois ele possui atributos, isto quer dizer que o relacionamento contém dados inerentes ao fato e não as entidades (Felipe Nery Rodrigues, Maurício Pereira, 2002).

6. DESCRIÇÃO DO AMBIENTE DE DESENVOLVIMENTO

Nesta seção do trabalho serão descritas as ferramentas e linguagens utilizadas no desenvolvimento do software, abordando os pontos positivos de cada uma e sua finalidade.

6.1. LINGUAGENS UTILIZADAS

6.1.1. JAVA

A linguagem de programação Java é sofisticada, robusta, segura e de alto desempenho. Iniciada num projeto em 1991 pela Sun Microsystems, mas apresentada formalmente em 1995, oferece uma infinidade de bibliotecas que auxiliam no desenvolvimento das aplicações conhecidas como Java API (Application Programming Interfaces), tornando o desenvolvimento mais rápido melhorando a produtividade (Deitel, 2009). Por ser uma linguagem orientada a objetos, torna possível o reuso de códigos e possui característica de portabilidade entre plataformas de sistemas operacionais devido à forma de compilação, gerando bytecodes executados pela JVM (Java Virtual Machine).

6.1.2. HTML

A linguagem de marcação HTML (HyperText Mark-up Language) foi criada por Tim Berners-Lee na década de 90, com a finalidade de se obter acesso e troca de informações entre diferentes universidades. O HTML é utilizado na programação de páginas Web, por meio de marcações ou *tags*, utilizados para formatar o conteúdo da página. Atualmente a versão mais recente do HTML é o HTML5 inicialmente desenvolvido pela WHATWG e depois desenvolvido juntamente a W3C, que trouxe

novas *tags*, melhores recursos, mais simplicidade e semântica ao código HTML (Eis, Ferreira, 2012).

6.1.3. CSS

A linguagem CSS (Cascading Style Sheets) é utilizada para formatar as informações da página HTML. O CSS torna possível alterar todas as características de estilos de qualquer *tag* do documento HTML, podendo ser implementado tanto dentro do arquivo HTML através da *tag* `<style></style>`, como fora do arquivo HTML fazendo referencia ao arquivo CSS.

6.1.4. UML

A linguagem de modelagem UML (Unified Modeling Language) surgiu a partir de três métodos de modelagem, o método de Booch, o método OMT (Object Modeling Technique) e o método de Jacobson (Guedes, 2011). A UML propõe a elaboração de diagramas para melhor compreensão do software a ser desenvolvido. Esta linguagem é utilizada na documentação do levantamento e análise de requisitos, narrativas, diagramas de casos de uso, atividade, sequência, dentre outros, fornecendo maneiras eficientes e ágeis na elaboração de projetos principalmente do ramo de software.

6.2. FERRAMENTAS DE DESENVOLVIMENTO

6.2.1. ECLIPSE

O Eclipse foi criado pela IBM em 2001 com apoio de algumas empresas do ramo de software. A ferramenta Eclipse é Open Source e possui hoje uma grande comunidade. O Eclipse é uma poderosa IDE que oferece suporte a linguagem Java, e implementa diversos frameworks que auxiliam no desenvolvimento e na produtividade. Possui também o padrão de desenvolvimento em camadas e oferece interface dinâmica para melhor se ajusta no tipo de projeto a ser desenvolvido (Disponível em: < <http://www.eclipse.org/org/> >. Acesso em: 12 jan. 2014).

6.2.2. MYSQL

O Banco de Dados Relacional MySQL iniciado pelos desenvolvedores David Axmark, Allan Larsson e Michael Widenius, na década de 90, foi projetado para aplicações de pequeno e médio porte, mas atualmente é o banco de dados *Open Source* de maior capacidade no mercado. O MySQL oferece alto poder de execução e armazenamento, compatibilidade com diversos sistemas operacionais e linguagens de programação, segurança no tráfego de dados, dentro outras características que o torna uma poderosa ferramenta *Open Source* para armazenamento e manutenção de dados.

6.2.3. NAVICAT

Navicat é uma ferramenta para administração de banco de dados multi-conexões, que oferece suporte a diversos bancos de dados como MySQL, SQL Server, Oracle. Oferece recursos de criação de bases de dados, tabelas, manutenção e consulta de

dados, geração de diagrama ER, UML, dentre outros que auxiliam a manutenção dos dados de forma segura e confiável. Esta ferramenta permite transferência de dados entre diferentes SGBDs ou em documento SQL. (Disponível em: <<http://www.navicat.com/products/navicat-premium>>. Acesso em: 11 fev. 2014).

6.3. FRAMEWORKS

6.3.1. HIBERNATE

Hibernate é um poderoso framework Open Source de persistência de dados Objeto-Relacional para aplicação Java. Esta ferramenta facilita o desenvolvimento de aplicações orientadas a objeto cuidando da geração dos códigos SQL através da Linguagem Java, podendo gerar estes código para diversos bancos de dados diferentes, oferecendo a possibilidade de portabilidade entre banco de dados e mantendo intactas as classes Java, bastando informar o dialeto que o Hibernate deve obedecer.

6.3.2. JSF (JAVA SERVER FACES)

JSF (Java Server Faces) é um framework Open Source para desenvolvimento de aplicações Java Web que possui várias implementações. Com ele o desenvolvedor não precisa se preocupar com códigos Javascript e protocolos HTML, bastando apenas adicionar componentes como tabelas, caixas de texto, etc., que são renderizados e exibidos em HTML. O JSF utiliza o padrão de projeto MVC onde a camada de visualização baseia-se em Facelets e nas classes Managed Beans, que fazem a ligação entre a interface HTML e a aplicação Java (Caleum, Curso Laboratório com Teste, JSF e Design Patters).

7. ANÁLISE

Esta seção aborda as ferramentas utilizadas na análise do projeto, bem como toda estrutura de diagramas e descrição de funcionalidades do software construída a partir do levantamento de requisitos.

7.1. FERRAMENTAS DE ANÁLISE

7.1.1. DBDESIGNER 4

O DBDesigner é uma ferramenta case para modelagem de banco de dados. Desenvolvida pela empresa Fabulous Force Database Tools, sua versão mais atual é o DBDesigner Fork. Esta ferramenta oferece suporte ao banco de dados MySQL, e possui recursos de geração de Modelo E-R, bem como exportação do código SQL para o banco de dados MySQL, dentre outros.

7.1.2 ASTAH COMMUNITY

Astah Community é uma ferramenta para modelagem UML que oferece suporte a UML 2.1 e possui grande variedade de diagramas como Caso de Uso, Sequência e de Atividades. Também oferecer o recurso de exportação dos diagramas nos formatos de imagens PNG e JPEG e geração de códigos Java e SQL a partir dos diagramas, sendo assim uma ferramenta de grande utilidade na fase de implementação de projetos de software (Disponível em: <<http://astah.net/editions>>. Acesso em: 12 fev. 2014).

7.1.3. MICROSOFT VISIO 2013

O Microsoft Visio 2013 é uma ferramenta que oferece recursos para elaboração de diagramas de atividade, diagrama de sequência, diagrama de redes, cronogramas, entre outros diagramas. Oferece suporte para padrões atualizados de diagramação, incluindo UML 2.4 e BPMN 2.0. (Disponível em: <<http://office.microsoft.com/pt-br/visio/visio-pro-for-office-365-online-diagram-software-FX103791920.aspx>>.

Acessado em: 11 de mar. 2014).

7.2. LEVANTAMENTO E ESPECIFICAÇÃO DOS REQUISITOS

O levantamento dos requisitos é uma fase importante para o desenvolvimento do software. Neste processo são analisadas todas as funcionalidades que o programa deve possuir para suprir as necessidades para o fim que este foi proposto a servir.

7.2.1. DESCRIÇÃO DOS OBJETIVOS

Manter (Projetos, Colaboradores, Tarefas, Tópicos, Cargos, Equipes). Iniciar e Concluir Tarefas e Projetos. Exibir Horas Trabalhadas pelos Colaboradores. Mural de Comentários (Equipe, Empresa). Relatórios (Colaboradores, Equipes, Projetos, Tarefas).

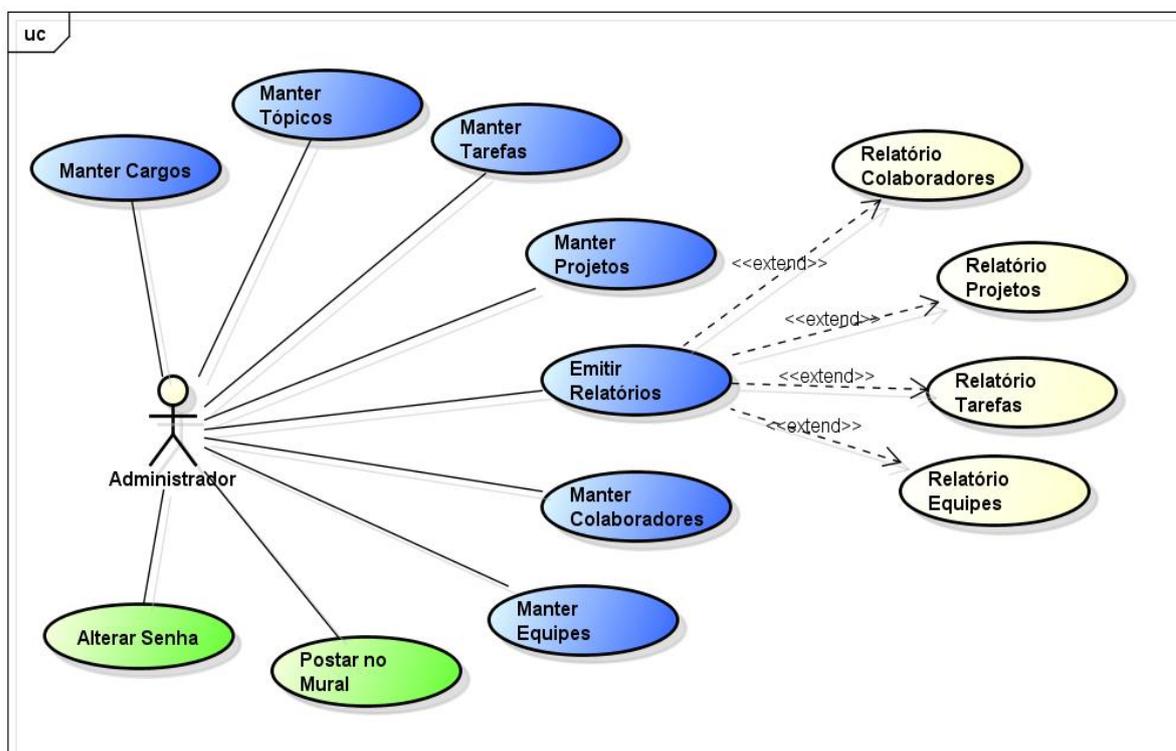
7.2.2. RESULTADOS ESPERADOS

O sistema irá fornecer ao usuário, praticidade e agilidade no controle dos funcionários e das designações de tarefas, bem como dados e relatórios para a análise da produtividade dos funcionários e o andamento dos projetos.

7.3. DIAGRAMAS UML

7.3.1. CASOS DE USO ADMINISTRADOR

Caso de Uso que representa as funcionalidades incorporadas ao usuário com permissão de administrador do sistema. O administrador é o único responsável pela emissão de relatórios e administrar todos os cadastros de Cargos, Colaboradores, Equipes, Projetos, Tarefas e Tópicos.

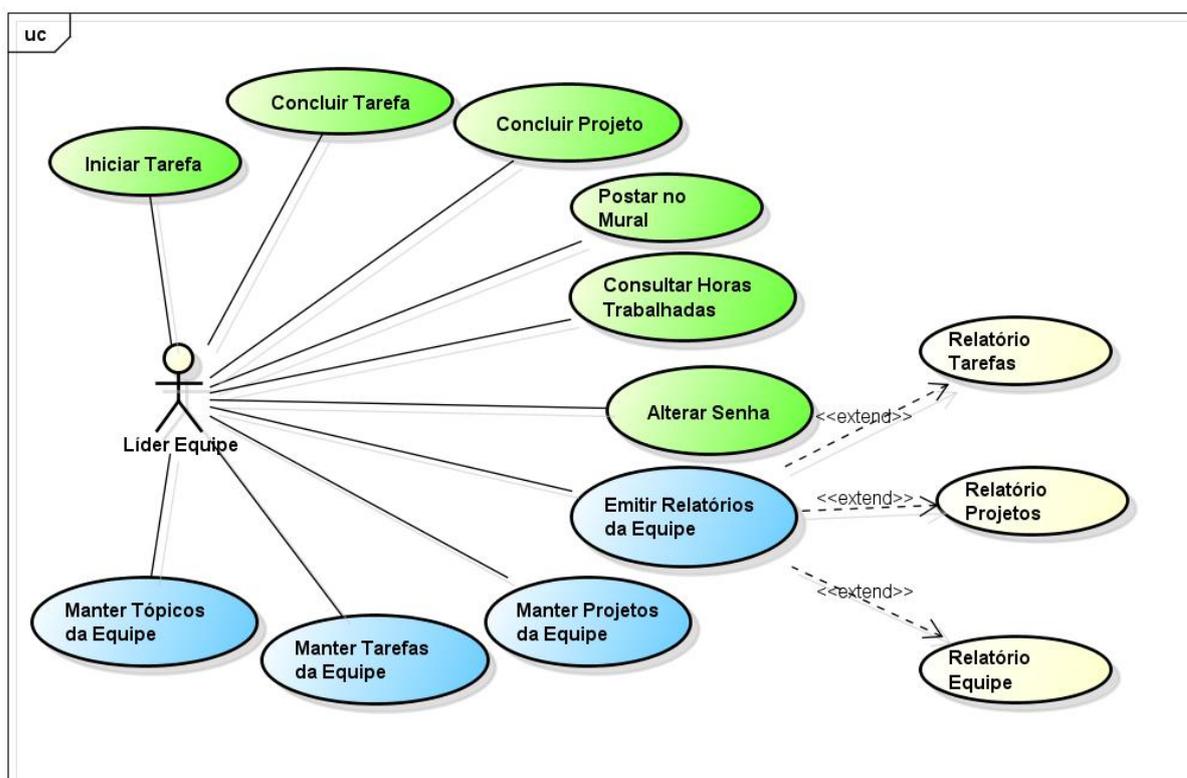


powered by Astah

Figura 1. Diagrama de Caso de Uso Administrador

7.3.2. CASOS DE USO LÍDER DE EQUIPE

Caso de Uso que representa as funcionalidades incorporadas ao usuário com permissão de líder de equipe. O líder de equipe é o responsável pelos projetos da equipe, sendo o único com permissão para concluir projetos e com capacidade para administrar os Projetos, Tarefas e Tópicos e Emitir Relatórios de sua equipe.

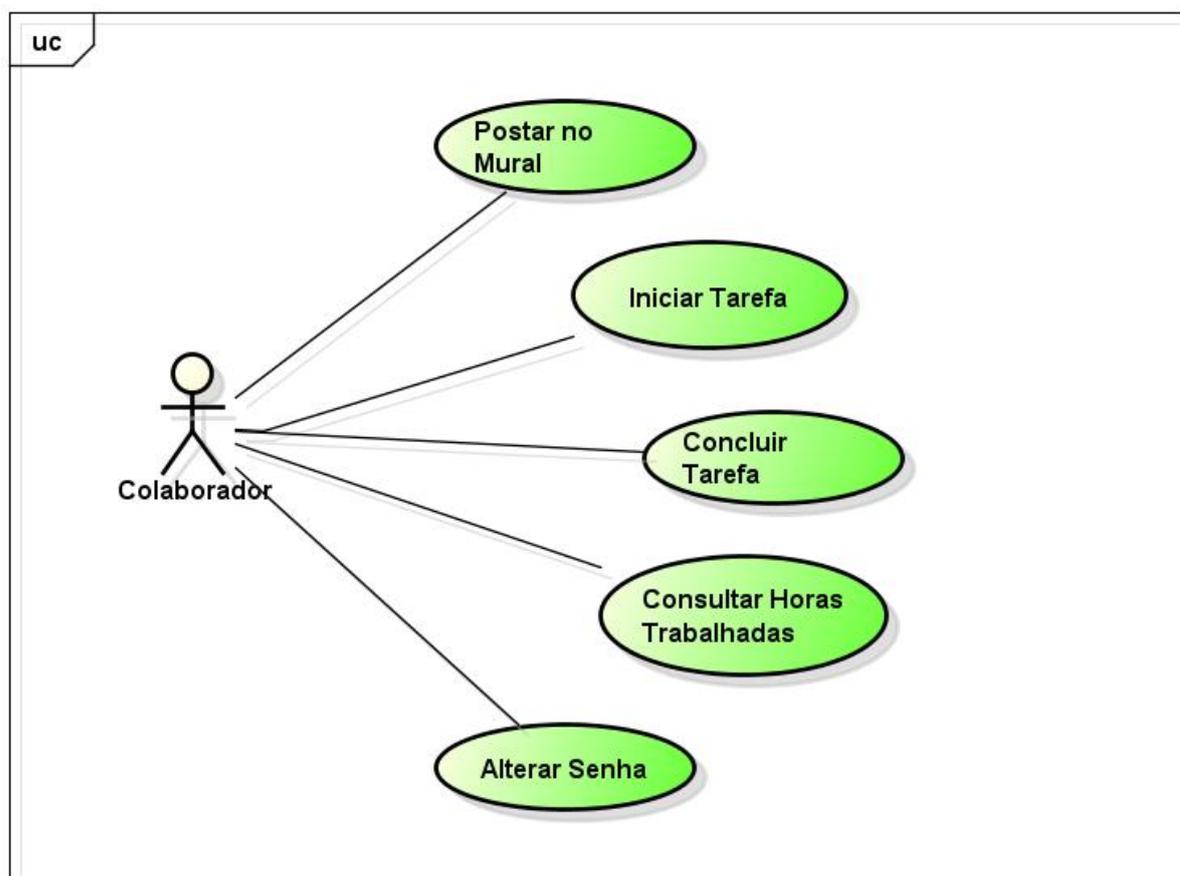


powered by Astah

Figura 2. Diagrama de Caso de Uso Líder de Equipe

7.3.3. CASOS DE USO COLABORADOR

Caso de Uso que representa as funcionalidades incorporadas ao usuário com permissão de colaborador. O colaborador é responsável pelo andamento das tarefas a que lhe foi designada.



powered by Astah

Figura 3. Diagrama de Caso de Uso Colaborador

7.3.4. ESPECIFICAÇÃO CASO DE USO MANTER CARGOS

Manter Cargos	
Ator	Administrador
Descrição	Este caso de uso descreve as etapas para o usuário manter os cargos.
Pré-Requisito	Efetuar login no sistema
Fluxo Principal	<ol style="list-style-type: none"> 1. O ator inicia o caso de uso selecionando cadastro de cargos. 2. O sistema oferece opções de manutenção. 3. O ator deseja incluir um novo cargo. [A1, A2] 4. O sistema oferece a tela para inclusão. 5. O ator entra com as informações e seleciona salvar.
Fluxo Alternativo A1: Alteração	<ol style="list-style-type: none"> 3. O ator deseja alterar um cargo. 4. O sistema oferece o cargo para alteração. 5. O ator entra com as informações e seleciona salvar.
Fluxo Alternativo A2: Exclusão	<ol style="list-style-type: none"> 3. O ator deseja excluir um cargo. 4. O sistema oferece o cargo para a exclusão. 5. O sistema solicita a confirmação da exclusão. 6. O ator confirma a exclusão.

Tabela 1. Especificação Caso de Uso Manter Cargos

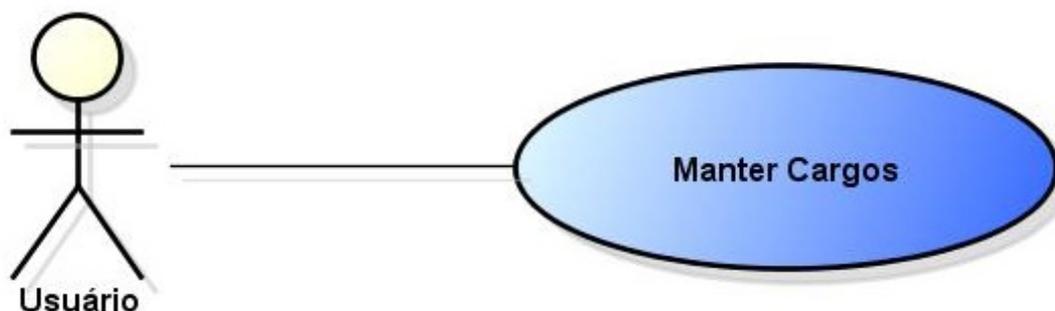


Figura 4. Diagrama de Caso de Uso Manter Cargos

7.3.5. ESPECIFICAÇÃO CASO DE USO MANTER COLABORADORES

Manter Colaboradores	
Ator	Administrador
Descrição	Este caso de uso descreve as etapas para o usuário manter colaboradores.
Pré-Requisito	Efetuar login no sistema.
Fluxo Principal	<ol style="list-style-type: none"> 1. O ator inicia o caso de uso selecionando cadastro de colaboradores. 2. O sistema oferece opções de manutenção. 3. O ator deseja incluir um novo colaborador. [A1, A2] 4. O sistema oferece a tela para inclusão. 5. O ator entra com as informações e seleciona salvar.
Fluxo Alternativo A1: Alteração	<ol style="list-style-type: none"> 3. O ator deseja alterar um colaborador. 4. O sistema oferece o funcionário para alteração. 5. O ator entra com as informações e seleciona salvar.
Fluxo Alternativo A2: Exclusão	<ol style="list-style-type: none"> 3. O ator deseja excluir um colaborador. 4. O sistema oferece o colaborador para a exclusão. 5. O sistema solicita a confirmação da exclusão. 6. O ator confirma a exclusão.

Tabela 2. Especificação Caso de Uso Manter Colaboradores

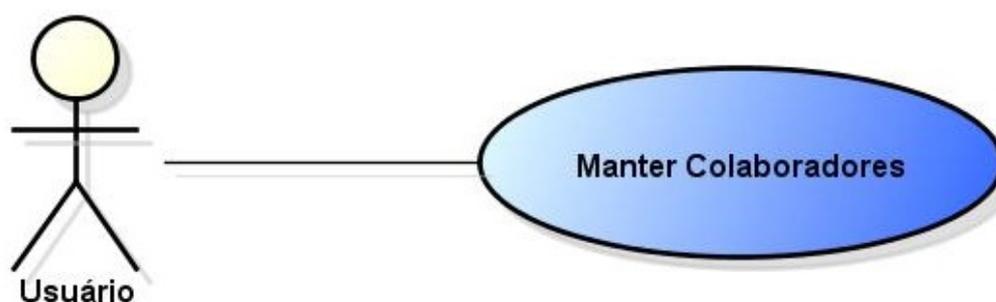


Figura 5. Diagrama de Caso de Uso Manter Colaboradores

7.3.6. ESPECIFICAÇÃO CASO DE USO MANTER PROJETOS

Manter Projetos	
Ator	Administrador, Líder de Equipe
Descrição	Este caso de uso descreve as etapas para o usuário manter os projetos.
Pré-Requisito	Efetuar login no sistema.
Fluxo Principal	<ol style="list-style-type: none"> 1. O ator inicia o caso de uso selecionando cadastro de projetos. 2. O sistema oferece opções de manutenção. 3. O ator deseja incluir um novo projeto. [A1, A2] 4. O sistema oferece a tela para inclusão. 5. O ator entra com as informações e seleciona salvar.
Fluxo Alternativo A1: Alteração	<ol style="list-style-type: none"> 3. O ator deseja alterar um projeto. 4. O sistema oferece o projeto para alteração. 5. O ator entra com as informações e seleciona salvar.
Fluxo Alternativo A2: Exclusão	<ol style="list-style-type: none"> 3. O ator deseja excluir um projeto. 4. O sistema oferece o projeto para a exclusão. 5. O sistema solicita a confirmação da exclusão. 6. O ator confirma a exclusão.

Tabela 3. Especificação Caso de Uso Manter Projetos

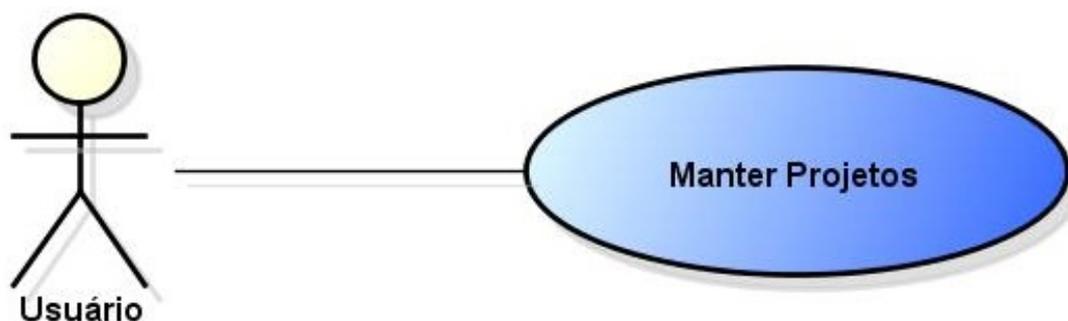


Figura 6. Diagrama de Caso de Uso Manter Projetos

7.3.7. ESPECIFICAÇÃO CASO DE USO MANTER TAREFAS

Manter Tarefas	
Ator	Administrador, Líder de Equipe
Descrição	Este caso de uso descreve as etapas para o usuário manter as tarefas.
Pré-Requisito	Efetuar login no sistema.
Fluxo Principal	<ol style="list-style-type: none"> 1. O ator inicia o caso de uso selecionando cadastro de tarefas. 2. O sistema oferece opções de manutenção. 3. O ator deseja incluir uma nova tarefa. [A1, A2] 4. O sistema oferece a tela para inclusão. 5. O ator entra com as informações e seleciona salvar. 6. Trigger insert_tarefa disparada.
Fluxo Alternativo A1: Alteração	<ol style="list-style-type: none"> 3. O ator deseja alterar uma tarefa. 4. O sistema oferece a tarefa para alteração. 5. O ator entra com as informações e seleciona salvar. 6. Trigger update_tarefa disparada.
Fluxo Alternativo A2: Exclusão	<ol style="list-style-type: none"> 3. O ator deseja excluir uma tarefa. 4. O sistema oferece a tarefa para a exclusão. 5. O sistema solicita a confirmação da exclusão. 6. O ator confirma a exclusão. 7. Trigger delete_tarefa disparada.

Tabela 4. Especificação Caso de Uso Manter Tarefas

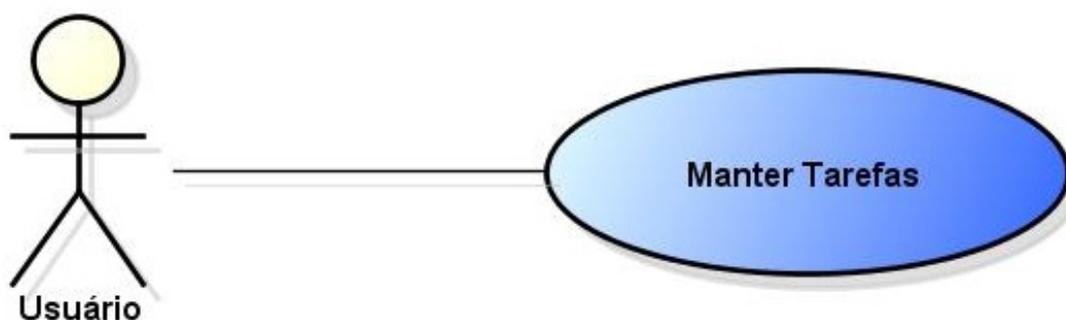


Figura 7. Diagrama de Caso de Uso Manter Tarefas

7.3.8. ESPECIFICAÇÃO CASO DE USO MANTER TÓPICOS

Manter Tópicos	
Ator	Administrador, Líder de Equipe.
Descrição	Este caso de uso descreve as etapas para o usuário manter os tópicos.
Pré-Requisito	Efetuar login no sistema.
Fluxo Principal	<ol style="list-style-type: none"> 1. O ator inicia o caso de uso selecionando cadastro de tópicos. 2. O sistema oferece opções de manutenção. 3. O ator deseja incluir um novo tópico. [A1, A2] 4. O sistema oferece a tela para inclusão. 5. O ator entra com as informações e seleciona salvar.
Fluxo Alternativo A1: Alteração	<ol style="list-style-type: none"> 3. O ator deseja alterar um tópico. 4. O sistema oferece o tópico para alteração. 5. O ator entra com as informações e seleciona salvar. 6. Trigger update_topico disparada.
Fluxo Alternativo A2: Exclusão	<ol style="list-style-type: none"> 3. O ator deseja excluir um tópico. 4. O sistema oferece o tópico para a exclusão. 5. O sistema solicita a confirmação da exclusão. 6. O ator confirma a exclusão.

Tabela 5. Especificação Caso de Uso Manter Tópicos

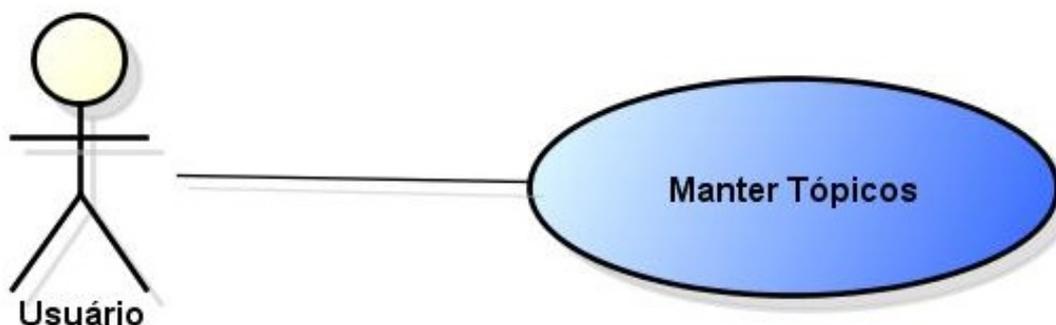


Figura 8. Diagrama Caso de Uso Manter Tópicos

7.3.9. ESPECIFICAÇÃO CASO DE USO MANTER EQUIPE

Manter Equipes	
Ator	Administrador.
Descrição	Este caso de uso descreve as etapas para o usuário manter as equipes.
Pré-Requisito	Efetuar login no sistema.
Fluxo Principal	<ol style="list-style-type: none"> 1. O ator inicia o caso de uso selecionando cadastro de equipes. 2. O sistema oferece opções de manutenção. 3. O ator deseja incluir uma nova equipe. [A1, A2] 4. O sistema oferece a tela para inclusão. 5. O ator entra com as informações e seleciona salvar. 6. Trigger insert_equipe disparada.
Fluxo Alternativo A1: Alteração	<ol style="list-style-type: none"> 3. O ator deseja alterar uma equipe. 4. O sistema oferece a equipe para alteração. 5. O ator entra com as informações e seleciona salvar.
Fluxo Alternativo A2: Exclusão	<ol style="list-style-type: none"> 3. O ator deseja excluir uma equipe. 4. O sistema oferece a equipe para a exclusão. 5. O sistema solicita a confirmação da exclusão. 6. O ator confirma a exclusão. 7. Trigger delete_equipe disparada.

Tabela 6. Especificação Caso de Uso Manter Equipes

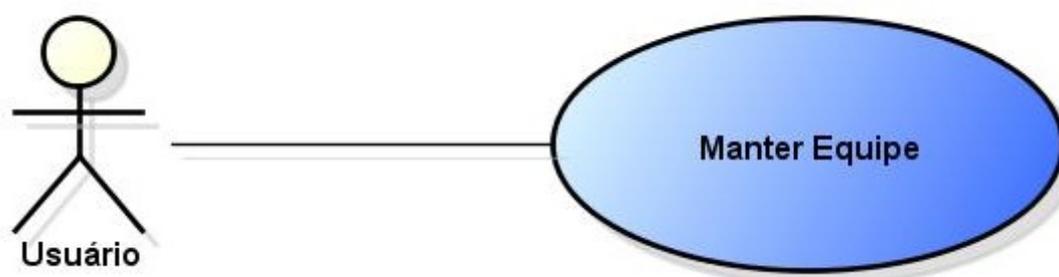


Figura 9. Diagrama Caso de Uso Manter Equipes

7.3.10. ESPECIFICAÇÃO CASO DE USO EMITIR RELATÓRIOS

Emitir Relatórios	
Ator	Administrador, Líder de Equipe
Descrição	Este caso de uso descreve as etapas para o usuário emitir relatórios.
Pré-Requisito	Efetuar login no sistema.
Fluxo Principal	<ol style="list-style-type: none"> 1. O ator inicia o caso de uso selecionando relatórios. 2. O sistema oferece opções de relatórios. 3. O ator seleciona o relatório que deseja gerar e entra com as informações necessárias. 4. O sistema exibe o relatório desejado.

Tabela 7. Especificação Caso de Uso Emitir Relatórios

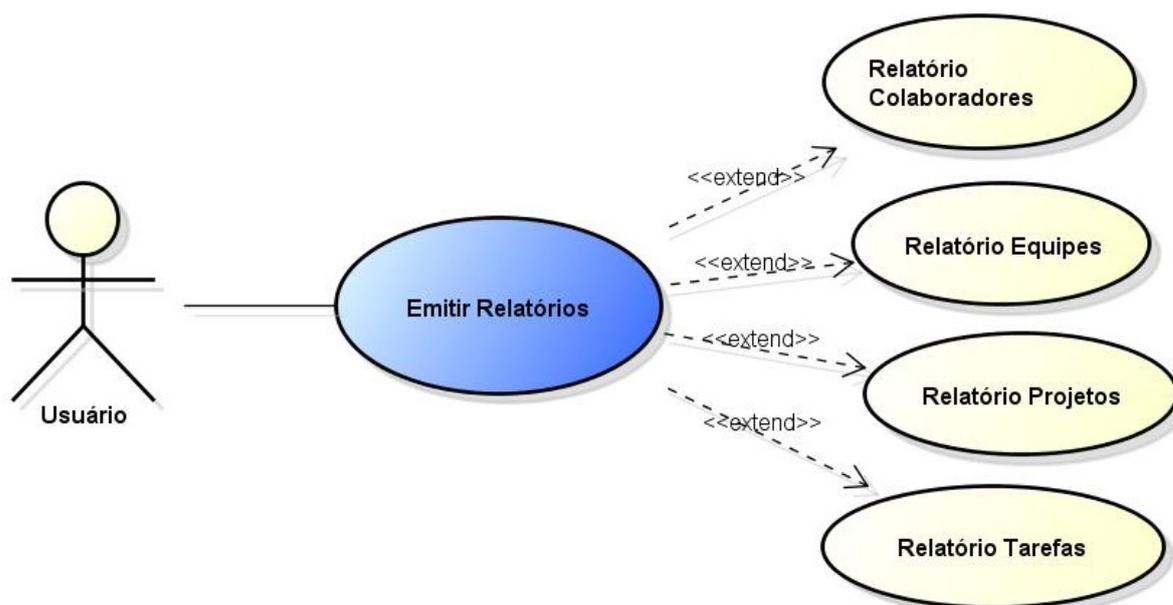


Figura 10. Diagrama de Caso de Uso Emitir Relatórios

7.3.11. ESPECIFICAÇÃO CASO DE USO POSTAR NO MURAL

Postar no Mural	
Ator Principal	Administrador, Líder de Equipe, Colaborador.
Descrição	Este caso de uso descreve as etapas para o usuário postar mensagens no mural.
Pré-Requisito	Efetuar login no sistema.
Fluxo Principal	<ol style="list-style-type: none"> 1. O ator inicia o caso de uso selecionado a opção Utilidades. 2. O sistema oferece tela de utilidades. 3. O ator selecionado mural da empresa. 4. O sistema exibe a tela do mural da empresa. 5. O ator inserir um texto e envia a mensagem.
Fluxo Alternativo A1: Mural da Equipe	<ol style="list-style-type: none"> 3. O ator selecionado mural da equipe. 4. O sistema exibe a tela do mural da empresa. 5. O ator inserir um texto e envia a mensagem.

Tabela 8. Especificação Caso de Uso Postar no Mural

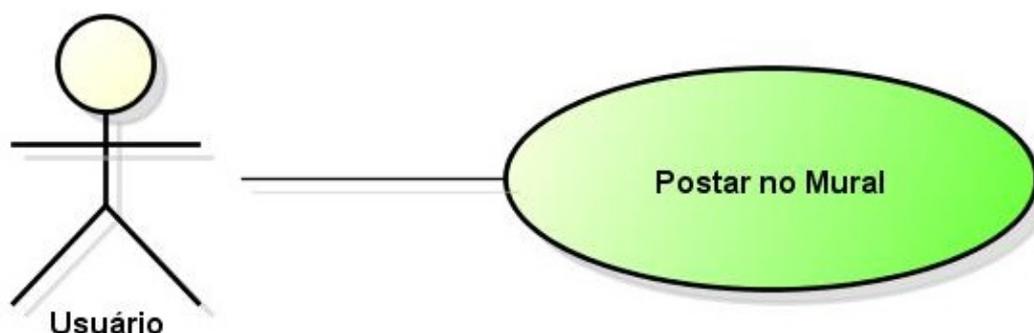


Figura 11. Diagrama de Caso de Uso Postar no Mural

7.3.12. ESPECIFICAÇÃO CASO DE USO CONSULTAR HORAS TRABALHADAS

Consultar Horas Trabalhadas	
Ator Principal	Líder de Equipe, Colaborador.
Descrição	Este caso de uso descreve as etapas para o usuário consultar as horas trabalhadas
Pré-Requisito	Efetuar login no sistema.
Fluxo Principal	<ol style="list-style-type: none"> 1. O ator inicia o caso de uso consultar horas trabalhadas. 2. O sistema oferece tela com quadro de horas trabalhadas. [A1] 3. O ator finaliza o caso de uso.
Fluxo Alternativo A1: Filtrar consulta	<ol style="list-style-type: none"> 3. O ator informa o projeto que deseja filtrar. 4. O sistema exibe as tarefas do usuário no projeto. 5. O ator seleciona a(s) tarefa(s). 6. O sistema exibe o quadro de horas atualizado.

Tabela 9. Especificação Caso de Uso Consultar Horas Trabalhadas

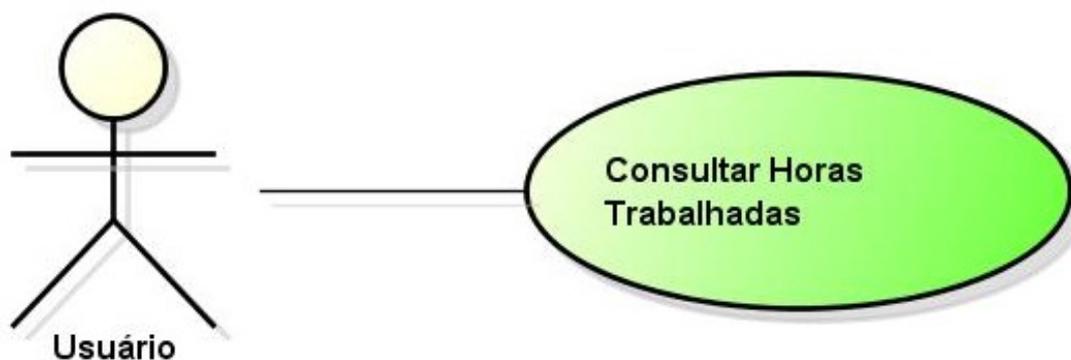


Figura 12. Diagrama de Caso de Uso Consultar Horas Trabalhadas

7.3.13. ESPECIFICAÇÃO CASO DE USO INICIAR TAREFA

Iniciar Tarefa	
Ator Principal	Líder de Equipe, Colaborador.
Descrição	Este caso de uso descreve as etapas para o usuário iniciar a tarefa.
Pré-Requisito	Efetuar login no sistema; Existir tarefa cadastrada para o usuário.
Fluxo Principal	1. O ator inicia o caso de uso iniciar tarefa selecionando a opção minhas tarefas. 2. O sistema oferece tela com quadro de tarefas pendentes. [A1] 3. O ator seleciona a tarefa e em seguida opção iniciar tarefa. 4. O sistema contabiliza o tempo ativo do caso de uso. [A2] 5. Trigger update_tarefa disparada.
Fluxo Alternativo A1: Ator seleciona Minhas Tarefas Em Andamento.	3. O ator seleciona a tarefa e em seguida opção iniciar tarefa. 4. O sistema contabiliza o tempo ativo do caso de uso. [A2] 5. Trigger update_tarefa disparada.
Fluxo Alternativo A2: Alterar status de um tópico	5. Ator seleciona o tópico e alterar o status. 6. O sistema informa que o tópico foi atualizado. 7. Trigger update_topico disparada.

Tabela 10. Especificação Caso de Uso Iniciar Tarefa

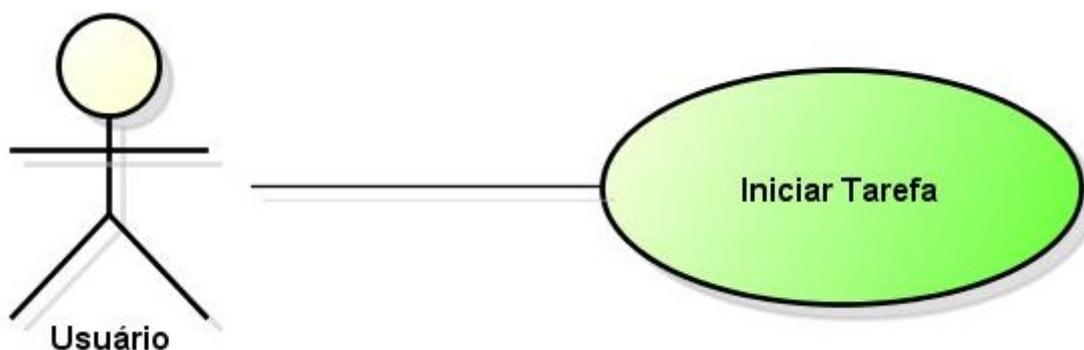


Figura 13. Diagrama de Caso de Uso Iniciar Tarefa

7.3.14. ESPECIFICAÇÃO CASO DE USO CONCLUIR TAREFA

Concluir Tarefa	
Ator Principal	Líder de Equipe, Colaborador.
Descrição	Este caso de uso descreve as etapas para o usuário concluir tarefas.
Pré-Requisito	Efetuar login no sistema; Existir tarefa cadastrada para o usuário.
Fluxo Principal	<ol style="list-style-type: none"> 1. O ator inicia o caso de uso concluir tarefa selecionando a opção minhas tarefas. 2. O sistema oferece tela com quadro de tarefas pendentes. 3. O ator seleciona opção tarefas em andamento. 4. O sistema oferece o quadro de tarefas em andamento. 5. O ator seleciona a opção concluir tarefa. [E1] 6. O sistema confirma a entrega da tarefa.
Fluxo Exceção E1: Tarefa não concluída	6. O sistema informa que a tarefa possui tópicos pendentes e cancela a operação.

Tabela 11. Especificação Caso de Uso Concluir Tarefa

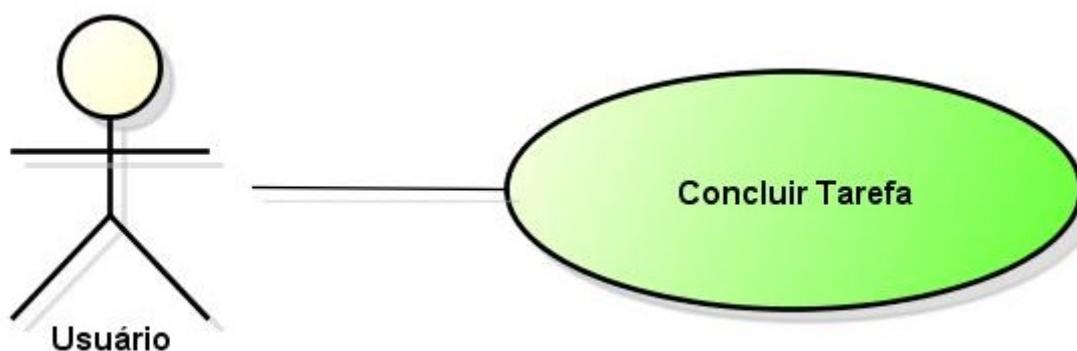


Figura 14. Caso de Uso Concluir Tarefa

7.3.15. ESPECIFICAÇÃO CASO DE USO CONCLUIR PROJETO

Entregar Projeto	
Ator Principal	Líder de Equipe
Descrição	Este caso de uso descreve as etapas para o usuário concluir projetos.
Pré-Requisito	Efetuar login no sistema; Existir projeto cadastrado para o usuário.
Fluxo Principal	<ol style="list-style-type: none"> 1. O ator inicia o caso de uso concluir tarefa selecionando a opção meus projetos. 2. O sistema oferece tela com quadro de projetos pendentes. 3. O ator seleciona opção projetos em andamento. 4. O sistema exibe o quadro de projetos em andamento. 5. O ator seleciona a opção concluir projeto. [E1] 6. O sistema confirma a entrega da tarefa
Fluxo Exceção E1: Projeto não concluído	6. O sistema informa que projeto não possui tarefas pendentes e cancela a operação.

Tabela 12. Especificação Caso de Uso Concluir Projeto

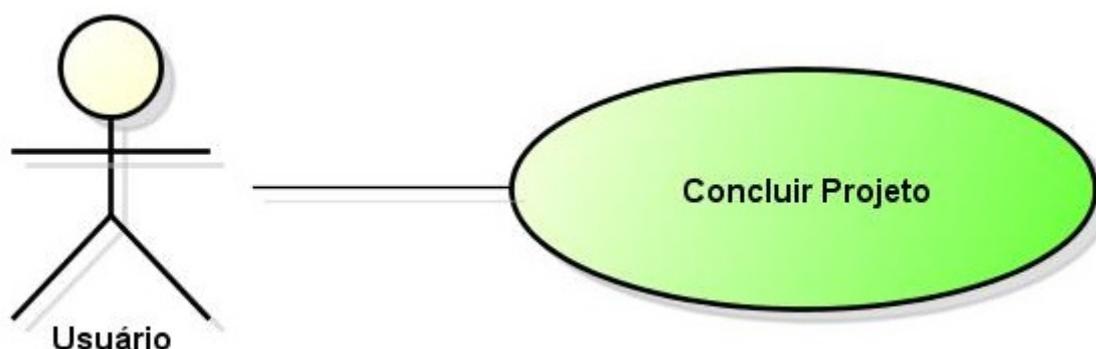
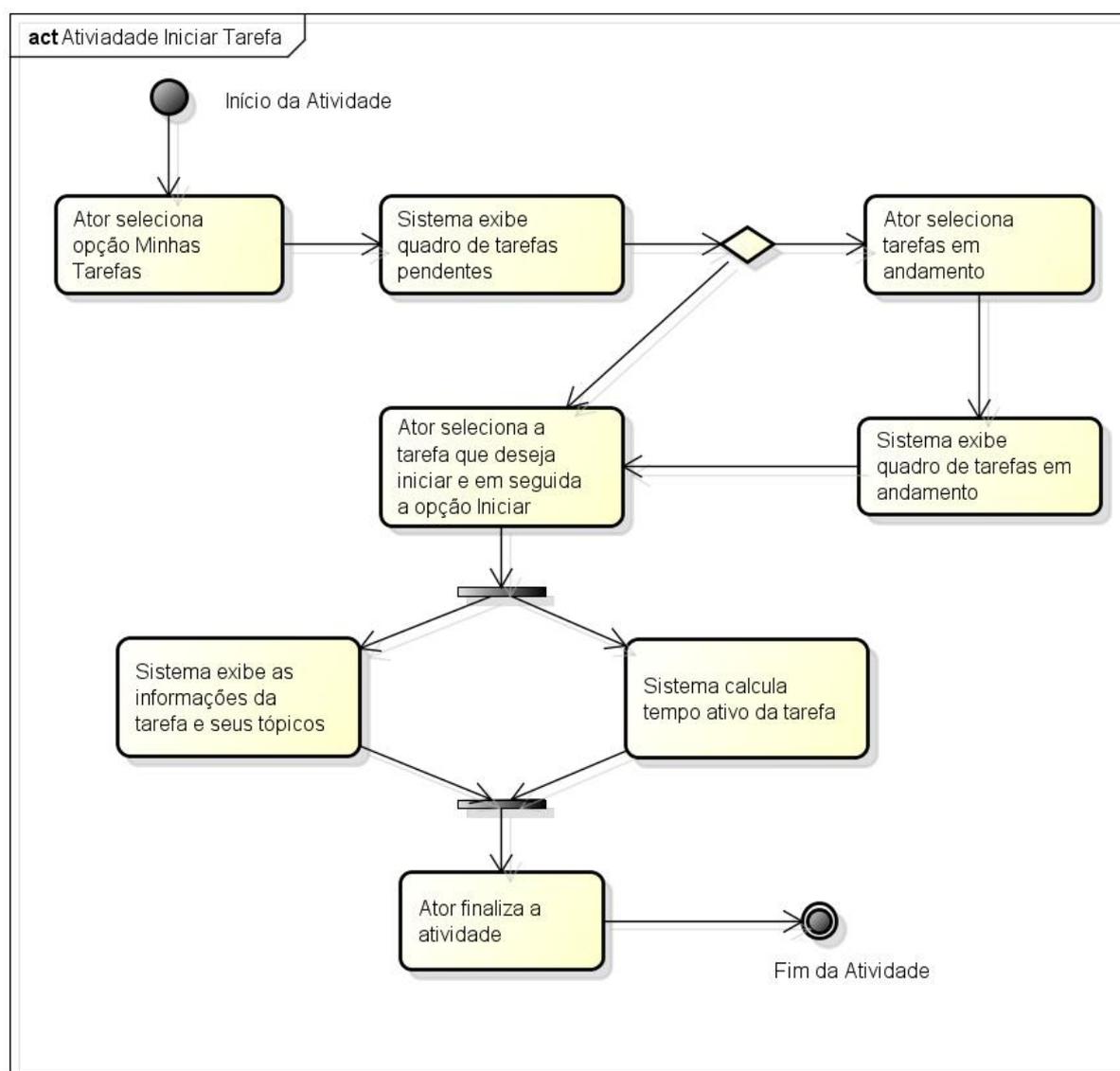


Figura 15. Caso de Uso Concluir Projeto

7.3.16. DIAGRAMA ATIVIDADE INICIAR TAREFA

O Diagrama de Atividade Iniciar Tarefa ilustra os passos para o usuário executar a atividade utilizando a interface do sistema, do início ao fim do processo, onde o usuário necessita iniciar uma tarefa pendente ou continuar uma tarefa que esteja em andamento.

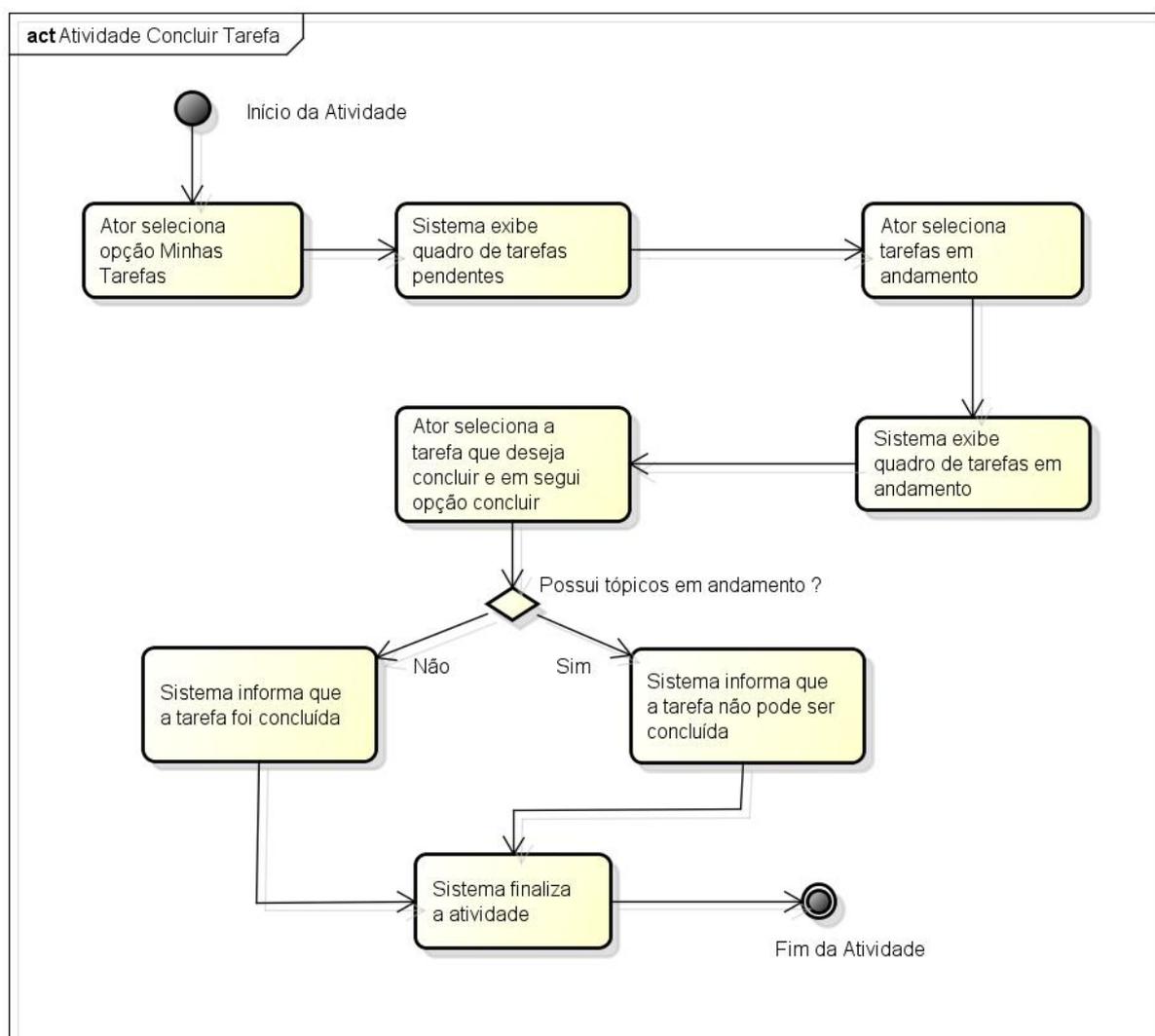


powered by Astah

Figura 16. Diagrama de Atividade Iniciar Tarefa

7.3.17. DIAGRAMA ATIVIDADE CONCLUIR TAREFA

O Diagrama de Atividade Concluir Tarefa ilustra os passos para o usuário executar a atividade utilizando a interface do sistema, do início ao fim do processo, onde o usuário necessita concluir uma tarefa que esteja em andamento.



powered by Astah

Figura 17. Diagrama de Atividade Concluir Tarefa

7.3.18. DIAGRAMA DE CLASSES

O Diagrama de Classes representa os modelos que gerenciam as principais informações do sistema e a maneira como eles se relacionam no paradigma orientado a objetos.

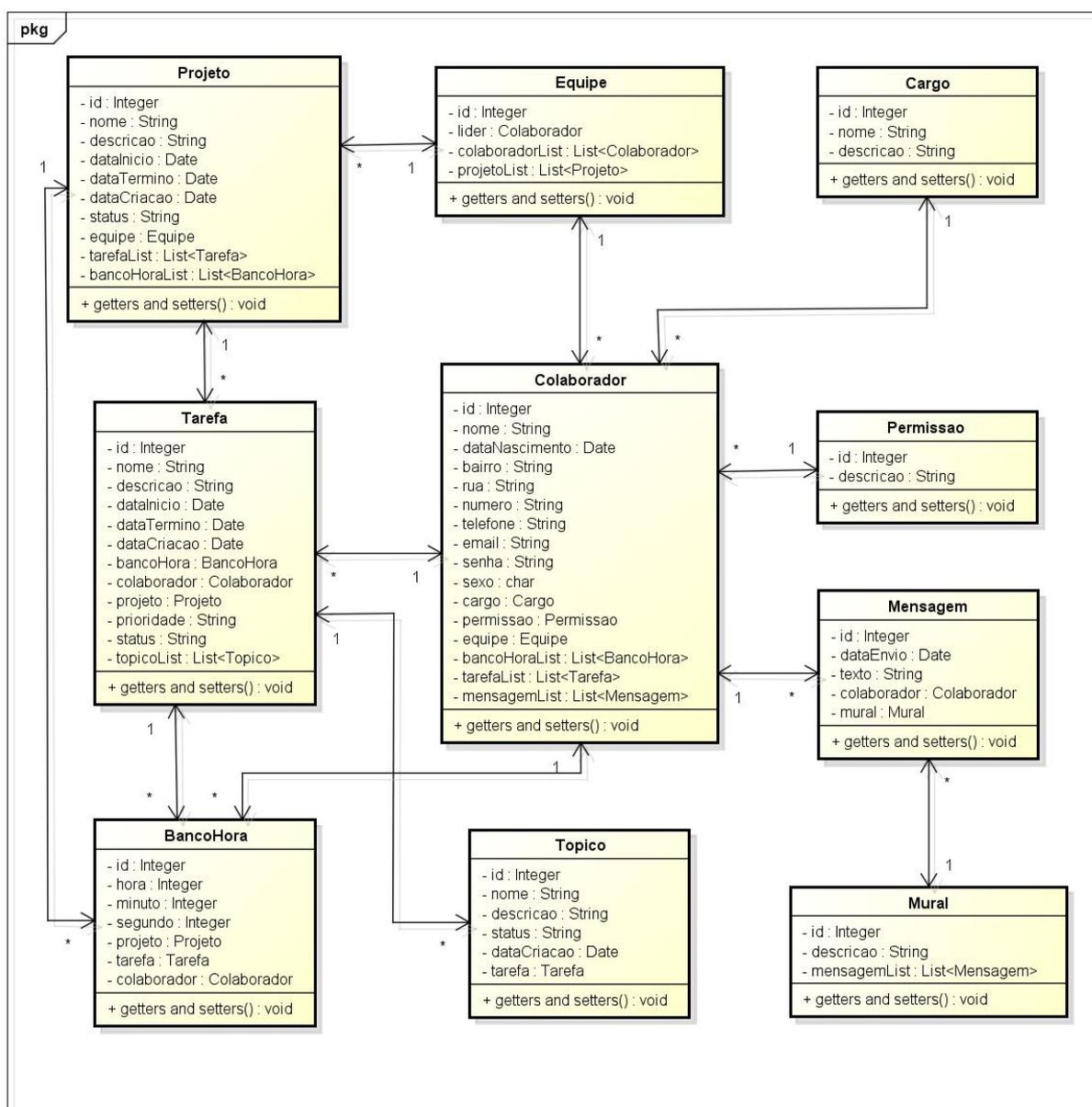


Figura 18. Diagrama de Classe

7.3.19. DIAGRAMA ER

O Diagrama de Entidade-Relacionamento representam a estrutura de tabelas construída no banco de dados para armazenar de forma segura e eficiente as informações geradas pelo software, e a maneira como essas tabelas se relacionam para garantir a integridade dos dados.

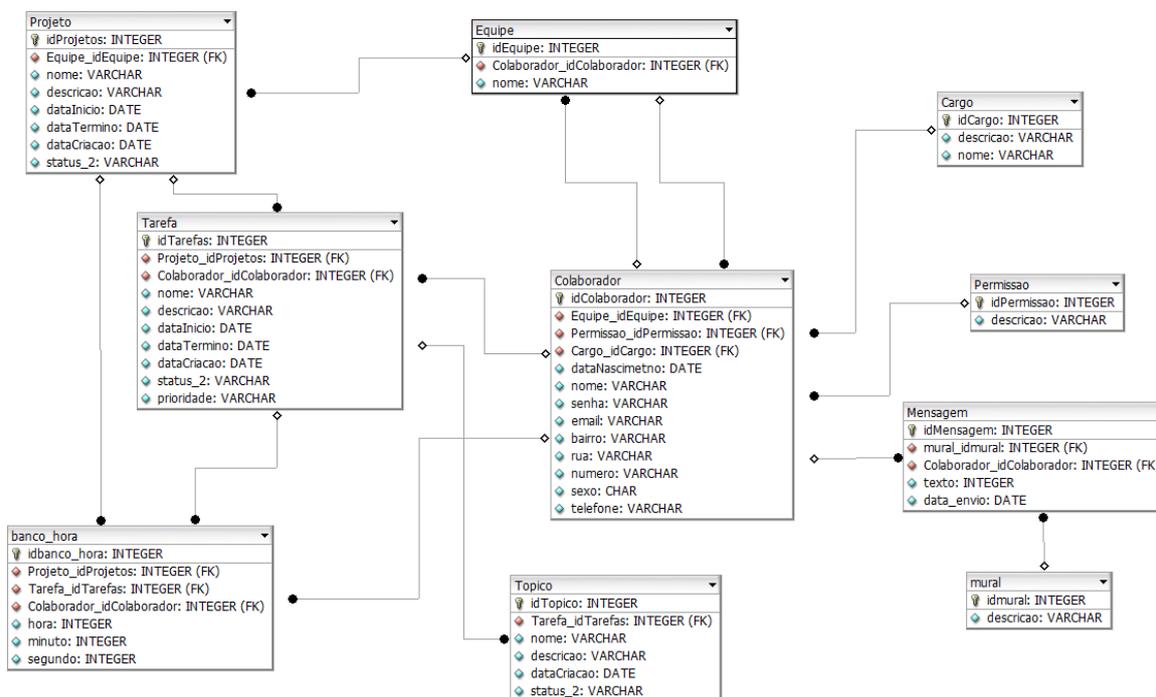


Figura 19. Diagrama Entidade-Relacionamento

7.3.20. WBS

O Diagrama *Work Breakdown Structure* representa os processos realizados no desenvolvimento deste projeto, descrevendo as atividades de análise, desenvolvimento e testes.

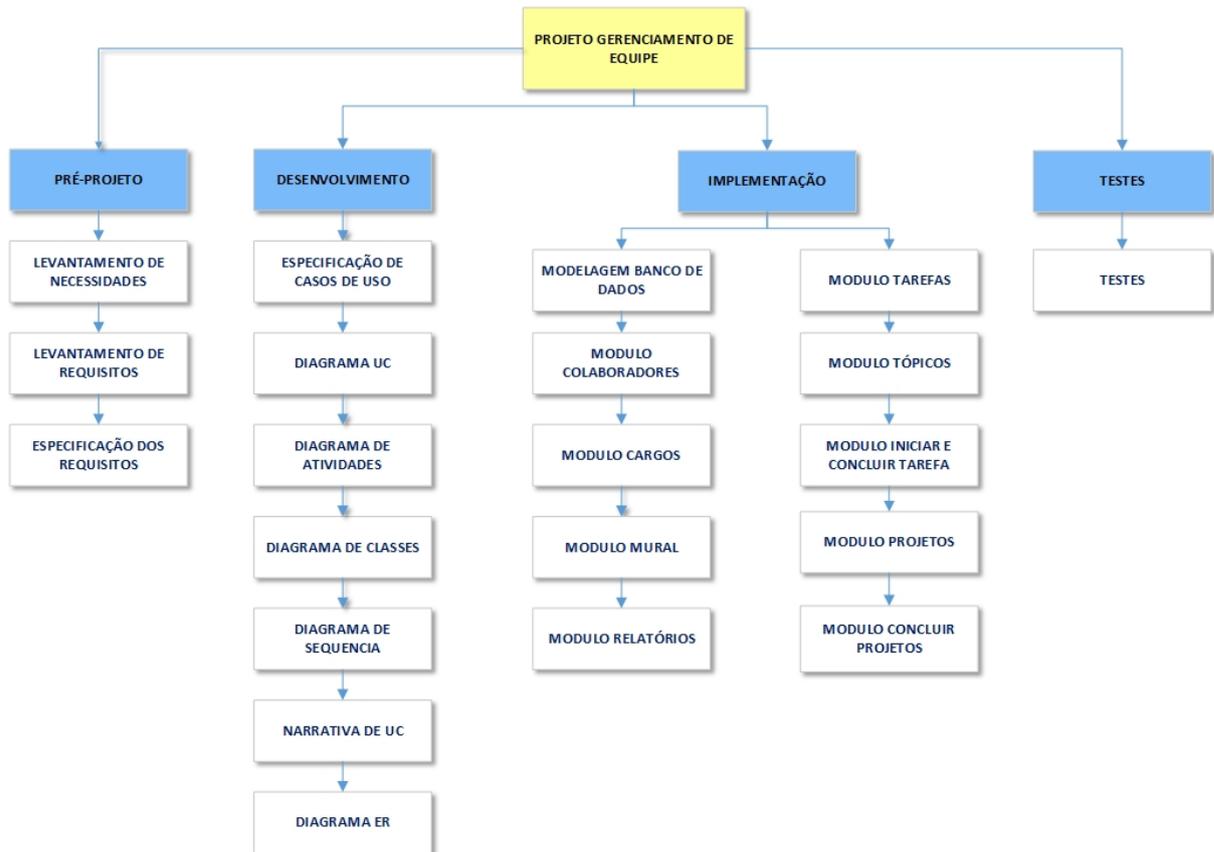


Figura 20. Diagrama WBS

7.3.21. SEQUÊNCIA DAS ATIVIDADES

O Diagrama de Sequência de Atividades representa a sequência em que as atividades foram realizadas durante o desenvolvimento deste trabalho e o período de duração de cada atividade.

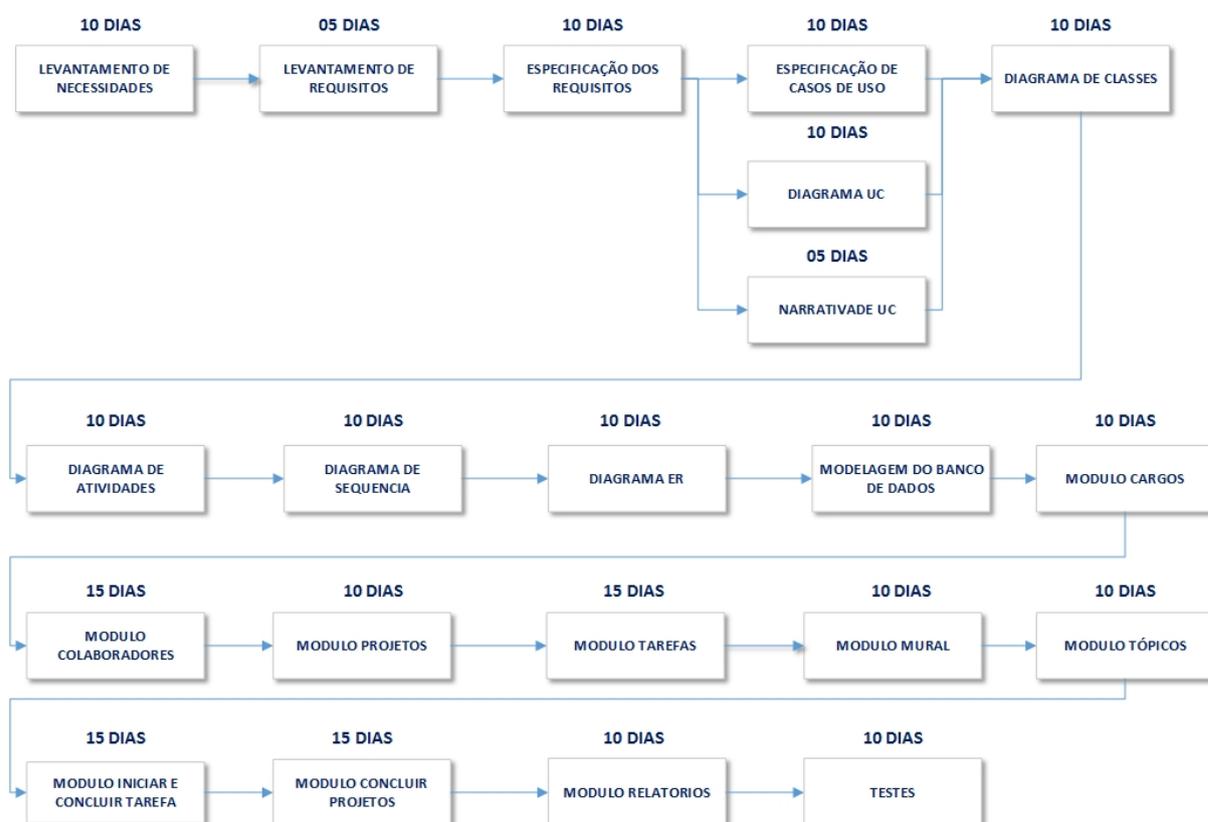


Figura 21. Diagrama Sequência de Atividades

7.4. ACESSO AO SISTEMA

O acesso ao sistema tem como finalidade de proteger as informações de usuários não cadastrados e filtrar as funcionalidades a que os usuários poderão utilizar no sistema, tendo como referência, seu nível de permissão cadastrador no banco de dados podendo ser administrador, líder de equipe ou colaborador. Para tal tarefa foi utilizado o sistema de Filtros (*Filter*) próprio da API de *Servlets* da plataforma Java EE, juntamente com a criptografia MD5.

A captura de tela mostra uma interface web de login. No topo, há uma barra de navegador com o endereço `localhost:45/ProjetoGE/pages/login.jsf`. Abaixo, o cabeçalho da página apresenta o logo "WEB SOLUTIONS.COM" e o subtítulo "Team Managemant". O conteúdo principal é um formulário centralizado com o título "Acesso ao Sistema". O formulário possui dois campos de entrada: "Login *" e "Senha *", ambos com ícones de olho para alternar a visibilidade. Abaixo dos campos, há um botão azul com o texto "Efetuar Login". No rodapé da página, há uma linha de texto que diz "Desenvolvido por Felipe de Lima Rodrigues - TCC 2014".

Figura 22. Acesso ao Sistema

7.4.1. FILTROS (FILTER)

Os filtros são classes que permitem implementar lógicas de validação antes de uma requisição ao servidor e também depois que essa resposta foi gerada, tornando possível restringir acesso indiscriminado as páginas da aplicação e também desacoplar essas lógicas. Para implementar essa interface *Filter* é necessário implementar três métodos: *init*, *destroy* e *doFilter*.

O método *init* é responsável uma lógica quando o filtro é carregado, o *destroy* quando ele é descarregado, e finalmente o *doFilter* é responsável pela lógica que irá decidir se há permissão ou não.

7.4.2. CRIPTOGRAFIA MD5

A Criptografia MD5 é um algoritmo de hash de 128 bits unidirecional desenvolvido pela RSA Data Security, Inc.. Por ser um algoritmo unidirecional, um hash MD5 não pode ser decriptografado. Para verificação é utilizada uma comparação das duas hash MD5. O MD5 é de domínio público para uso em geral e é utilizado como mecanismo de integridade em vários protocolos na Internet (MIT Laboratory for Computer Science and RSA Data Security, 1992).

A figura a seguir ilustra a utilização da classe Java MessageDigest para criptografar as senhas geradas ou alteradas pelo sistema no padrão de Hash MD5.

```
public class CriptografiaMD5 {

    public String convertStringToMd5(String valor) {
        MessageDigest mDigest;
        try {
            // Instanciando HASH MD5
            mDigest = MessageDigest.getInstance("MD5");

            // Convert a String valor para um array de bytes em MD5
            byte[] valorMD5 = mDigest.digest(valor.getBytes("UTF-8"));

            // Convertemos os bytes para hexadecimal
            StringBuffer sb = new StringBuffer();
            for (byte b : valorMD5) {
                sb.append(Integer.toHexString((b & 0xFF) |
0x100).substring(1, 3));
            }

            return sb.toString();

        } catch (NoSuchAlgorithmException e) {
            e.printStackTrace();
            return null;
        } catch (UnsupportedEncodingException e) {
            e.printStackTrace();
            return null;
        }
    }
}
```

Figura 23. Criptografia MD5

8. INTERFACES

Nesta seção serão apresentadas as principais interfaces desenvolvidas e uma breve descrição de sua utilização no sistema.

8.1. INTERFACE INICIAR TAREFA

Nesta interface são exibidas as informações da tarefa selecionada e o usuário possui as opções de iniciar, pausar ou fechar o diálogo de exibição. O usuário tem pode alterar nome, descrição e status dos tópicos cadastrados ou remove-lo.

Gerência de Equipes

localhost:8280/ProjetoGE/pages/secured/minhas-tarefas-em-andamento.jsf

Google

Mais visitados Primeiros passos JAVA Cursos JAVA, SAP, Vag... Sites Professores http://webmail.comu... Gabriel Zigolis - webst...

Opções

Tarefa

INFORMAÇÕES DA TAREFA

Projeto: Banco do Brasil
Código: 9
Nome: Services
Descrição: Camada de Serviços no Servidor
Data de Início: 10/08/2014

TÓPICOS				
Código	Nome	Descrição	Status	Opções
5	model	Desenvolver camadas de model	Em Andamento	Remover
6	View	Camada View	Em Andamento	Remover
7	Dao	Camada Dao	Em Andamento	Remover
8	Controller	Camada Controller	Parado	Remover

Duração: 00:00:38

▶ Iniciar || Pausar Fechar

Usuário Logado: Equipe de suporte

Figura 24. Interface iniciar tarefa

8.2. INTERFACE CONCLUIR TAREFA

Nesta interface é exibido o quadro de tarefas em andamento e o usuário possui as opções de concluir, visualizar informações ou abrir dialogo de iniciar a tarefa.

Gerência de Equipes

localhost:8280/ProjetoGE/pages/secured/minhas-tarefas-em-andamento.jsf

WEB SOLUTIONS.com
Team Managemant

Menu

Minhas Tarefas

- Pendentes
- Em Andamento
- Concluídas

Minhas Tarefas Em Andamento

Código	Nome	Descrição	Data de Início	Data de Término
9	Services	Camada de Serviços no Servidor	10/08/2014	31/08/2014

Iniciar Adicionar Tópico Concluir

Usuário Logado: Felipe de Lima - Cargo 5

Figura 25. Interface concluir tarefa

8.3. INTERFACE CONCLUIR PROJETO

Nesta interface é exibido o quadro de projetos em andamento e o usuário possui as opções de concluir e visualizar informações do projeto.

Gerência de Equipes

localhost:8280/ProjetoGE/pages/secured/meus-projetos-em-andamento.jsf

WEB SOLUTIONS.com
Team Management

Menu

Meus Projetos

- Pendentes
- Em Andamento
- Concluídos

Meus Projetos Em Andamento

Código	Nome	Descrição	Data de Início	Data de Término
9	Farmacia	Farmacia Catedral	10/08/2014	31/12/2014
10	Banco do Brasil	Computação Distribuida	10/08/2014	31/08/2014

Informações Concluir Projeto

Usuário Logado: Felipe de Lima - Cargo 5

Figura 26. Interface concluir projeto

8.4. INTERFACE CONSULTAR HORAS TRABALHADAS

Nesta interface é exibido o quadro de tarefas cadastradas para o usuário, constando o tempo trabalhado em cada tarefa e o tempo total. O usuário tem a opção de filtrar o período que deseja consultar ou consultar o período todo.

The screenshot displays a web browser window with the URL `localhost:8280/ProjetoGE/pages/secured/tempo-trabalhado.jsf`. The page title is "WEB SOLUTIONS.com Team Management". On the left, a sidebar lists "Utilidades" with options: "Mural Empresa", "Mural Equipe", and "Meu Tempo Trabalhado". The main content area is titled "TEMPO TRABALHADO" and features a search filter for the period: "Período De: 01/01/2014 Até: 01/01/2014" with a checked "Período inteiro:" option and a "Consultar" button. Below this is a table of tasks:

TAREFAS					
Nome	Status	Prioridade	Data de Início	Data de Término	Tempo Trabalho
Tarefa Joao	Em Andamento	Baixa	07/09/2014	12/09/2014	00:00:05
Nome	Status	Prioridade	Data de Início	Data de Término	Tempo Trabalho

Below the table, there are navigation controls (1, 7) and a "Total: 00:00:05" summary.

At the bottom of the page, the user is logged in as "Usuário Logado: João Carlos - Estagiário 1".

Figura 27. Interface consultar horas trabalhadas

8.5. INTERFACE DO GRÁFICO DE SITUAÇÃO DAS TAREFAS POR EQUIPE

Nesta interface é apresentado o gráfico sobre a situação do andamento das tarefas nos últimos 30 dias, em relação à equipe do usuário.

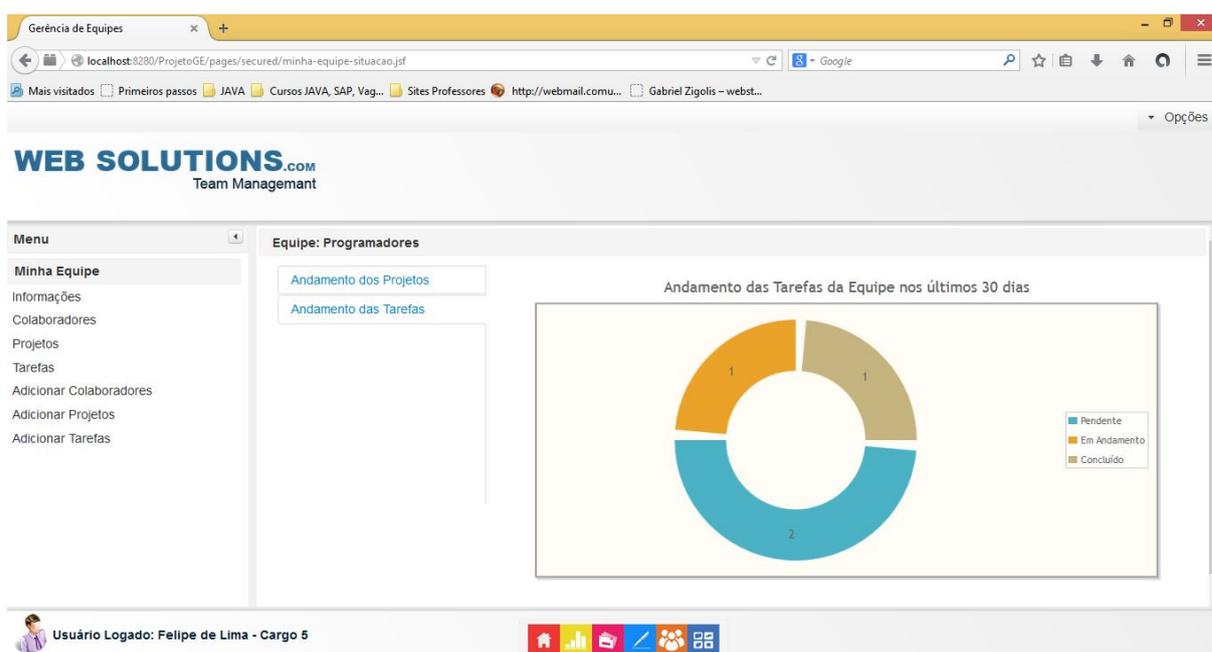
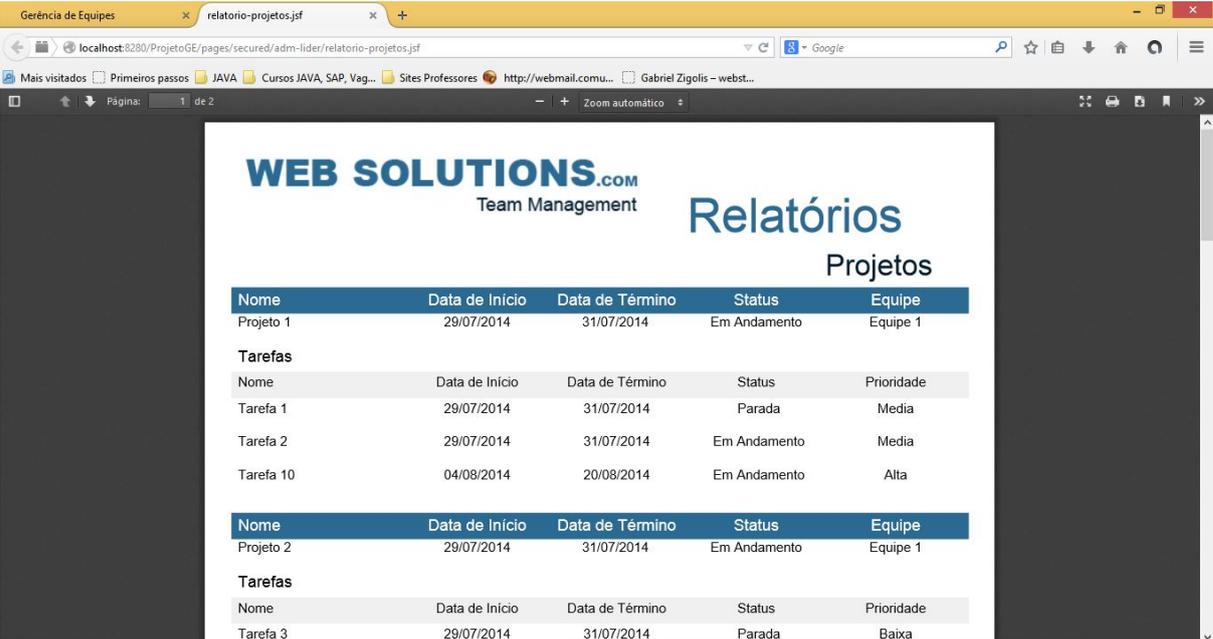


Figura 28. Interface do gráfico de situação das tarefas por equipe

8.6. INTERFACE RELATÓRIO GERADO PELO SISTEMA

Nesta interface é apresentado o relatório de projetos gerado pelo sistema e exibido pelo navegador no formato PDF. O usuário tem a opção de salvar o relatório gerado ou não.



Nome	Data de Início	Data de Término	Status	Equipe
Projeto 1	29/07/2014	31/07/2014	Em Andamento	Equipe 1

Tarefas

Nome	Data de Início	Data de Término	Status	Prioridade
Tarefa 1	29/07/2014	31/07/2014	Parada	Media
Tarefa 2	29/07/2014	31/07/2014	Em Andamento	Media
Tarefa 10	04/08/2014	20/08/2014	Em Andamento	Alta

Nome	Data de Início	Data de Término	Status	Equipe
Projeto 2	29/07/2014	31/07/2014	Em Andamento	Equipe 1

Tarefas

Nome	Data de Início	Data de Término	Status	Prioridade
Tarefa 3	29/07/2014	31/07/2014	Parada	Baixa

Figura 29. Interface relatório gerado pelo sistema

9. CONCLUSÃO

O software que foi elaborado é capaz de fornecer aos usuários funcionalidades para gerenciar suas equipes de trabalho, visando o controle e acompanhamento sobre as equipes, projetos, designação das tarefas aos colaboradores envolvidos, oferecendo também dados relativos às tarefas como status, prioridade, data de início e término e tempo gasto para realização de cada tarefa.

Outras funcionalidades implementadas no software é elaboração de relatórios sobre as equipe, projetos, tarefas e colaboradores da empresa de acordo com as datas que o usuário julgar relevante e também a disponibilização de gráficos que ilustram em tempo real o andamento das atividades que estão sendo realizadas.

A realização deste trabalho de conclusão de curso agregou conhecimentos no desenvolvimento de software, padrões de projeto, linguagem de programação Java, HTML, SQL, UML, utilização de frameworks para auxiliar no desenvolvimento de forma eficiente e produtiva, conhecimento sobre gerenciamento de equipes de trabalho, métodos de análise e acompanhamento de projetos entre outros benefícios de valor inestimável.

Como objetivo de trabalhos futuros, deseja-se realizar alterações que contribuam para a qualidade do produto, tornando-o um software robusto, que possa auxiliar o usuário da melhor e mais eficiente forma possível, podendo ser utilizado por empresas de médio e grande porte.

REFERÊNCIAS

EIS, Diego, FERREIRA, Elcio. **HTML5 e CSS3 com Farinha e Pimenta**. Tableless, 2012.

SANTOS, Rafael. **Introdução à Programação Orientada a Objetos usando Java**. Campus Editora, 2003.

DEITEL, Harvey M., DEITEL, Paul J.. **Java Como Programar, 8ª Edição**. Tradução Edson Furmankiewicz. São Paul: Pearson Prentice Hall, 2010.

Project Management Institute. **Um Guia do Conhecimento em Gerenciamento de Projetos (Guia PMBOK) – Quarta Edição**. Project Management Inst-id, 2009, Edição Digital.

GUEDES, Gilleanes T. A. **UML2 Uma Abordagem Prática**, São Paulo Editora Novatec, 2009.

BEIGHLEY, Lynn. **Use a Cabeça SQL**. Editora Alta Books, 2008.

MACHADO, Felipe Nery Rodrigues, ABREU, Maurício Pereira de. **Projeto Banco de Dados Uma Visão Prática**. Editora Érica Ltda, 2002.

Change Vision, Inc. **Astah Reference Manual Ver. 6.7**. Change Vision, Inc, 2006. Edição Digital.

Ministério do Planejamento, Orçamento e Gestão, Secretaria de Logística e Tecnologia da Informação. **Metodologia de Gerenciamento de Projetos do SISP**, Brasília: MP, 2011.

CARIBÉ, João Carlos. **Gestão de Projetos**. Engenharia Industrial Madeireirab – Universidade do Paraná. Disponível em: < <http://www.madeira.ufpr.br/disciplinasklock/gestaodeprojetos/Apostila%20GP%20leitura.pdf>>. Acesso em: 24 out. 2013.

CALEUM. **Apostila Java para Desenvolvimento WEB**. Disponível em: <http://www.caelum.com.br/apostila-java-web/recursos-importantes-filtros/>. Acesso em: 24 set. 2014.

Eclipse. **The Eclipse Foundation open source community website**. Disponível em: < <http://www.eclipse.org/org/> >. Acesso em: 12 jan. 2014.

CALEUM. **Uma introdução prática ao JPA com Hibernate**. Disponível em: <<http://www.caelum.com.br/apostila-java-web/uma-introducao-pratica-ao-jpa-com-hibernate/>>. Acesso em: 15 jan. 2014.

POLETTTO, Alex Sandro Romeo de Souza. **Banco de Dados I**. Assis-SP: FEMA, 2014. 124 p. Apostila para disciplina de graduação da Coordenadoria de Informática.

Adobe Systems Incorporated. **Programação do Adobe ActionScript 3.0 para Adobe Flash**. 2008. 750 p. Manual para Adobe ActionScript 3.0.

BERKUN, Scott. **A Arte do Gerenciamento de Projetos**. Bookman, 2008.

SOUZA, Thiago Hernandes de. **Relatórios Profissionais para Aplicação Java com as Ferramentas iReport e Pentalho Design**. Editora Ciência Moderna Ltda., 2013.

SOMMERVILLE, Ian. **Engenharia de Software**. 6ª Edição, São Paulo: Pearson Addison-Wesley, 2003.

RFC 1321. **The MD5 Message-Digest Algorithm**. Disponível em: <<http://www.faqs.org/rfcs/rfc1321.html>>. Acesso em: 10 set. 2014.

APÊNDICE A – Trigger insert_tarefa

```
CREATE TRIGGER insert_tarefa
AFTER INSERT ON tarefa
FOR EACH ROW
BEGIN
INSERT INTO banco_hora (hora, minuto, segundo, id_tarefa, id_projeto,
id_colaborador)
VALUES (0, 0, 0, NEW.idTarefa, NEW.id_projeto, NEW.id_colaborador);
END
```

APÊNDICE B – Trigger update_tarefa

```
BEFORE UPDATE ON tarefa
FOR EACH ROW
BEGIN
IF NEW.status = 'Em Andamento' THEN
UPDATE projeto SET status = 'Em Andamento' WHERE idProjeto = NEW.id_projeto;
END IF;
END
```

APÊNDICE C – Trigger delete_tarefa

```
CREATE TRIGGER delete_tarefa  
BEFORE DELETE ON tarefa  
FOR EACH ROW  
BEGIN  
DELETE FROM banco_hora WHERE banco_hora.id_tarefa = OLD.idTarefa;  
DELETE FROM topico WHERE topico.id_tarefa = OLD.idTarefa;  
END
```

APÊNDICE D – Trigger update_topico

```
CREATE TRIGGER update_topico  
BEFORE UPDATE ON topico  
FOR EACH ROW  
BEGIN  
IF NEW.status = 'Em Andamento' THEN  
UPDATE tarefa SET status = 'Em Andamento' WHERE idTarefa = NEW.id_tarefa;  
END IF;  
END
```

APÊNDICE E – Trigger update_banco_hora

```
CREATE TRIGGER update_banco_hora
BEFORE UPDATE ON banco_hora
FOR EACH ROW
BEGIN
IF NEW.hora > 0 OR NEW.minuto > 0 OR NEW.segundo > 0 THEN
UPDATE tarefa SET status = 'Em Andamento' WHERE idTarefa = NEW.id_tarefa;
END IF;
END;
```

APÊNDICE F – Trigger insert_equipe

```
CREATE TRIGGER insert_equipe
AFTER INSERT ON equipe
FOR EACH ROW
BEGIN
INSERT INTO mural (descricao, id_equipe)
VALUES (concat('Mural da ', NEW.nome), NEW.idEquipe);
END
```

APÊNDICE G – Trigger delete_equipe

```
CREATE TRIGGER delete_equipe  
BEFORE DELETE ON equipe  
FOR EACH ROW  
BEGIN  
DELETE FROM mural WHERE mural.id_equipe = OLD.idEquipe;  
END
```