



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

AMANDA NOGUEIRA

SISTEMA DE GESTÃO PARA LEGIÃO MIRIM DE BAURU-SP

Assis
2015

AMANDA NOGUEIRA

SISTEMA DE GESTÃO PARA LEGIÃO MIRIM DE BAURU-SP

Trabalho de Conclusão de Curso apresentado ao curso de Análise e Desenvolvimento de Sistemas 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.

Orientanda: Amanda Nogueira

Orientador: Luiz Ricardo Begosso

Orientador: Luiz Ricardo Begosso

Área de Concentração: Desenvolvimento de Sistemas

Assis
2015

FICHA CARTOGRÁFICA

NOGUEIRA, Amanda

Sistema de Gestão para Legião Mirim de Bauru-SP/ Amanda Nogueira. Fundação Educacional do Município de Assis – Assis, 2015.

58 páginas

Orientador: Luiz Ricardo Begosso

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

1. Java 2. Web

CDD: 001.61
Biblioteca da FEMA

SISTEMA DE GESTÃO PARA LEGIÃO MIRIM DE BAURU-SP

AMANDA NOGUEIRA

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

Orientador: Luiz Ricardo Begosso

Analisadora: Diomara Martins Reigato Barros

Assis
2015

DEDICATÓRIA

Dedico este trabalho a minha família e meu noivo, em especial meus pais, que me apoiaram durante toda minha faculdade sem medir esforços para que eu concluísse meu objetivo.

AGRADECIMENTOS

Primeiramente a Deus, por tudo de bom que tem realizado em minha vida e pela força e paciência que me proveu ao longo deste trabalho.

Aos meus familiares e meu noivo que me deram apoio e ajuda, principalmente meus pais, que mesmo em meio às dificuldades, nunca deixaram de dar toda ajuda e apoio de que precisei.

Ao professor, Luiz Ricardo Begosso, pela orientação e apoio dedicados durante, não somente este trabalho, mas, todo o curso.

Não há lugar para a sabedoria onde não há paciência.

Santo Agostinho

(354 – 430)

RESUMO

Este trabalho descreve o desenvolvimento de um sistema de gestão para Legião Mirim, que auxilia no gerenciamento dos processos organizacionais, administrando os legionários, cursos, matriculas e empresas parceiras.

O sistema será Web, desenvolvido com a IDE Eclipse na linguagem JAVA, utilizando o Hibernate para persistência de dados no Sistema Gerenciador de Banco de Dados MySQL, além do Spring para injeção de dependências e o PrimeFaces para a criação de interfaces.

Palavras-chave: Java, Hibernate, Spring, PrimeFaces.

ABSTRACT

The present work describes a managing system development for Legião Mirim, which assists the management of organizational systems, administering the legionaries, courses, registers and partner companies.

The system is Web, developed with IDE Eclipse in the JAVA language, using Hibernate for data persistence in the Data Base Management System MySQL, besides the Spring for dependency injection and the PrimeFaces to create interfaces.

Keywords: Java, Hibernate, Spring, PrimeFaces.

LISTA DE LUSTRAÇÕES

Figura 1 – Diagrama de Casos de Uso Geral.....	22
Figura 2 – Diagrama de Casos de Uso Manter legionário	23
Figura 3 – Diagrama de Casos de Uso Consultar legionário	24
Figura 4 – Diagrama de Casos de Uso Consultar currículo.....	25
Figura 5 – Diagrama de Casos de Uso Manter empresa.....	26
Figura 6 – Diagrama de Casos de Uso Manter Curso	27
Figura 7 – Diagrama de Casos de Uso Manter turma	28
Figura 8 – Diagrama de Casos de Uso Manter colaborador	29
Figura 9 – Diagrama de Casos de Uso Matricular legionário.....	30
Figura 10 – Diagrama de Casos de Uso Cancelar matrícula.....	31
Figura 11 – Diagrama de Casos de Uso Manter notas e faltas	32
Figura 12 – Diagrama de Casos de Uso Empregar legionário.....	33
Figura 13 – Diagrama de Casos de Uso Rescindir contrato	34
Figura 14 – Diagrama de Casos de Uso Gerar folha de pagamento	35
Figura 15 – Diagrama de Casos de Uso Manter pagamentos	36
Figura 16 – Diagrama de Casos de Uso Manter recebimentos	37
Figura 17 – Diagrama de Casos de Uso Manter férias	38
Figura 18 – Diagrama de Atividades Cancelar matrícula.....	39
Figura 19 – Diagrama de Atividades Rescindir contrato	40
Figura 20 – Diagrama de Atividades Inserir notas e faltas	41
Figura 21 – Diagrama de Atividades Editar notas e faltas	42
Figura 22 – Diagrama de Classes	43
Figura 23 – Diagrama Entidade-Relacionamento	44

LISTA DE TABELAS

Tabela 1 – Manter legionário.....	23
Tabela 2 – Consultar legionário	24
Tabela 3 – Consultar currículo	25
Tabela 4 – Manter empresa	26
Tabela 5 – Manter curso	27
Tabela 6 – Manter turma.....	28
Tabela 7 – Manter colaborador	29
Tabela 8 – Matricular legionário	30
Tabela 9 – Cancelar matrícula	31
Tabela 10 – Manter notas e faltas	32
Tabela 11 – Empregar legionário	33
Tabela 12 – Rescindir contrato	34
Tabela 13 – Gerar folha de pagamento.....	35
Tabela 14 – Manter pagamentos.....	36
Tabela 15 – Manter recebimentos.....	37
Tabela 16 – Manter férias	38

SUMÁRIO

1 INTRODUÇÃO	14
1.1 OBJETIVOS	14
1.2 PÚBLICO ALVO	15
1.3 JUSTIFICATIVA	15
2 TECNOLOGIAS	16
2.1 JAVA	16
2.1.1 Bibliotecas	17
2.1.2 Java Development Kit	17
2.1.3 Java Virtual Machine	17
2.2 PRIMEFACES	17
2.3 HIBERNATE	18
2.4 SPRING	18
2.5 MYSQL	19
2.6 APACHE TOMCAT	19
2.7 MODEL-VIEW-CONTROLLER	20
3 MODELAGEM	21
3.1 LISTA DE EVENTOS	21
3.2 DIAGRAMA DE CASOS DE USO GERAL	22
3.3 ESPECIFICAÇÃO DE CASOS DE USO	23
3.3.1 Manter legionário	23
3.3.2 Consultar legionário	24
3.3.3 Consultar currículo	25
3.3.4 Manter empresa	26
3.3.5 Manter curso	27
3.3.6 Manter turma	28
3.3.7 Manter colaborador	29
3.3.8 Matricular legionário	30
3.3.9 Cancelar matrícula	31
3.3.10 Manter notas e faltas	32
3.3.11 Empregar legionário	33
3.3.12 Rescindir contrato	34

3.3.13 Gerar folha de pagamento	35
3.3.14 Manter pagamentos	36
3.3.15 Manter recebimentos	37
3.3.16 Manter férias	38
3.4 DIAGRAMA DE ATIVIDADES	39
3.4.1 Cancelar Matrícula	39
3.4.2 Rescindir Contrato	40
3.4.3 Inserir notas e faltas	41
3.4.4 Editar notas e faltas	42
3.5 DIAGRAMA DE CLASSES	43
3.6 DIAGRAMA ENTIDADE-RELACIONAMENTO	44
4 CONCLUSÃO	45
REFERÊNCIAS	46

1. INTRODUÇÃO

Cada vez mais são indispensáveis sistemas no gerenciamento das empresas. Esses sistemas auxiliam no controle dos dados gerados no decorrer das atividades e melhoram o desempenho dos processos das organizações. Seguindo desse conceito, este trabalho irá desenvolver um sistema de gestão para Legião Mirim de Bauru.

A Legião Mirim de Bauru foi fundada oficialmente em Assembleia Geral na sede da Associação Comercial de Bauru, com aprovação de Estatutos em 22/06/1960, mas desde 04 de janeiro desse mesmo ano, funcionava em caráter experimental em uma sala cedida pelo antigo Núcleo Profissional na Estrada de Ferro Noroeste de Brasil.

No entanto, sua constituição, data do último trimestre de 1959, em reunião do Rotary Clube de Bauru, onde o presidente Délio Hermes de Oliveira Coragem, ficou autorizado a tomar medidas necessárias para a organização da Legião Mirim, passando esta a funcionar sob os cuidados do Rotary Clube de Bauru, agora com sua própria personalidade jurídica.

Com objetivo de oferecer aos adolescentes o conhecimento sobre o mundo do trabalho e educação desenvolvendo competências específicas e básicas, a Legião Mirim oferece diversos cursos com oportunidade de emprego ao adolescente. São mais de 80 empresas parceiras que pagam a ela o valor de uma bolsa salário, os encargos sociais, tais como INSS, FGTS e uma taxa de administração.

O sistema de gestão irá administrar os alunos, cursos, empresas parceiras e atividades desenvolvidas pela organização, com o intuito de facilitar e melhorar seu desempenho.

1.2 OBJETIVOS

Este trabalho tem por objetivo desenvolver um sistema que irá gerenciar o fluxo de informações da Legião Mirim de Bauru - SP, auxiliando na administração dos alunos, cursos oferecidos e no controle de empresas parceiras, com as seguintes funções:

- Controlar dados cadastrais dos colaboradores da Legião Mirim;
- Controlar dados cadastrais dos jovens;
- Controlar exames médicos, bem como demais benefícios recebidos;
- Permitir o cadastro das Empresas parceiras, que empregam os jovens;
- Permitir o cadastramento com carga horária e critérios de desempenho dos cursos os quais são oferecidos aos jovens;
- Controlar o desempenho dos jovens nos cursos;
- Controlar a vida profissional dos jovens, controlando períodos trabalhados por esse jovem e empresas que o empregaram;
- Controle dos valores mensais devidos pelas empresas parceiras, correspondentes ao número de jovens que prestaram serviços naquela organização;
- Controle da Folha de Pagamento dos estudantes.

1.3 PÚBLICO ALVO

O trabalho terá como público alvo a Legião Mirim de Bauru - SP.

1.4 JUSTIFICATIVA

Gerenciar e controlar uma organização pequena no papel pode até funcionar, porém a medida que ela vai crescendo e as informações aumentando fica mais difícil e inviável. Desta forma dá-se a necessidade de um sistema para o desenvolvimento de sua função de forma mais eficaz.

2. TECNOLOGIAS

O sistema apresentado neste trabalho será desenvolvido para a Web utilizando a linguagem orientada a objetos Java. A IDE Eclipse será o ambiente de desenvolvimento juntamente com a ferramenta de mapeamento objeto/relacional Hibernate que auxiliará na persistência dos dados no MySQL, banco de dados que será utilizado neste projeto.

O MVC será o padrão de projeto de software que permitirá a separação do projeto em camadas de modelo, visão e controle. A biblioteca Primefaces facilitará a criação de interfaces enquanto o Apache Tomcat será o servidor de aplicação utilizado.

2.1 JAVA

Java é uma linguagem de programação que desde seu início tem crescido. Segundo DEITEL (2010), tornou-se a preferida para implementar aplicativos baseados na Internet e software para dispositivos que se comunicam por uma rede.

A princípio o projeto da Sun Microsystems era desenvolver uma linguagem para dispositivos eletrônicos, a linguagem foi criada por James Gosling e foi chamada de Oak, porém descobriu-se que já havia uma linguagem com esse nome e então quando uma equipe da Sun visitou uma cafeteria, o nome da cidade de origem de um café importado, Java, foi sugerido e ficou.

Mas havia um problema, o mercado de dispositivos eletrônicos não estava se desenvolvendo como previa a Sun e o projeto passou por dificuldades. Por coincidência foi quando a Web começou a se popularizar e a Sun viu uma oportunidade de utilizar o Java para adicionar conteúdo dinâmico e animações às páginas Web.

Por fim em maio de 1995 a Sun anunciou o Java formalmente e atraiu a atenção da comunidade de negócios devido ao grande interesse pela Web e hoje é utilizada para o desenvolvimento de aplicativos para dispositivos móveis, aplicações corporativas de grande porte, entre outros.

2.1.1 Bibliotecas

A linguagem Java é orientada a objetos e, portanto, os programas consistem em partes chamadas classes, que por sua vez, incluem partes chamadas métodos, que realizam tarefas e retornam informações. E para otimizar a programação existem as Java APIs (Application Programming Interfaces), que são bibliotecas de classe Java com classes já existentes para serem utilizadas no desenvolvimento.

2.1.2 Java Development Kit

O JDK (Java Development Kit) é um kit de desenvolvimento que possui as ferramentas necessárias para o desenvolvimento como o Java Runtime Environment, o compilador Java e as APIs Java.

2.1.3 Java Virtual Machine

Após o programa ter sido digitado no editor e corrigido pelo mesmo, o compilador Java converte o código-fonte em bytecodes, que representam tarefas a serem executadas, pela Java Virtual Machine (JVM). Segundo DEITEL (2010), se a mesma Virtual Machine for implantada nas várias plataformas do computador, os aplicativos que ela executa podem ser utilizados em todas essas plataformas.

2.2 PRIMEFACES

O primefaces é um conjunto de componentes JSF, com o objetivo de facilitar a criação de interfaces mais ricas, com componentes mais sofisticados. Com o primefaces é possível adicionar a interface DataTable, AutoComplete, HTML editor, gráficos, temas, etc. de maneira bem mais simplificada sem precisar, o

desenvolvedor, ficar se preocupando com chamadas assíncronas ao servidor já que os componentes trabalham com AJAX por "default".

2.3 HIBERNATE

Hibernate é um framework de mapeamento objeto/relacional para persistência de dados em um banco de forma simplificada possibilitando assim que o desenvolvedor tenha mais liberdade para se concentrar na lógica do negócio. A linguagem utilizada para consulta do Hibernate chama-se HQL (Hibernate Query Language) muito parecida com a linguagem SQL, possuindo apenas algumas diferenças como, enquanto a SQL usa tabelas e colunas a HQL utiliza nome de classes e nome de propriedades.

Em poucas palavras, mapeamento objeto/relacional é a persistência automatizada (e transparente) dos objetos em uma aplicação Java para as tabelas de um banco de dados relacional, utilizando metadados que descrevem o mapeamento entre objetos e o banco de dados (BAUER; KING, 2007).

2.4 SPRING

Spring é um framework que fornece uma série de soluções para o desenvolvimento, tais como injeção de dependências, várias opções para persistência em banco de dados, recursos para envio de e-mails, mapeamento automático a partir das hierarquias de exceções proprietárias, entre outras diversas funcionalidades.

Um benefício que o Spring oferece é ser organizado de maneira modular, permitindo que você utilize apenas os módulos que seu sistema precisará.

Os módulos que utilizarei no desenvolvimento deste projeto serão:

Spring Core – é o módulo base para o Spring que fornece, além de outros recursos, injeção de dependências e gerenciamento dos beans.

Spring Context – é outro módulo também muito importante que contém funcionalidades fundamentais como ApplicationContext, WebApplicationContext, recursos para envio de e-mails, para validação de campos da interface Web dentre outros pacotes que este módulo oferece.

Spring ORM – é o módulo que oferece suporte de integração aos produtos ORM (Object-Relational Map). No desenvolvimento desse trabalho a integração será com o Hibernate.

2.5 MYSQL

Para o projeto o sistema gerenciador banco de dados será o MySQL que é um banco de dados relacional. O banco de dados relacional, possui tabelas e colunas que se relacionam entre si e dentre outras vantagens garantem e protegem a integridade dos dados. Os dados não se perdem ao encerrarmos a aplicação e por isso é possível que os dados sejam compartilhados por mais de uma aplicação e tecnologias diferentes.

O MySQL foi desenvolvido por uma empresa da Suécia com o intuito de obterem um sistema de banco de dados rápido e flexível, para tal, se basearam em um outro sistema gerenciador de banco de dados chamado mSQL. Hoje o MySQL é utilizado por muitos lugares no mundo e por grandes organizações tais como Facebook, Google, Adobe, Alcatel Lucent e Zappos.

2.6 APACHE TOMCAT

O servidor de aplicação Apache Tomcat é um contêiner servlet criado a partir da junção dos projetos Java Web Server, que teve seu código doado pela Sun Microsystems em 1999 ao Apache Software Foundation (ASF), e JServ, um servlet engine que se integrava com o servidor Web Apache.

O Tomcat é uma implementação das tecnologias Java Servlet e JavaServer Pages que, além de ser um Container Web, o Tomcat pode também funcionar como servidor Web/HTTP, assim como também, pode funcionar integrado a um servidor web dedicado.

2.7 MODEL-VIEW-CONTROLLER

O MVC é um paradigma de desenvolvimento e design muito aceito no desenvolvimento de aplicações Java, principalmente Web, que separa em três partes uma aplicação: Model, View e Controller.

A camada Model é composta pelas classes que ficam responsáveis por manejar os dados, armazenando e buscando. Já a camada View é responsável pela apresentação visual dos dados para o usuário. Enquanto a camada Controller gerencia como as ordens do usuário deverão ser executadas sobre os dados e as apresentações que deverão ser exibidas.

3. MODELAGEM

Neste capítulo serão apresentadas as informações, modelagem e análise do sistema.

3.1 LISTA DE EVENTOS

Para o sistema foi desenvolvida a seguinte lista de eventos:

1. Manter legionário
2. Consultar currículo
3. Manter empresa
4. Manter curso
5. Manter turma
6. Manter colaborador
7. Matricular legionário
8. Cancelar matrícula
9. Manter notas e faltas
10. Empregar legionário
11. Rescindir contrato
12. Gerar folha de pagamento
13. Manter pagamentos
14. Manter recebimentos
15. Manter férias

3.2 DIAGRAMA DE CASOS DE USO GERAL

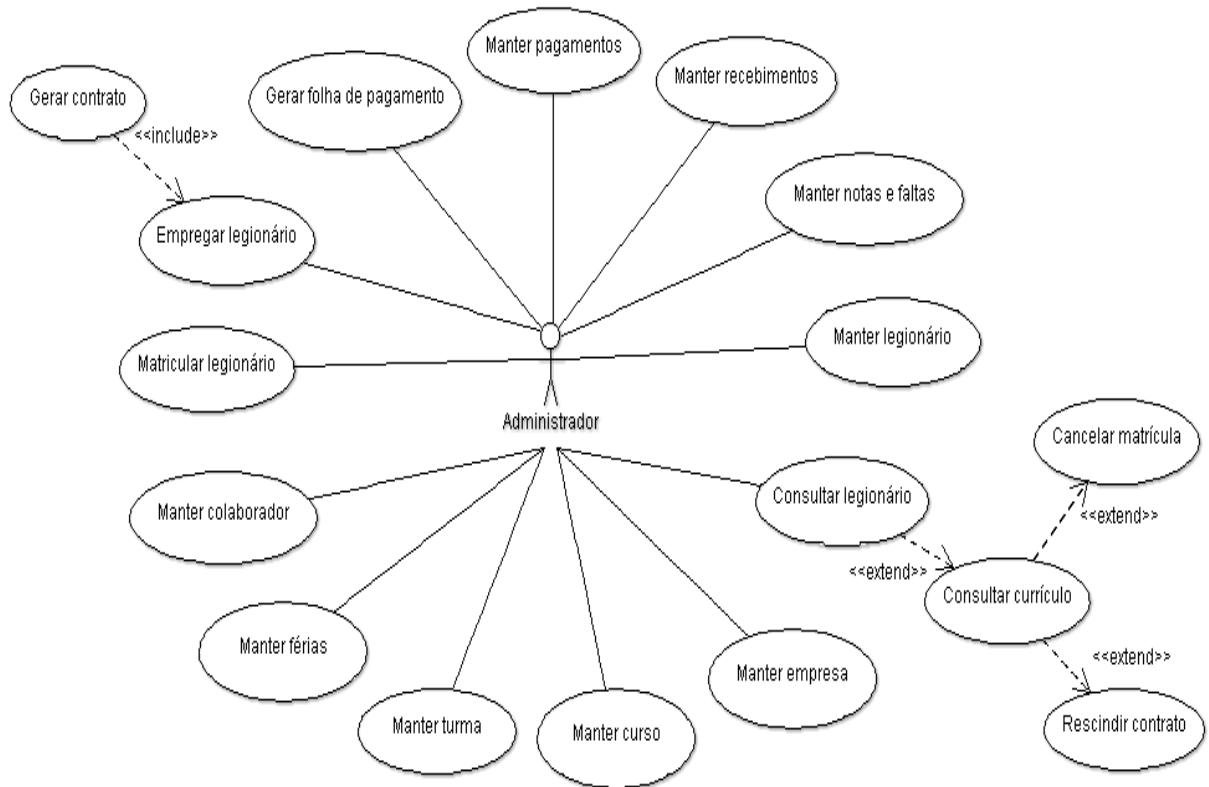


Figura 1 – Diagrama de Casos de Uso Geral

3.3 ESPECIFICAÇÃO DE CASOS DE USO

3.3.1 Manter legionário

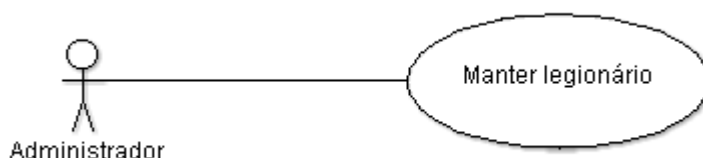


Figura 2 – Diagrama de Casos de Uso Manter legionário

Nome do Caso de Uso:	Manter Legionário
Atores:	Administrador
Pré-condições:	O administrador deve estar logado no sistema
Fluxo Principal:	1 - O usuário acessa a opção cadastrar Legionário 2 - O sistema dispõe campos para o preenchimento 3 - O usuário preenche os campos e confirma o cadastro 4 - O sistema salva os dados do legionário
Fluxos alternativos:	O usuário pode cancelar operação.
Fluxos de Exceção:	O sistema verifica que o legionário já possui cadastro e exibe a mensagem de legionário já cadastrado

Tabela 1 – Manter legionário

3.3.2 Consultar legionário

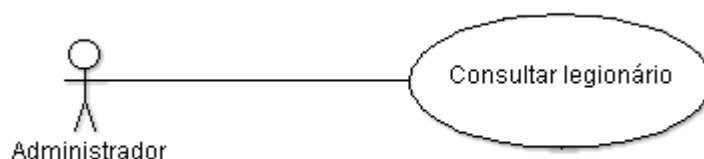


Figura 3 – Diagrama de Casos de Uso Consultar legionário

Nome do Caso de Uso:	Consultar legionário
Atores:	Administrador
Pré-condições	O administrador deve estar logado no sistema
Fluxo Principal:	1- O usuário acessa a opção consultar legionário 2 - O sistema dispõe os campos nome e CPF para serem preenchidos 3 - O usuário preenche um dos campos e clica em buscar 4 - O sistema busca o legionário e retorna os dados e as opções editar e currículo
Fluxos Alternativos:	O usuário pode alterar os dados ou consultar o currículo do legionário
Fluxos de Exceção:	O sistema não encontra o legionário e retorna a mensagem de legionário não encontrado

Tabela 2 – Consultar legionário

3.3.3 Consultar currículo

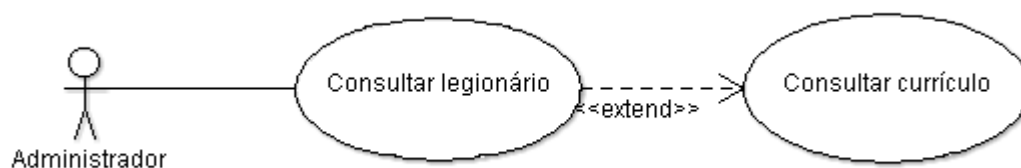


Figura 4 – Diagrama de Casos de Uso Consultar currículo

Nome do Caso de Uso:	Consultar currículo
Atores:	Administrador
Pré-condições:	O administrador deve estar logado no sistema
Fluxo Principal:	1 – O usuário acessa a opção consultar legionário 2 – O sistema dispõe os campos nome e CPF para serem preenchidos 3 – O usuário preenche um dos campos e clica em buscar 4 – O sistema busca o legionário e retorna os dados e as opções 'editar' e 'currículo' 5 – O usuário clica em currículo 6 – O sistema exibe os cursos e empregos do legionário
Fluxos Alternativos:	
Fluxos de Exceção:	O sistema não encontra o legionário e retorna a mensagem de legionário não encontrado

Tabela 3 – Consultar currículo

3.3.4 Manter empresa

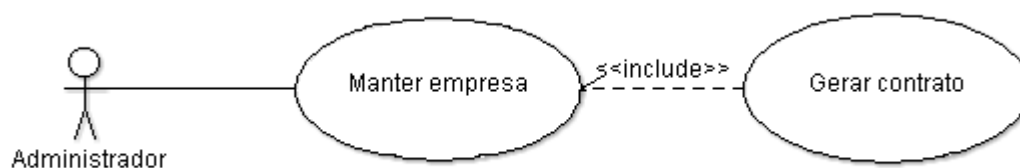


Figura 5 – Diagrama de Casos de Uso Manter empresa

Nome do Caso de Uso:	Manter empresa
Atores:	Administrador
Pré-condições:	O administrador deve estar logado no sistema
Fluxo Principal:	<p>1 – O usuário acessa a opção cadastrar empresa</p> <p>2 – O sistema dispõe os campos para preenchimento</p> <p>3 – O usuário preenche os campos e confirma o cadastro</p> <p>4 – O sistema salva os dados da empresa e gera o contrato para impressão</p>
Fluxos Alternativos:	O usuário pode cancelar a operação, alterar ou excluir os dados
Fluxos de Exceção:	O sistema verifica que a empresa já possui cadastro e retorna a mensagem de empresa já cadastrada

Tabela 4 – Manter empresa

3.3.5 Manter curso

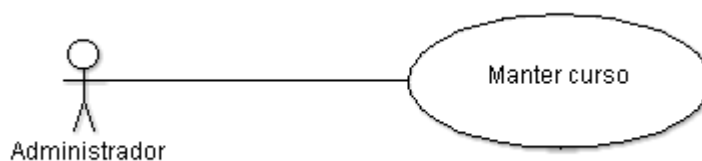


Figura 6 – Diagrama de Casos de Uso Manter curso

Nome do Caso de Uso:	Manter curso
Atores:	Administrador
Pré-condições:	O administrador deve estar logado no sistema
Fluxo Principal:	1 – O usuário acessa a opção cadastrar curso 2 – O sistema dispõe os campos para preenchimento 3 – O usuário preenche os campos e confirma o cadastro 4 – O sistema salva os dados do curso
Fluxos Alternativos:	O usuário pode cancelar operação, alterar ou excluir dados.
Fluxos de Exceção:	

Tabela 5 – Manter curso

3.3.6 Manter turma

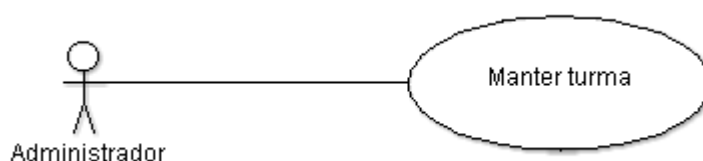


Figura 7 – Diagrama de Casos de Uso Manter turma

Nome do Caso de Uso:	Manter turma
Atores:	Administrador
Pré-condições:	1 – O administrador deve estar logado no sistema 2 – O curso o qual deseja cadastrar turma deve estar cadastrado
Fluxo Principal:	1 – O usuário acessa o sistema e a opção cadastrar turma 2 – O sistema dispõe os campos para o usuário selecionar o curso e preencher demais informações 3- O usuário seleciona o curso, preenche os campos e confirma o cadastro 4 – O sistema salva os dados do curso
Fluxos Alternativos:	O usuário cancelar operação, alterar ou excluir dados
Fluxos de Exceção:	

Tabela 6 – Manter turma

3.3.7 Manter colaborador

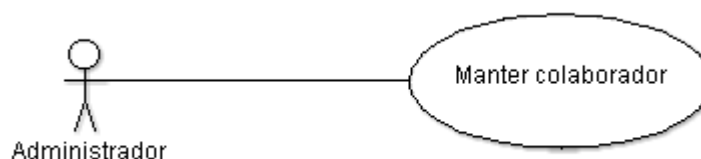


Figura 8 – Diagrama de Casos de Uso Manter colaborador

Nome do Caso de Uso:	Manter colaborador
Atores:	Administrador
Pré-condições:	O administrador deve estar logado no sistema
Fluxo Principal:	1 - O usuário acessa a opção cadastrar colaborador 2 - O sistema dispõe os campos para preenchimento 3 - O usuário preenche os campos e confirma o cadastro 4 - O sistema salva os dados do colaborador
Fluxos Alternativos:	O usuário pode cancelar operação, alterar ou excluir dados
Fluxos de Exceção:	O sistema verifica que o colaborador já possui cadastro e retorna a mensagem de colaborador já cadastrado

Tabela 7 – Manter colaborador

3.3.8 Matricular legionário

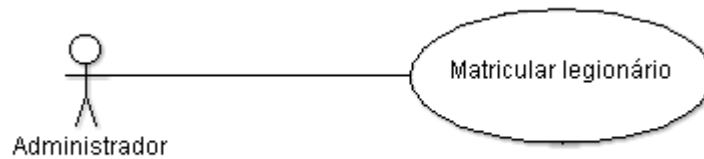


Figura 9 – Diagrama de Casos de Uso Matricular legionário

Nome do Caso de Uso:	Matricular legionário
Atores:	Administrador
Pré-condições:	1 – O administrador deve estar logado no sistema
Fluxo Principal:	1 – O usuário acessa a opção matricular legionário 2 – O sistema pede o CPF do legionário 3 – O usuário preenche o campo e confirma 4 – O sistema busca o aluno e retorna as opções para seleção 5 – O usuário seleciona o curso e confirma matrícula 6 – O sistema salva as informações
Fluxos Alternativos:	O usuário não confirma matrícula
Fluxos de Exceção:	O sistema não encontra o legionário

Tabela 8 – Matricular legionário

3.3.9 Cancelar matrícula

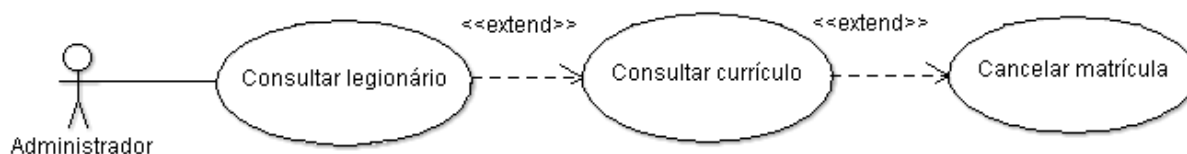


Figura 10 – Diagrama de Casos de Uso Cancelar matrícula

Nome do Caso de Uso:	Cancelar matrícula
Atores:	Administrador
Pré-condições:	1 - O administrador deve estar logado no sistema
Fluxo Principal:	1 - O usuário acessa a opção consultar legionário 2 - O sistema dispõe os campos nome e CPF para serem preenchidos 3 - O usuário preenche um dos campos e clica em buscar 4 - O sistema busca o legionário e retorna os dados e as opções editar e currículo 5 - O usuário clica na opção currículo 6 - O sistema exibe os cursos do legionário e a opção cancelar 7 - O usuário clica na opção cancelar 8 - O sistema exibe as informações do cancelamento e um campo para preencher com o motivo do cancelamento 9 - O usuário preenche o campo do motivo do cancelamento e confirma o cancelamento 10 - O sistema salva as alterações
Fluxos Alternativos:	O usuário pode cancelar a operação
Fluxos de Exceção:	4.1 - O sistema não encontra o legionário 6.1 - O sistema não encontra nenhum curso vinculado ao aluno

Tabela 9 – Cancelar matrícula

3.3.10 Manter notas e faltas

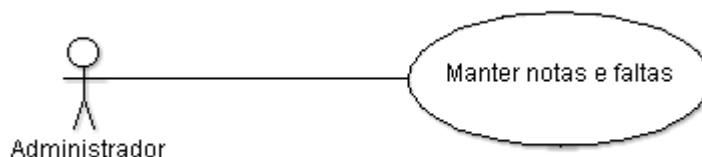


Figura 11 – Diagrama de Casos de Uso Manter notas e faltas

Nome do Caso de Uso:	Manter notas e faltas
Atores:	Administrador
Pré-condições:	O administrador deve estar logado no sistema
Fluxo Principal:	<p>1 - O usuário acessa a opção notas e faltas</p> <p>2 - O sistema dispõe os campos para seleção do curso</p> <p>3 - O usuário seleciona o curso</p> <p>4 - O sistema exibe os legionários matriculados no curso selecionado e a opção inserir notas e faltas</p> <p>5 - O usuário clica na opção inserir notas e faltas</p> <p>6 - O sistema exibe a tela para preenchimento de notas e faltas dos legionários</p> <p>7 - O usuário insere os dados e clica em salvar</p> <p>8 - O sistema salva os dados</p>
Fluxos Alternativos:	O usuário pode cancelar a operação e alterar os dados
Fluxos de Exceção:	O sistema não encontra nenhum legionário matriculado no curso

Tabela 10 – Manter notas e faltas

3.3.11 Empregar legionário

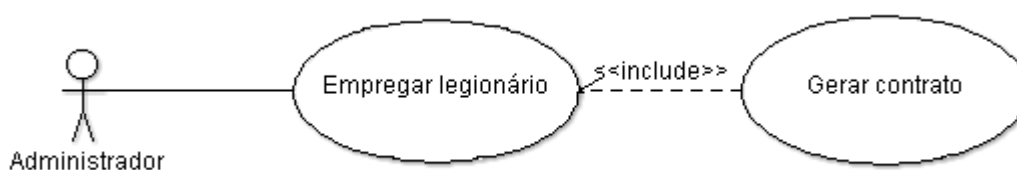


Figura 12 – Diagrama de Casos de Uso Empregar legionário

Nome do Caso de Uso:	Empregar legionário
Atores:	Administrador
Pré-condições:	O administrador deve estar logado no sistema
Fluxo Principal:	1 - O usuário acessa a opção empregar legionário 2 - O sistema pede o CPF do legionário 3 - O usuário preenche o campo e confirma 4 - O sistema busca o aluno e retorna os campos para serem preenchidos 5 - O usuário seleciona a empresa, preenche todos os campos e confirma 6 - O sistema salva as informações e gera o contrato para ser impresso
Fluxos Alternativos:	O usuário pode cancelar a operação
Fluxos de Exceção:	O sistema não encontra o legionário

Tabela 11 – Empregar legionário

3.3.12 Rescindir contrato

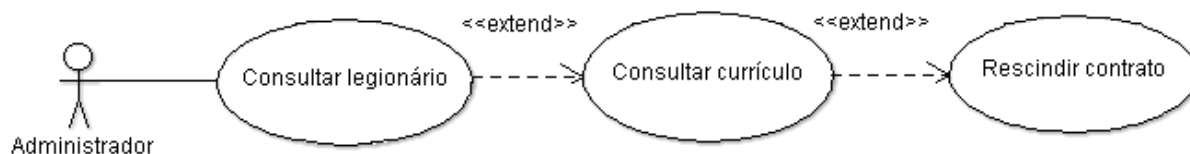


Figura 13 – Diagrama de Casos de Uso Rescindir contrato

Nome do Caso de Uso:	Rescindir contrato
Atores:	Administrador
Pré-condições:	O administrador deve estar logado no sistema
Fluxo Principal:	<p>1 - O usuário acessa a opção consultar legionário</p> <p>2 - O sistema dispõe os campos nome e CPF para serem preenchidos</p> <p>3 - O usuário preenche um dos campos e clica em buscar</p> <p>4 - O sistema busca o legionário e retorna os dados e as opções editar e currículo</p> <p>5 - O usuário clica em currículo</p> <p>6 - O sistema exibe os cursos e empregos do legionário</p> <p>7 - O usuário clica em rescindir contrato</p> <p>8 - O sistema exibe a tela de rescisão com os dados para preenchimento</p> <p>9 - O usuário preenche os dados e clica em rescindir contrato</p> <p>10 - O sistema grava os dados</p>
Fluxos Alternativos:	O usuário pode cancelar a operação
Fluxos de Exceção:	<p>4.1 - O sistema não encontra o legionário</p> <p>6.1 - O sistema não encontra o emprego vinculado ao legionário</p>

Tabela 12 – Rescindir contrato

3.3.13 Gerar folha de pagamento

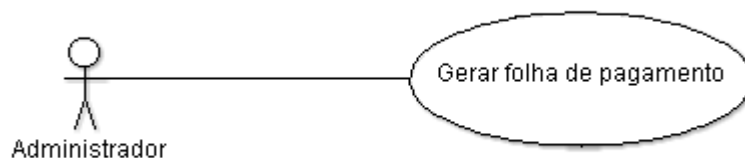


Figura 14 – Diagrama de Casos de Uso Gerar folha de pagamento

Nome do Caso de Uso:	Gerar folha de pagamento
Atores:	Administrador
Pré-condições:	O administrador deve estar logado no sistema
Fluxo Principal:	1 - O usuário acessa a opção gerar folha de pagamento 2 - O sistema dispõe os campos para seleção da empresa e data 3 - O usuário seleciona a empresa e o mês 4 - O sistema exibe os legionários ativos vinculados com a empresa 5 - O usuário seleciona os legionários que deseja gerar a folha de pagamento e clica em gerar 6 - O sistema gera as folhas de pagamento dos legionários selecionados
Fluxos Alternativos:	
Fluxos de Exceção:	O sistema não encontra nenhum legionário ativo vinculado à empresa

Tabela 13 – Gerar folha de pagamento

3.3.14 Manter pagamentos

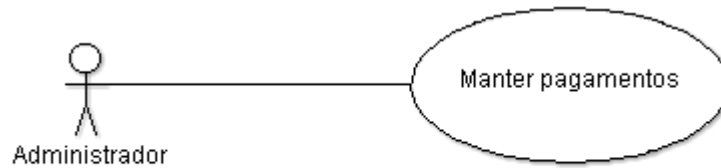


Figura 15 – Diagrama de Casos de Uso Manter pagamentos

Nome do Caso de Uso:	Manter pagamentos
Atores:	Administrador
Pré-condições:	O administrador deve estar logado no sistema
Fluxo Principal:	1 - O usuário acessa a opção gerar pagamentos 2 - O sistema dispõe os campos para seleção da empresa e data 3 - O usuário seleciona a empresa e o mês 4 - O sistema exibe os legionários ativos vinculados com a empresa e os valores que a empresa deverá pagar a cada legionário 5 - O usuário seleciona se a empresa realizou o pagamento sim ou não e clica em salvar 6 - O sistema salva as informações
Fluxos Alternativos:	O usuário pode consultar os pagamentos
Fluxos de Exceção:	O sistema não encontra nenhum legionário ativo vinculado à empresa

Tabela 14 – Manter pagamentos

3.3.15 Manter recebimentos

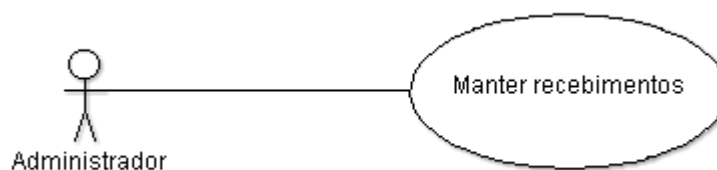


Figura 16 – Diagrama de Casos de Uso Manter recebimentos

Nome do Caso de Uso:	Gerar recebimentos
Atores:	Administrador
Pré-condições:	O administrador deve estar logado no sistema
Fluxo Principal:	<p>1 - O usuário acessa a opção gerar recebimentos</p> <p>2 - O sistema dispõe os campos para seleção da empresa</p> <p>3 - O usuário seleciona a empresa</p> <p>4 - O sistema exibe os legionários ativos vinculados com a empresa e os valores que legionário deve receber</p> <p>5 - O usuário seleciona os usuários que irão ser pagos e clica na opção pagar</p> <p>6 - O sistema salva as informações</p>
Fluxos Alternativos:	O usuário pode consultar os recebimentos
Fluxos de Exceção:	O sistema não encontra nenhum legionário ativo vinculado à empresa

Tabela 15 – Manter recebimentos

3.3.16 Manter férias

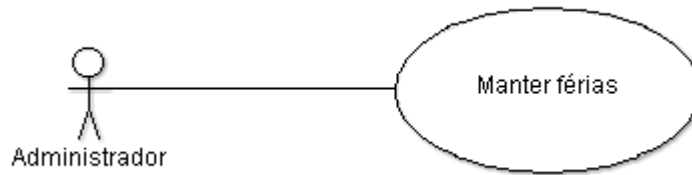


Figura 17 – Diagrama de Casos de Uso Manter férias

Nome do Caso de Uso:	Manter férias
Atores:	Administrador
Pré-condições:	O administrador deve estar logado no sistema
Fluxo Principal:	1 - O usuário acessa a opção consultar férias 2 - O sistema dispõe os campos para seleção da empresa 3 - O usuário seleciona a empresa 4 - O sistema exibe os legionários ativos vinculados com a empresa, as datas das últimas férias e data prevista para as próximas férias 5 - O usuário insere a data que o legionário entrou de férias 6 - O sistema salva as informações
Fluxos Alternativos:	O usuário pode consultar as férias
Fluxos de Exceção:	O sistema não encontra nenhum legionário ativo vinculado à empresa

Tabela 16 – Manter férias

3.4 DIAGRAMA DE ATIVIDADES

3.4.1 Cancelar matrícula

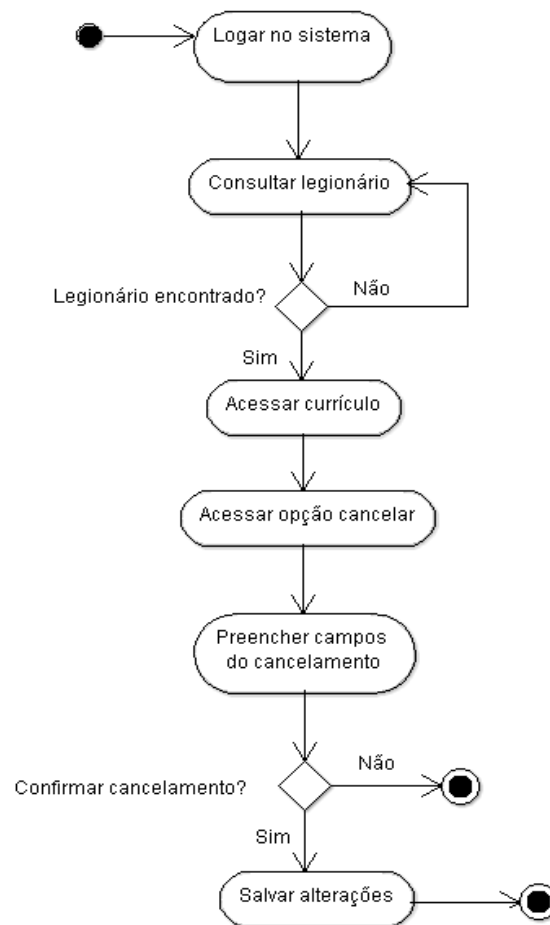


Figura 18 – Diagrama de Atividades Cancelar matrícula

3.4.2 Rescindir contrato

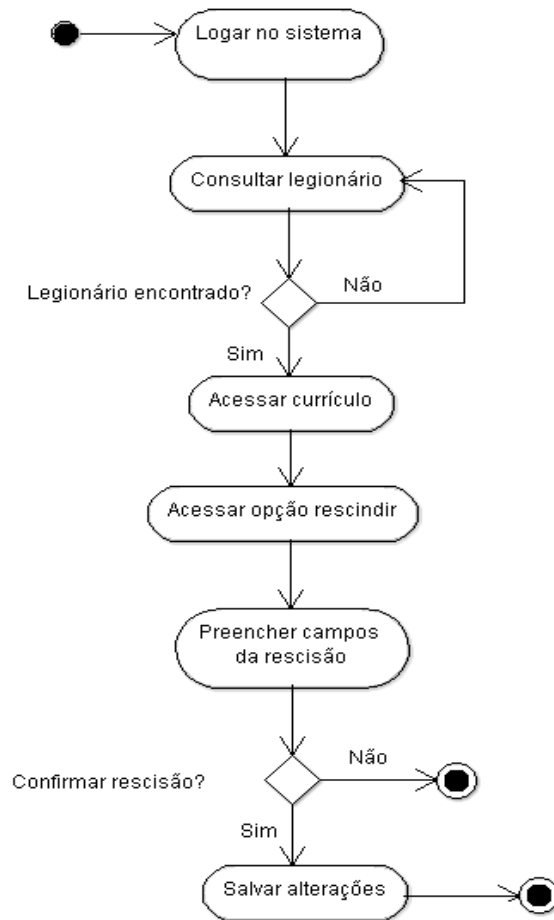


Figura 19 – Diagrama de Atividades Rescindir contrato

3.4.3 Inserir notas e faltas

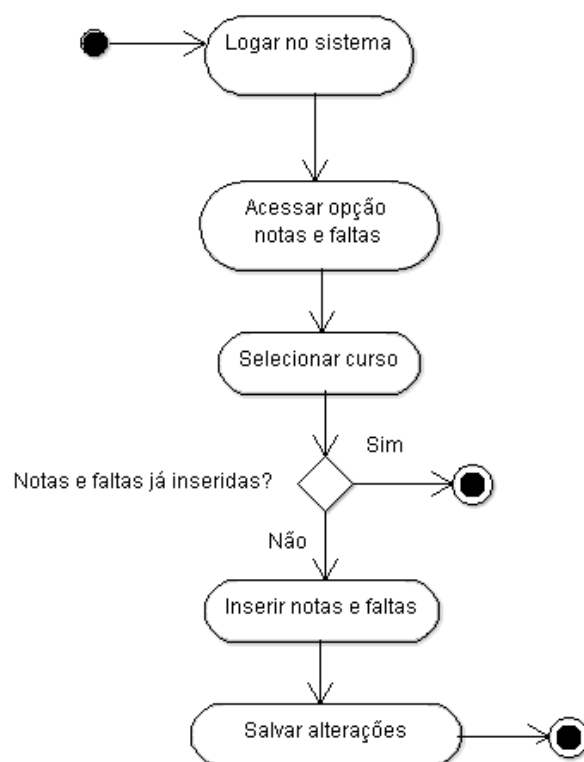


Figura 20 – Diagrama de Atividades Inserir notas e faltas

3.4.4 Editar notas e faltas

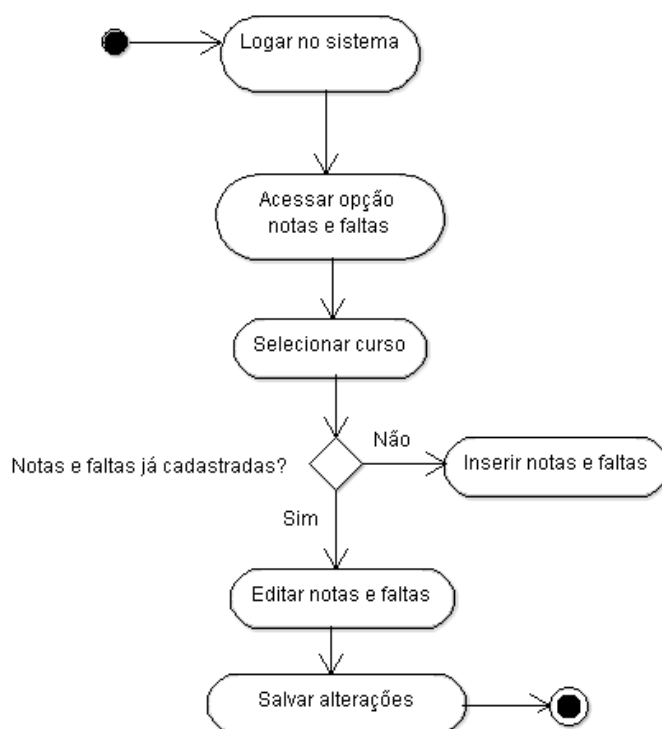


Figura 21 – Diagrama de Atividades Editar notas e faltas

3.5 DIAGRAMA DE CLASSES

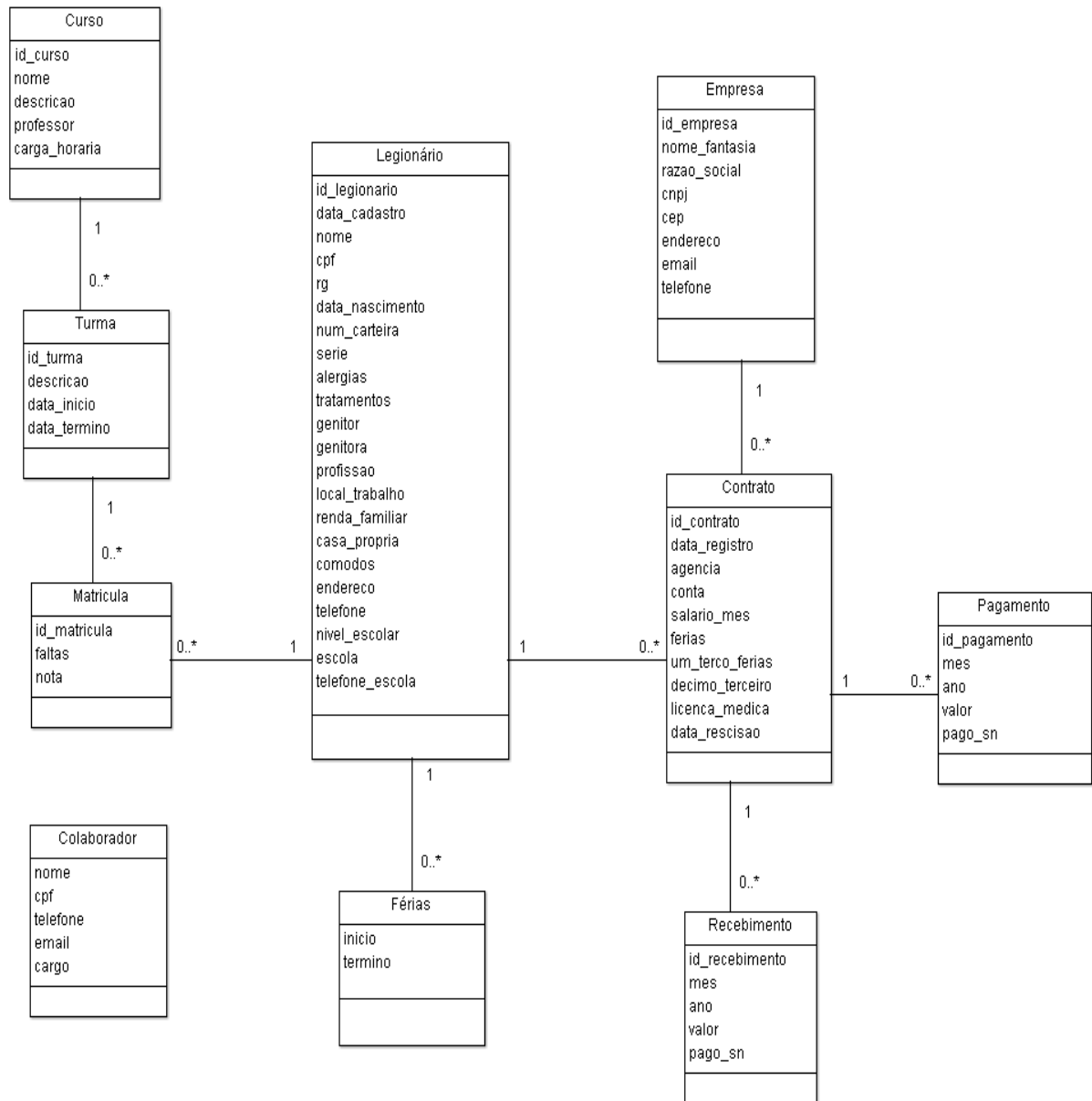


Figura 22 – Diagrama de classes

3.6 DIAGRAMA ENTIDADE-RELACIONAMENTO

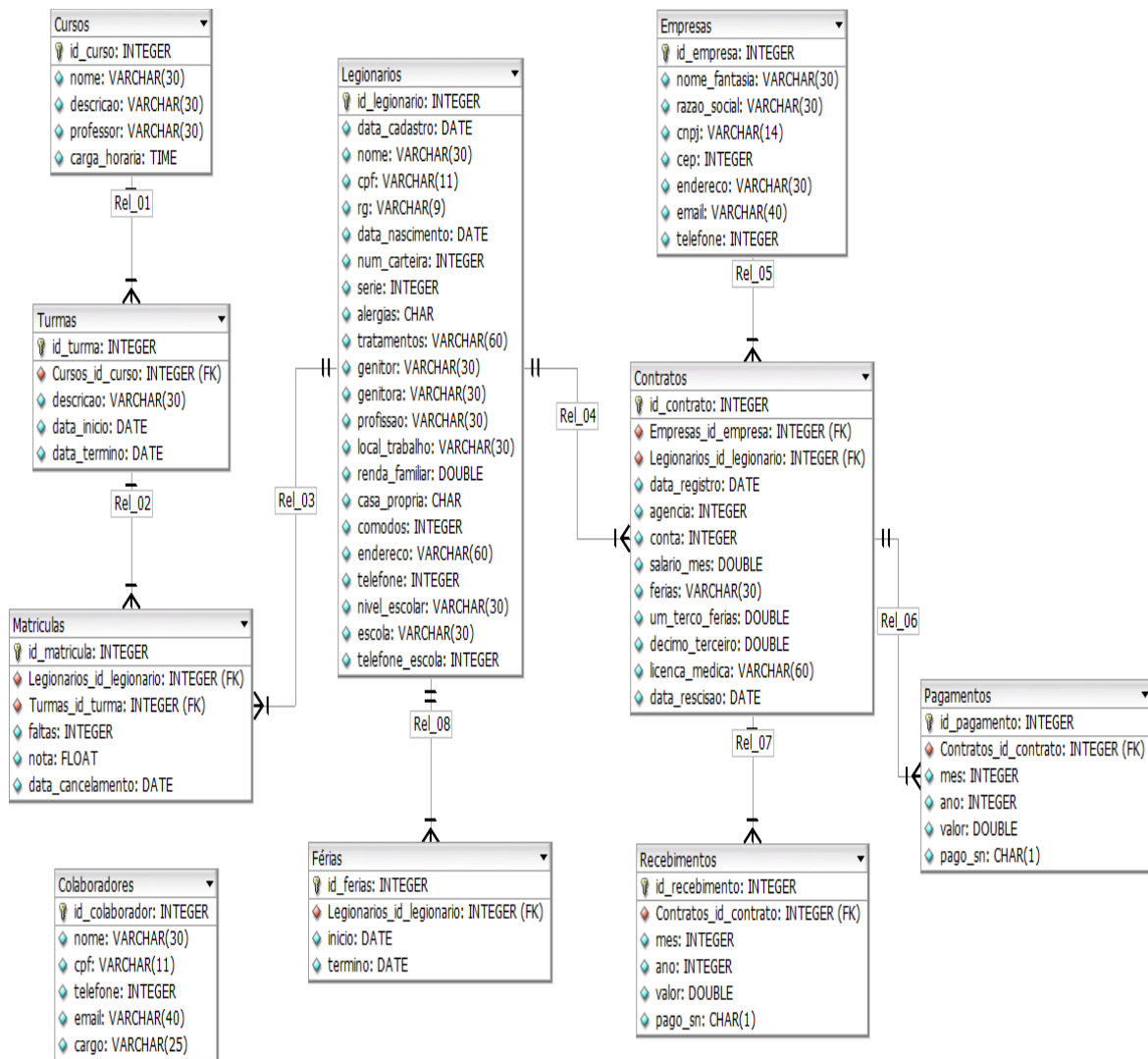


Figura 23 – Diagrama Entidade-Relacionamento

4. CONCLUSÃO

Gerenciar uma organização de forma ágil requer mecanismos que auxiliem nos processos organizacionais, tais como softwares que possam controlar o fluxo de dados que a organização gera no decorrer de seu trabalho.

Com esse intuito o sistema apresentado nesse trabalho dará mais agilidade nos processos realizados pela empresa, controlará os dados dos alunos, empresas parceiras e cursos, atendendo as necessidades específicas da empresa para que ela não necessite mais utilizar fichas de papel para fazer seu controle. Para isso foi feito todo o processo de análise para que o sistema atendesse os requisitos específicos desta organização e utilizado tecnologias capazes de dar todo o suporte a implementação do sistema.

Fica como projeto futuro a expansão do sistema para atender outras áreas da empresa de forma que fique englobado em um único sistema todas as atividades que realizam.

REFERÊNCIAS

BATISTA, Emerson de O. **Sistemas de Informação: o uso consciente da tecnologia para o gerenciamento**. 1. ed. São Paulo: Editora Saraiva, 2006.

BAUER, Christian e KING, Gavin. **Java Persistence com Hibernate**. 1. ed. Tradução de Gustavo Vinocur. Rio de Janeiro: Editora Ciência Moderna Ltda., 2007.

BOEKEL, Rafael Van. **Por dentro do PrimeFaces 2.2 – Artigo Revista Java Magazine 93**. Disponível em <<http://www.devmedia.com.br/por-dentro-do-primefaces-2-2-artigo-revista-java-magazine-93/21651>>. Acesso em: 29 jan. 2015.

COLEMAN, Derek. **Desenvolvimento orientado a objetos: o método fusion**. Tradução de Geraldo Costa Filho. Rio de Janeiro: Campus, 1996.

DEITEL, Paul; DEITEL Harvey. **Java Como Programar**. 8. ed. São Paulo: Pearson Prentice Hall, 2010.

GONÇALVES, Edson. **JSP, Servlets, JavaServer Faces, Hibernate, EJB3 Persistence e AJAX**. 1. ed. Rio de Janeiro: Editora Ciência Moderna Ltda., 2007.

HEMRAJANI, ANIL. **Desenvolvimento Ágil em Java com Spring, Hibernate e Eclipse**. 1. ed. São Paulo: Pearson Prentice Hall, 2007.

Como posso começar a desenvolver programas Java com o Java Development Kit (JDK)? Disponível em <https://www.java.com/pt_BR/download/faq/develop.xml>. Acesso em 06 de março de 2014.

Why MySQL? Disponível em <<http://www.mysql.com/why-mysql/>>. Acesso em: 10 fev. 2015.