



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

**RAFAEL NATELLI BILCHE**

**SISTEMA DE GESTÃO E AVALIAÇÃO DE ALUNOS EM ACADEMIAS**

**RAFAEL NATELLI BILCHE**

**SISTEMA DE GESTÃO E AVALIAÇÃO DE ALUNOS EM ACADEMIAS**

Trabalho de Conclusão de Curso apresentado ao Curso de Bacharelado Tecnólogo em Análise e Desenvolvimento de Sistemas do Instituto Municipal do Ensino Superior de Assis – IMESA e Fundação Educacional do Município de Assis – FEMA, como requisito para a obtenção do Certificado de Conclusão.

**Orientadora:** Me. Diomara Martins Reigato Barros

**Área de Concentração:** Análise e Desenvolvimento de Sistemas

# **SISTEMA DE GESTÃO E AVALIAÇÃO DE ALUNOS EM ACADEMIAS**

**RAFAEL NATELLI BILCHE**

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

**Orientador:** \_\_\_\_\_

**Diomara Martins Reigato Barros**

**Examinador:** \_\_\_\_\_

**Osmar Aparecido Machado**

## **AGRADECIMENTO**

Agradeço primeiramente à Deus por me dar disposição para superar todas as dificuldades.

Agradeço à (FEMA), Fundação Educacional do Município de Assis, por proporcionar um ensino de qualidade, na qual pude usufruir de todos os recursos disponíveis durante o curso de Análise e Desenvolvimento de Sistemas.

Agradeço também a minha família, pelo apoio e ajuda nos momentos mais difíceis que enfrentei antes, durante e agora nestes momentos finais do curso para obtenção do grau de Bacharel.

Agradeço aos meus amigos de sala, pelos anos de companheirismo que foram compartilhados entre inúmeras situações, desde o início da faculdade até os dias de hoje.

Agradeço à minha esposa Donária Cristina Theodoro que, além do carinho incondicional, sempre me apoiou e me ajudou, desde do início dessa jornada.

Agradeço a Professora Diomara Martins Reigato Barros, por toda orientação prestada e por estar sempre à disposição nos momentos de dúvidas e dificuldades.

## RESUMO

Este projeto tem como objetivo desenvolver um sistema informatizado que auxiliará os professores e Gestor da Academia na avaliação dos alunos, utilizando várias ferramentas tecnológicas, PHP, MySQL, Java Script, Html5, Ajax, NetBeans, Bootstrap, com a finalidade de guardar informações convenientes para o acompanhamento e desempenho do cliente, e também tendo total controle administrativo da empresa, como as matrículas, mensalidades, produtos, serviços e relatórios através dos registros.

**Palavras Chave:** PHP, MySQL, Java Script, Bootstrap.

## **ABSTRACT**

This project aims to develop a computerized system that will assist the teachers and the Academy Manager in evaluating the students, using various technological tools, PHP, MySQL, Java Script, Html5, Ajax, NetBeans, Bootstrap, in order to store information convenient for the monitoring and performance of the client, and also having full administrative control of the company, such as enrollments, monthly fees, products, services and reports through the records.

**Keywords:** PHP, MySQL, Java Script, Bootstrap.

## LISTA ILUSTRÇÕES

Figura 1: Mapa Mental.....	16
Figura 2: UC – Caso de uso .....	17
Figura 3: UC – Manter Aluno.....	18
Figura 4: UC – Manter Treino.....	19
Figura 5: UC – Manter Plano.....	20
Figura 6: UC – Manter Produto.....	22
Figura 7: UC – Mensalidade.....	23
Figura 8: UC – Manter Despesas.....	24
Figura 9: UC - Emitir Venda.....	25
Figura 10: UC – Pagar Mensalidade.....	26
Figura 11: UC – Manter Usuário.....	28
Figura 12: UC – Manter Funcionário.....	29
Figura 13: UC – Manter Professores.....	30
Figura 14: UC – Manter Fornecedor.....	31
Figura 15: UC – Acompanhar Pagamento.....	32
Figura 16: Diagrama de Atividade Manter Aluno.....	33
Figura 17: Diagrama de Sequencia Manter Aluno.....	34
Figura 18: Diagrama de Classe.....	35
Figura 19: Modelagem e Entidade de Relacionamento.....	36
Figura 20: Estrutura Analítica do Projeto.....	37
Figura 21: Cronograma.....	38

## **ABREVIATURAS E SIGLAS**

PHP	PERSONAL HOME PAGE
SQL	STRUCTURED QUERY LANGUAGE
CSS	CASCADING STYLE SHEETS
SGBD	SISTEMA DE GESTÃO DE BASE DE DADOS
HTML	HYPertext MARKUP LANGUAGE
UML	UNIFIED MODELING LANGUAGE
DML	DATA MANIPULATION LANGUAGE
ASP	ACTIVE SERVER PAGES
JSON	JAVASCRIPT OBJECT NOTATION
DER	DIAGRAMA ENTIDADE RELACIONAMENTO

# SUMÁRIO

<b>1. INTRODUÇÃO</b> .....	<b>11</b>
<b>1.1 OBJETIVO</b> .....	<b>12</b>
<b>1.2 JUSTIFICATIVA</b> .....	<b>12</b>
<b>1.3 MOTIVAÇÃO</b> .....	<b>12</b>
<b>1.4 PERSPECTIVA DE CONTRIBUIÇÃO</b> .....	<b>12</b>
<b>2. METODOLOGIA E FERRAMENTAS UTILIZADAS</b> .....	<b>13</b>
<b>2.1 UML</b> .....	<b>13</b>
<b>2.2 PHP</b> .....	<b>14</b>
<b>2.3 MYSQL</b> .....	<b>14</b>
<b>2.4 JAVASCRIPT</b> .....	<b>14</b>
<b>2.5 BOOTSTRAP</b> .....	<b>14</b>
<b>2.6 AJAX</b> .....	<b>15</b>
<b>2.7 HTML5</b> .....	<b>15</b>
<b>2.8 NETBEANS</b> .....	<b>15</b>
<b>3.ANÁLISE E ESPECIFICAÇÃO DO SISTEMA</b> .....	<b>16</b>
<b>3.1 MAPA MENTAL</b> .....	<b>16</b>
<b>3.2 DIAGRAMA CASO DE USO</b> .....	<b>17</b>
<b>3.3 NARRATIVAS</b> .....	<b>18</b>
3.3.1 manter aluno.....	18
3.3.2 manter treino .....	20
3.3.3 manter plano.....	21
3.3.4 emitir mensalidade.....	22
3.3.5 manter produto .....	23
3.3.6 manter mensalidade.....	24
3.3.7 manter despesas .....	25
3.3.8 efetuar vendas.....	26
3.3.9 pagar mensalidade .....	27
3.3.10 pagamento avista .....	28
3.3.11 cartão de credito .....	28
3.3.12 manter usuário .....	29
3.3.13 manter funcionário.....	30

3.3.14 manter professores .....	31
3.3.15 manter fornecedor .....	32
3.3.16 acompanhar pagamento.....	33
<b>3.4 DIAGRAMA DE ATIVIDADES .....</b>	<b>35</b>
3.4.1 diagrama de atividades manter aluno .....	35
<b>3.5 DIAGRAMA DE SEQUENCIA.....</b>	<b>36</b>
3.5.1 diagrama de sequência manter aluno .....	36
<b>3.6 DIAGRAMA DE CLASSE.....</b>	<b>37</b>
<b>3.7 MODELAGEM ENTIDADE E RELACIONAMENTO .....</b>	<b>38</b>
<b>4. ESTRUTURA ANALÍTICA DO PROJETO .....</b>	<b>39</b>
<b>4.1 ORÇAMENTO.....</b>	<b>40</b>
4.1.1 especificação de custo .....	40
<b>5. TELAS DO SISTEMA.....</b>	<b>41</b>
<b>6. CONCLUSÃO .....</b>	<b>43</b>
<b>REFERENCIAS .....</b>	<b>44</b>

## 1. INTRODUÇÃO

A prática de exercícios físicos tem muita importância para nosso corpo, ajudando na melhoria da força, dos músculos e da flexibilidade, fortalecimento dos ossos e das articulações. Deste modo pessoas preocupadas com sua saúde buscam academias para realizarem exercícios físicos com profissionais capacitados para auxiliarem na alimentação e em um programa de exercícios adequado ao seu corpo para manter uma boa forma. Com o crescimento de pessoas interessadas em manter a saúde e ter um programa adequado de exercício, também cresce o número de academias.

Todas as práticas de exercícios físicos são importantes para saúde do nosso corpo, a tecnologia vem com fim de auxiliar, agregar e inovar. Uma dessas maneiras de se incorporar a tecnologia em seus objetivos e por meio de um Sistema de Gestão que auxiliará diretamente o público alvo.

Para Lacruz (2008), “Planejar detalhadamente significa antever nos mínimos detalhes o resultado futuro de ações que se pretendem tomar acerca de um empreendimento, objetivando indicar sua viabilidade ou inviabilidade”.

Integrar as áreas de uma academia é uma saída para evitar problemas futuros. Assim, torna-se possível acertar na hora de tomar decisões importantes, como conservação e manutenção da infraestrutura; treinamentos para a equipe e ajuste de processos. Um bom Software de Gestão atuará ao lado de seus colaboradores para criar uma comunicação efetiva com cada um. Além de ter maior conhecimento sobre o que acontece em cada área, também tendo um controle total nos processos internos.

Segundo Venliones (2005), o autor cita que através de um planejamento estratégico o gestor da academia adquire ferramentas para enxergar as ameaças e principalmente as oportunidades que o mercado e a sociedade oferecem para seu negócio. Além de poder identificar quais as fraquezas e as forças do seu negócio para saber exatamente quais os pontos a melhorar e aprimorar.

## 1.1 OBJETIVO

O objetivo é oferecer uma forma de que donos de academias, administradores, possam facilitar a decisão e estratégias a serem tomadas. Apresentando ferramentas e recursos para inserção dos dados como, matrículas de alunos, cadastro de funcionários, turmas e controle de pagamentos, dentre outras opções, permitindo um total controle de toda parte administrativa da empresa. O sistema vem pra oferecer segurança e eficácia em todos os controles necessários para administração da Academia.

## 1.2 JUSTIFICATIVA

O estudo consiste em informar sobre benefícios da utilização de um sistema de informação nas academias, para melhor desempenho do setor, assim tendo melhor gerencia do negócio. A tecnologia da informação agrega como instrumento que auxiliara o negócio em suas metas, ajudando a chegar a eficiência em suas operações e processos.

## 1.3 MOTIVAÇÃO

Devido a necessidade de administração da academia, decidiu-se desenvolver um sistema de gerenciamento visando integração total dos dados coletados, permitindo um total controle da parte administrativa, com motivação de futuramente implementar o sistema em outras empresas do segmento.

## 1.4 PERSPECTIVA DE CONTRIBUIÇÃO

Ao final do desenvolvimento deste trabalho, o mesmo será publicado no formato de artigo e divulgado na instituição, com o objetivo de promover e compartilhar os conhecimentos e resultados alcançados. A FEMA, disponibilizará em seu acervo digital, o que possibilitará uma comparação entre a tecnologia estudada, com o objetivo de contribuir com futuros projetos na área.

## 2. METODOLOGIA E FERRAMENTAS UTILIZADAS

Para o desenvolvimento do projeto será utilizado o conceito UML de modelagem para projetos com diagrama de casos de uso, narrativas, etc. A linguagem utilizada para o desenvolvimento será o PHP que fornece segurança e é uma das linguagens mais utilizadas para desenvolvimento web na atualidade. Para front-end será utilizada HTML5 junto com CSS para o designer e framework BOOTSTRAP. A interface entre o front-end (HTML5) e back-end (PHP) será utilizada AJAX e JAVASCRIPT. O banco de dados utilizado será o MYSQL.

### 2.1 UML

Para Booch et al (2016). UML é utilizada para elaborar estruturas de um projeto. Podendo ser usada para a visualizar, especificar, construir e documentar sistemas. Na Engenharia de Software, ela é usada como uma linguagem de modelagem permitindo representação de um sistema de forma padronizada, facilitando a compreensão antes da implementação. UML é usada para modelar sistemas, cuja seu alcance pode incluir diversos sistemas de informação, como corporativos, aplicações na Web e até sistemas mais complexos. Esta linguagem muito persuasiva, atende todas as visões necessárias para desenvolvimento e implantação de sistemas de informação.

A UML não é uma metodologia de desenvolvimento, ela não nos dá o passo a passo para o desenvolvimento de um projeto, mas te dá um auxílio para ter visão do cenário e a comunicação dos objetos, assim podendo identificar alguns processos.

## 2.2 PHP

Segundo Dall'Oglio (2018), A Linguagem PHP tem interpretação livre, usada inicialmente para o desenvolver aplicações, atua ao lado de um servidor, capacidade de criação de conteúdo eficiente na World Wide Web. É uma das primeiras linguagens na inserção de documentos HTML, assim deixando de utilizar arquivos externos para processamento de dados. Interpretação do código pelo próprio módulo do PHP, gerando a página web a ser visualizada pelo usuário. A evolução da linguagem tem como características oferecer funcionalismo em linha de comando, ganhando algumas outras características a mais, possibilitando usos adicionais do PHP que não tenha relação com sistemas web.

## 2.3 MYSQL

Segundo Milani (2007), MySQL é um Sistema de Banco de Dados que disponibiliza seu código fonte aberto muito popular no mundo da programação utilizada no mundo todo. Se torna popular pelo seu desempenho, confiabilidade e facilidade de uso, o MySQL tornou-se a principal escolha de banco de dados para aplicativos baseados na web.

## 2.4 JAVASCRIPT

Segundo Duckett (2016), JAVASCRIPT é uma linguagem de programação interpretada a orientação a objetos, ela é reconhecida como scripting para páginas Web, também sendo muito útil em vários outros ambientes que não se usa navegador. Linguagem em scripting que se baseia em protótipos, suportando a programação orientada a objetos.

## 2.5 BOOTSTRAP

Para Souza (2018) Bootstrap é um framework CSS de código-fonte aberto repleto de componentes prontos e utilizáveis, baseando em modelos de design, tornando-se melhor a experiência do usuário fazendo que o site se torne mais apresentável e responsivo.

## 2.6 AJAX

Ajax é uma sigla em inglês para Asynchronous JavaScript and XML e trata-se de técnica de carregamento de conteúdo em uma página web uso de JavaScript e XML, TXT, PHP, ASP, JSON ou qualquer linguagem de marcação ou programação capaz de ser recuperada de um servidor. (SILVA 2009).

## 2.7 HTML5

Segundo Samy Silva (2015) HTML é uma sigla inglês para Hyper Text Markup Language (Linguagem para marcação de hipertexto). Esta linguagem é que nós desenvolvedores utilizamos como forma de se comunicarmos com usuários fazendo que ele entenda, como uma página de um site.

## 2.8 NETBEANS

“O NetBeans é um ambiente de desenvolvimento, sendo ele multiplataforma, é uma ferramenta utilizada por programadores para criar, produzir, compilar, debugar e instalar aplicações. Ele fornece uma ampla base para tais criações, possuindo um vasto conjunto de bibliotecas, módulos e API's entre outros recursos necessários para o feito. Esta IDE conta também com amparo de uma vasta documentação produzida também em português. Esta é uma IDE produzida por total em java, mas que pode suportar qualquer outra linguagem de programação ou linguagem que desenvolva com Swing”. (SALOMÃO; MAGALHÃES, 2015)

### 3. ANÁLISE E ESPECIFICAÇÃO DO SISTEMA

Para o desenvolvimento da aplicação proposta, foi utilizada a tecnologia UML (Linguagem de Modelagem Unificada) para análise e validação dos requisitos, onde toda a parte gráfica foi criada na ferramenta Astah. Já para a criação do DER, foi utilizado o DBDesigner.

#### 3.1 MAPA MENTAL

O Mapa Mental tem grande importância para a implementação do sistema, ajudando na hora de escolher os requisitos que serão usados durante o desenvolvimento do projeto. Abaixo, terá a imagem do Mapa Mental do projeto que será elaborado, contendo 15 requisitos:

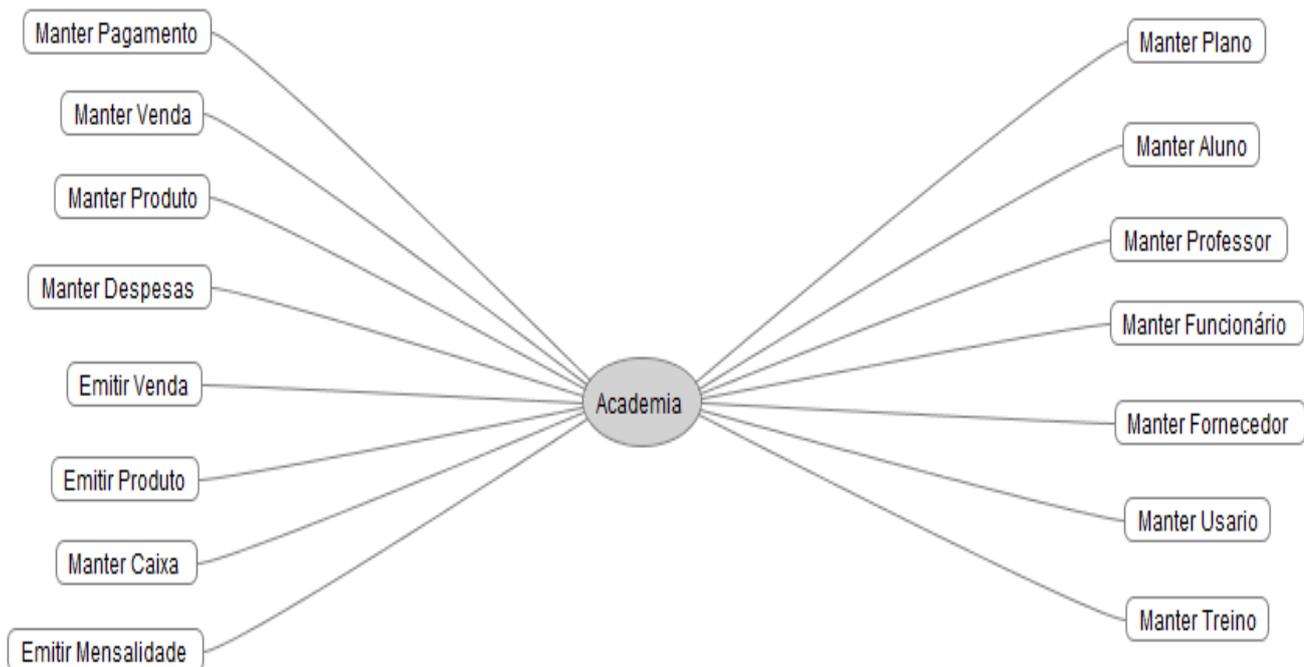


Figura 1: Mapa Mental

### 3.2 DIAGRAMA CASO DE USO

O Diagrama de caso de uso é uma descrição de todas as possíveis sequências de interações entre os atores e sistema. Ele descreve um cenário, que mostra passo a passo os eventos do sistema do ponto de vista do usuário. É um passo muito importante no desenvolvimento, pois através dele é feito o levantamento e registro dos requisitos de um sistema.

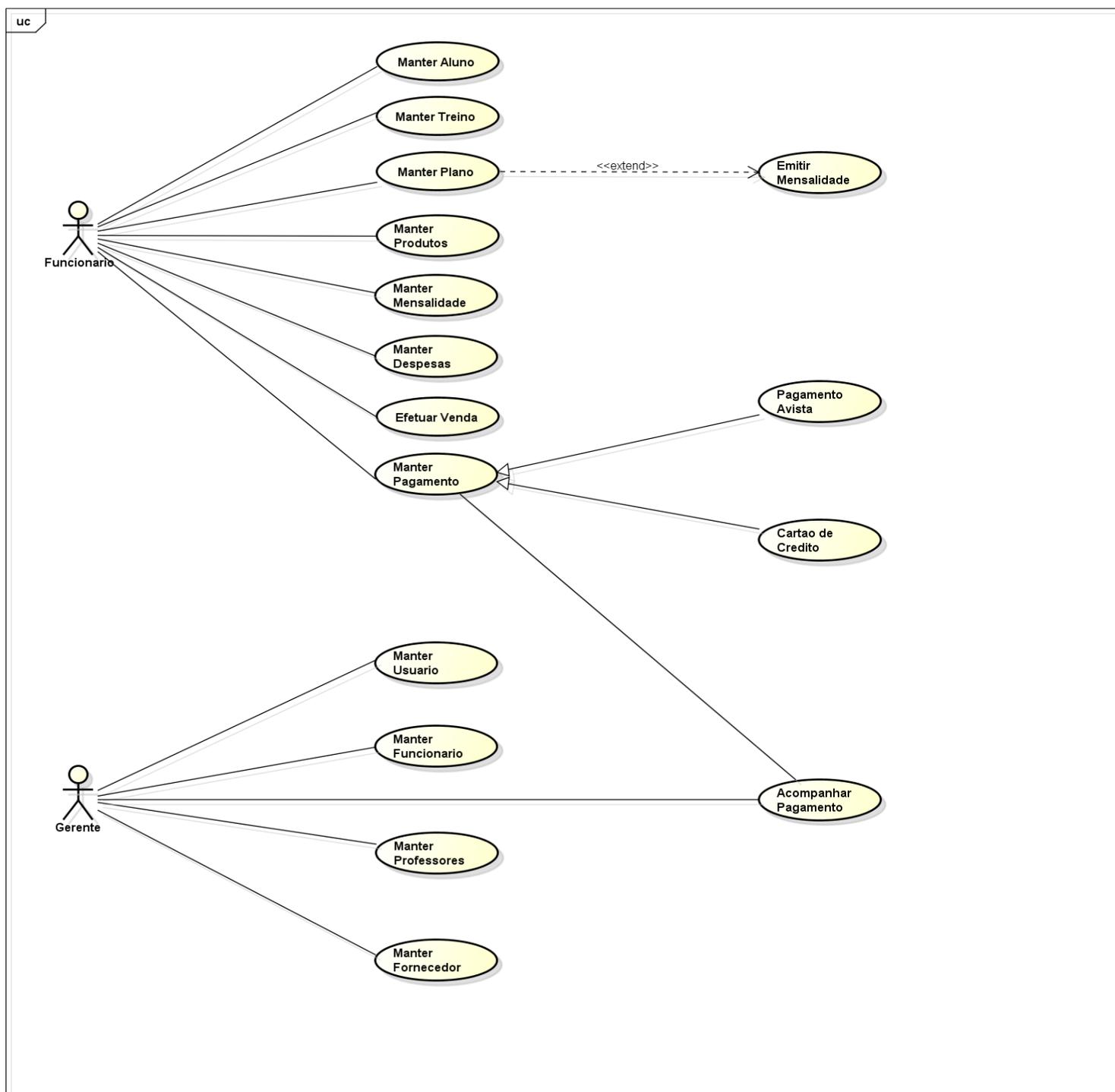


Figura 2: UC - Casos de Uso

### 3.3 NARRATIVAS

As Narrativas de caso de uso descrevem a funcionalidade proposta para um novo sistema que será projetado.

#### 3.3.1 Manter aluno

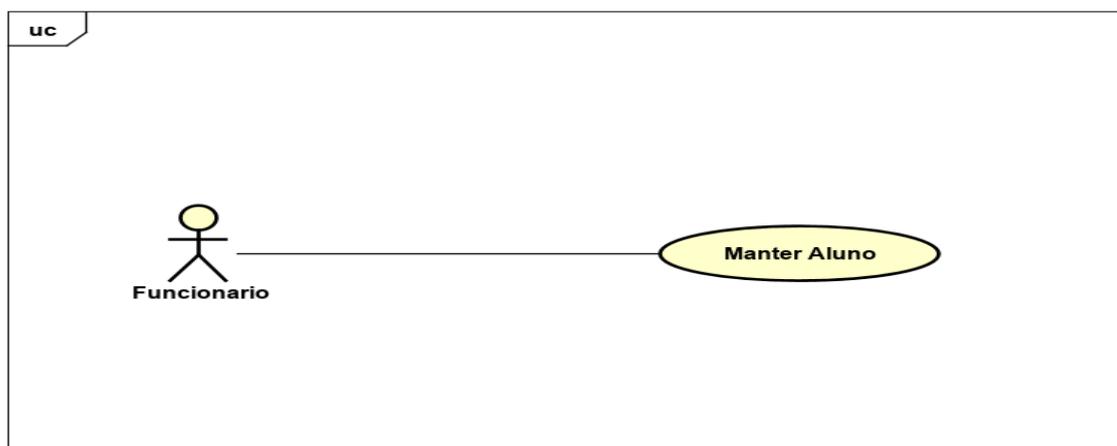


Figura 3: UC - Manter Aluno

<b>1. Finalidade/Objetivo</b>	O Caso de Uso “Manter Aluno” possibilita o usuário realizar algumas tarefas como cadastrar, remover e atualizar porem tudo isso são partes do objetivo maior, que é inserir novo aluno no sistema.
<b>2. Atores</b>	Funcionário
<b>3. Pré-condições</b>	
<b>4. Evento inicial</b>	O Funcionário Insere novo aluno.
<b>5. Fluxo principal</b>	<p>a) O funcionário entra no sistema.</p> <p>b) O funcionário realiza o cadastro, remoção e atualização de cadastros de alunos.</p> <p>c) Por fim, o funcionário confirma o cadastro de alunos no sistema (A1).</p> <p>d) Caso de Uso é encerrado.</p>
<b>6. Fluxos Alternativos</b>	<p><b>A1 – Não confirma cadastro.</b></p> <p>a) O funcionário não confirma cadastro.</p>

	<b>b)</b> O sistema segue para o passo d do fluxo principal.
--	--

### 3.3.2 Manter treino

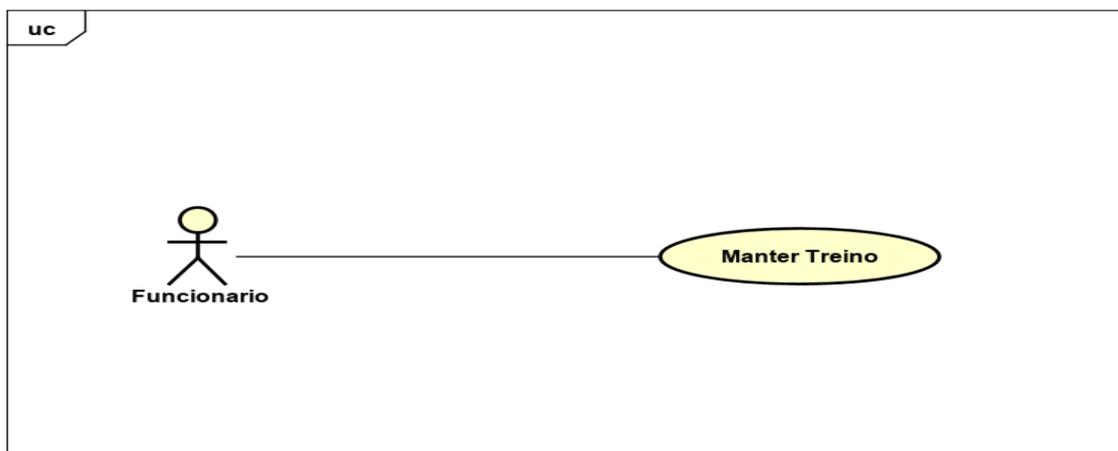


Figura 4: UC - Manter Treino

<b>1. Finalidade/Objetivo</b>	O Caso de Uso “Manter Treino” envolve algumas tarefas como, treino relacionado ao aluno, fazendo que tenha um acompanhamento para melhor desempenho do aluno nos treinos, porem tudo isso são partes do objetivo maior, que é ter o acompanhamento do treino do aluno.
<b>2. Atores</b>	Funcionário.
<b>3. Pré-condições</b>	O aluno solicita o acompanhamento do treino.
<b>4. Evento inicial</b>	O Funcionário cadastra treino
<b>5. Fluxo principal</b>	<p><b>a)</b> O funcionário entra no sistema.</p> <p><b>b)</b> O funcionário realiza o cadastro do treino recomendado.</p> <p><b>c)</b> O funcionário a pedido do aluno realiza consultas de treinos.</p> <p><b>d)</b> O funcionário faz o acompanhamento de desempenho do aluno.</p> <p><b>e)</b> Por fim, o funcionário escolhe a opção “Manter Treino” para acompanhar treino do aluno.</p>

### 3.3.3 Manter plano

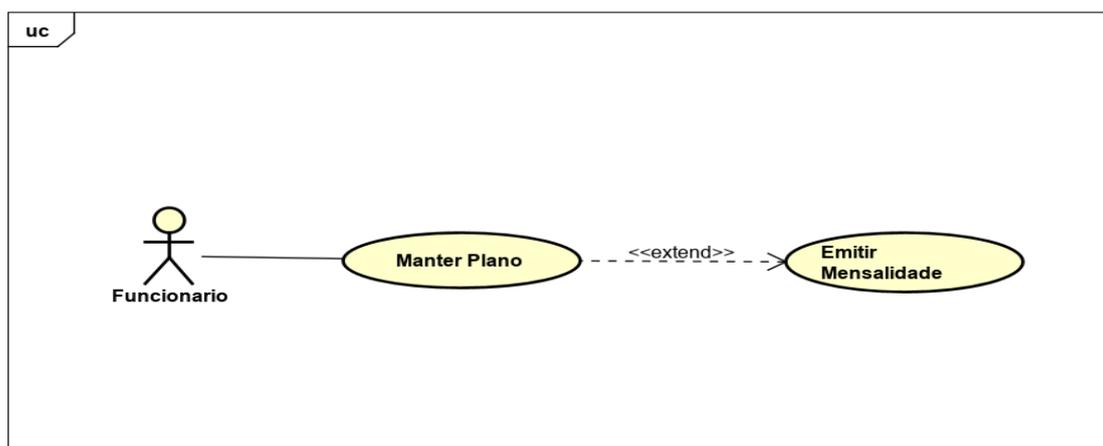


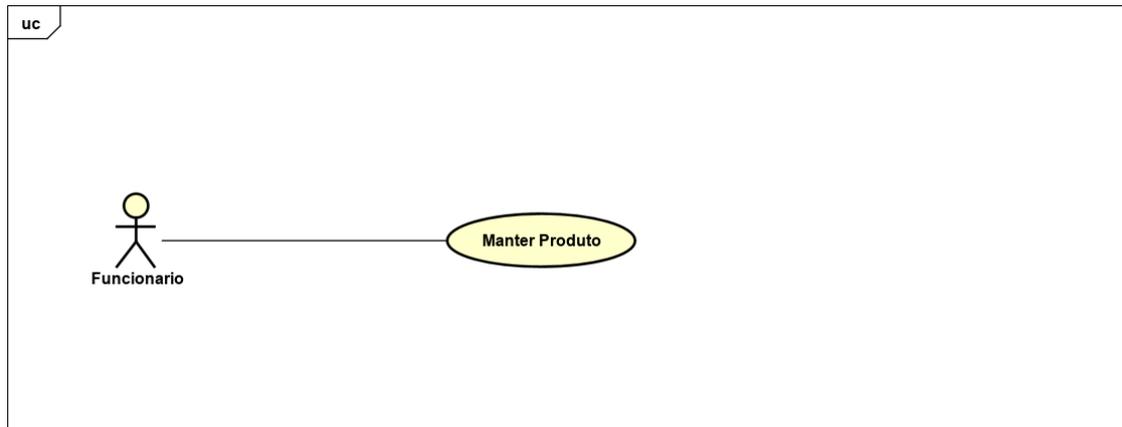
Figura 5: UC – Manter Plano

<b>1. Finalidade/Objetivo</b>	O Caso de Uso “Manter Plano” envolve a tarefa de escolher melhor plano de mensalidade para o aluno.
<b>2. Atores</b>	Funcionário.
<b>3. Pré-condições</b>	
<b>4. Evento inicial</b>	O Funcionária realiza escolhe do plano.
<b>5. Fluxo principal</b>	<p>a) O funcionário entra no sistema.</p> <p>b) O funcionário realiza cadastro do plano escolhido pelo cliente.</p> <p>c) Por fim, o funcionário confirma cadastro. (A1)</p> <p>d) <b>Caso de Uso é encerrado.</b></p>
<b>6. Fluxos Alternativos</b>	<p><b>A1 – Não confirma cadastro.</b></p> <p>a) O funcionário não confirma cadastro.</p> <p>b) O sistema segue para o passo d do fluxo principal.</p>

### 3.3.4 Emitir mensalidade

<b>1. Finalidade/Objetivo</b>	O Caso de Uso “Emitir Mensalidade” é referente ao plano escolhido pelo cliente, possibilitando a emissão da mensalidade, tudo isso são partes do objetivo maior, que é escolha do plano.
<b>2. Atores</b>	Funcionário.
<b>3. Pré-condições</b>	Cliente ter escolhido o plano desejado.
<b>4. Evento inicial</b>	O funcionário emite a mensalidade.
<b>5. Fluxo principal</b>	<p><b>a)</b> O funcionário entra no sistema.</p> <p><b>b)</b> O funcionário escolhe a melhor forma de pagamento para o cliente do plano escolhido.</p> <p><b>c)</b> O funcionário confirma emissão da mensalidade. <b>A1</b></p> <p><b>d)</b> Por fim, o cliente conclui a emissão da mensalidade.</p>
<b>6. Fluxos Alternativos</b>	<p><b>A1 – Não emitir a mensalidade.</b></p> <p><b>a)</b> O cliente tem opção de não emitir a mensalidade na hora da escolha do plano.</p> <p><b>b)</b> O sistema segue para o passo B do fluxo principal.</p>

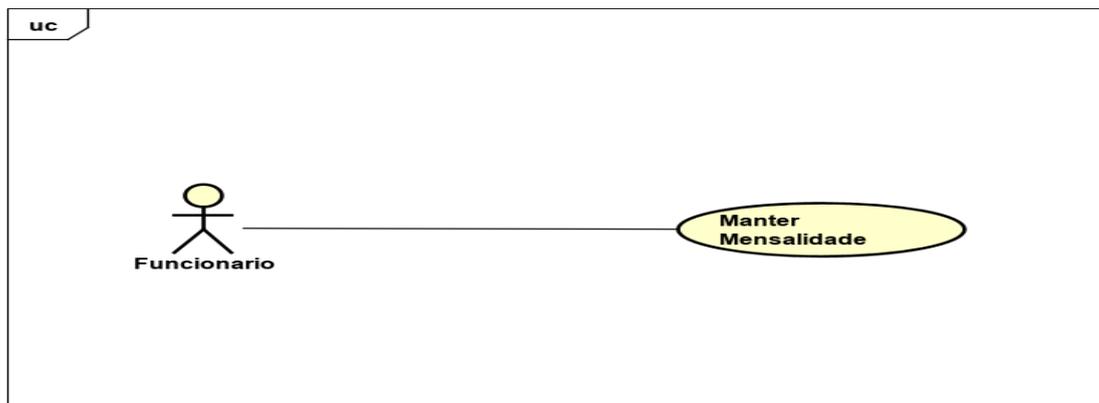
### 3.3.5 Manter produto



**Figura 6:** UC - Manter Produto

<b>1. Finalidade/Objetivo</b>	O Caso de Uso “Manter Produto” envolve o cadastro, baixa e controle de produtos no estoque.
<b>2. Atores</b>	Funcionário.
<b>3. Pré-condições</b>	
<b>4. Evento inicial</b>	O funcionário realiza cadastro de produtos.
<b>5. Fluxo principal</b>	<p><b>a)</b> O funcionário entra no sistema.</p> <p><b>b)</b> O funcionário cadastra os produtos.</p> <p><b>c)</b> O funcionário acompanha saída e entradas de produtos.</p> <p><b>d)</b> Por fim, o funcionário realiza pedidos de produtos que estão em falta.</p>

### 3.3.6 Manter mensalidade



**Figura 7:** UC - Manter Mensalidade

<b>1. Finalidade/Objetivo</b>	O Caso de Uso “Manter Mensalidade” envolve a tarefa de realizar o cadastro do cliente para emissão de mensalidade.
<b>2. Atores</b>	Funcionário.
<b>3. Pré-condições</b>	Cliente confirma escolha de um plano.
<b>4. Evento inicial</b>	O funcionário realiza cadastro de um novo cliente para gerar sua mensalidade.
<b>5. Fluxo principal</b>	<p><b>a)</b> O funcionário entra no sistema.</p> <p><b>b)</b> O funcionário realiza cadastro da mensalidade para o cliente.</p> <p><b>c)</b> O funcionário confirma os dados de todo o cadastro.</p> <p><b>d)</b> O funcionário realiza a escolha de melhor forma de pagamento ao cliente.</p> <p><b>e)</b> Por fim, o funcionário conclui o cadastro de mensalidade.</p>
<b>6. Fluxos Alternativos</b>	

### 3.3.7 Manter despesas

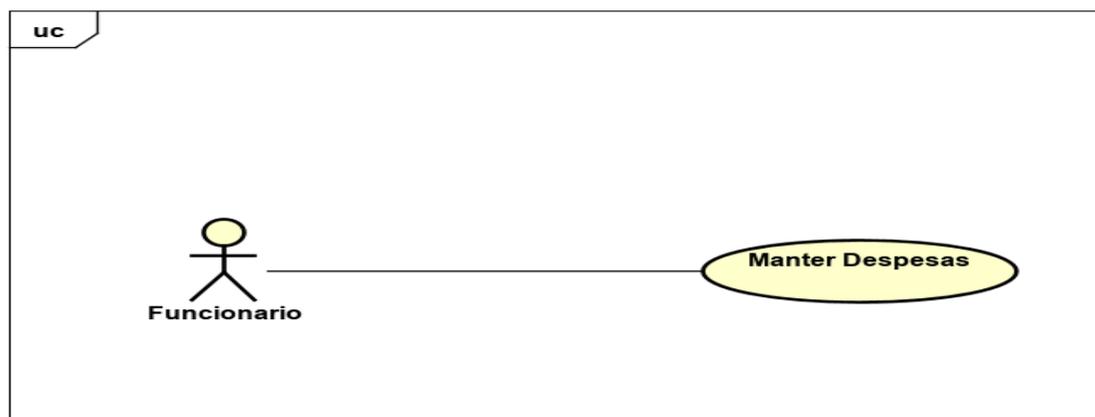
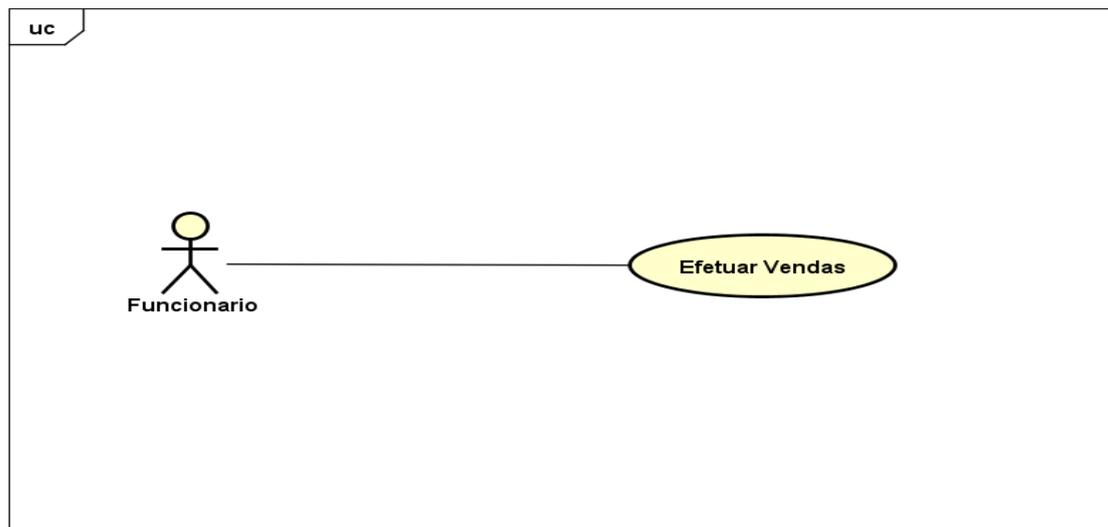


Figura 8: UC - Manter Despesas

<b>1. Finalidade/Objetivo</b>	O Caso de Uso “Manter Despesas” envolve o cadastro de todas as despesas da empresa no sistema.
<b>2. Atores</b>	Funcionário
<b>3. Pré-condições</b>	
<b>4. Evento inicial</b>	O funcionário realiza cadastro das despesas.
<b>5. Fluxo principal</b>	<p><b>a)</b> O funcionário entra no sistema.</p> <p><b>b)</b> O funcionário realiza cadastro das despesas no sistema.</p> <p><b>c)</b> Por fim, o funcionário faz acompanhamento no sistema das despesas da empresa.</p>
<b>6. Fluxos Alternativos</b>	

### 3.3.8 Efetuar vendas



powered by Astah

Figura 9: UC Efetuar Vendas

<b>1. Finalidade/Objetivo</b>	O Caso de Uso “Efetuar Vendas” envolve a tarefa em efetivar a compra feita pelo cliente.
<b>2. Atores</b>	Funcionário.
<b>3. Pré-condições</b>	
<b>4. Evento inicial</b>	O Funcionário concluir uma compra.
<b>5. Fluxo principal</b>	<p><b>a)</b> O funcionário entra no sistema.</p> <p><b>b)</b> O funcionário realiza cadastro de venda para o cliente.</p> <p><b>c)</b> Por fim, o funcionário efetiva as vendas para o cliente. <b>A1</b></p>
<b>6. Fluxos Alternativos</b>	<p><b>A1 – Não efetivar vendas</b></p> <p><b>a)</b> O funcionário não consegue efetivar as vendas.</p> <p><b>b)</b> O sistema segue para o passo A do fluxo principal.</p>

### 3.3.9 Pagar mensalidade

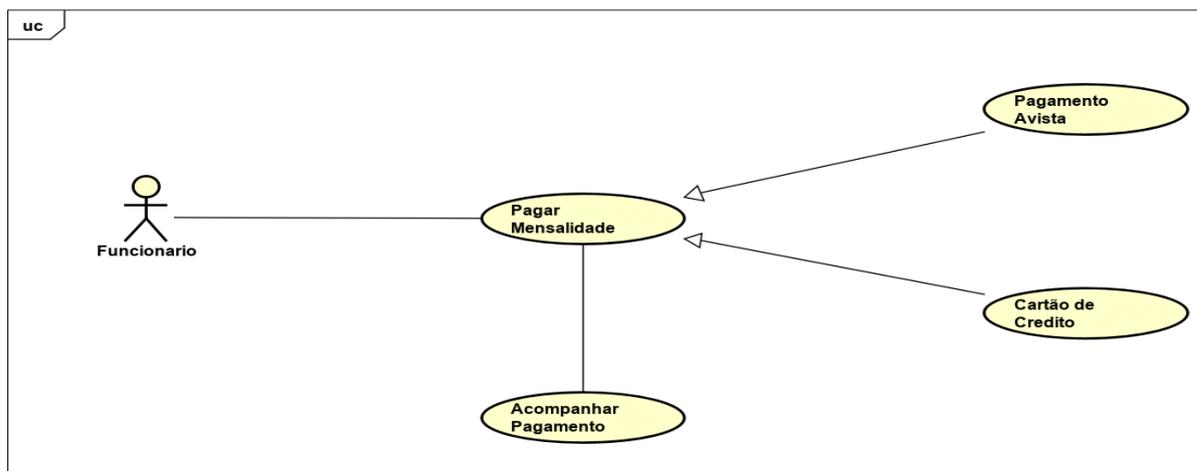


Figura 10: UC Pagar Mensalidade

<b>1. Finalidade/Objetivo</b>	No Caso de Uso “Pagar Mensalidade” o funcionário e gerente fica responsável em acompanhar a forma de pagamento.
<b>2. Atores</b>	Funcionário
<b>3. Pré-condições</b>	O cliente realiza pagamento.
<b>4. Evento inicial</b>	O funcionário gera o pagamento para o cliente.
<b>5. Fluxo principal</b>	<p>a) O funcionário entra no sistema.</p> <p>b) O funcionário realiza cadastro da mensalidade para pagamento.</p> <p>c) Por fim, o funcionário gera a mensalidade a pagar. <b>A1</b></p>
<b>6. Fluxos Alternativos</b>	<p><b>A1 - Não realiza o cadastro da mensalidade a pagar.</b></p> <p>a) Por algum motivo o funcionário não realiza cadastro de mensalidade.</p> <p>b) O funcionário segue para o passo A do fluxo principal.</p>

### 3.3.10 Pagamento avista

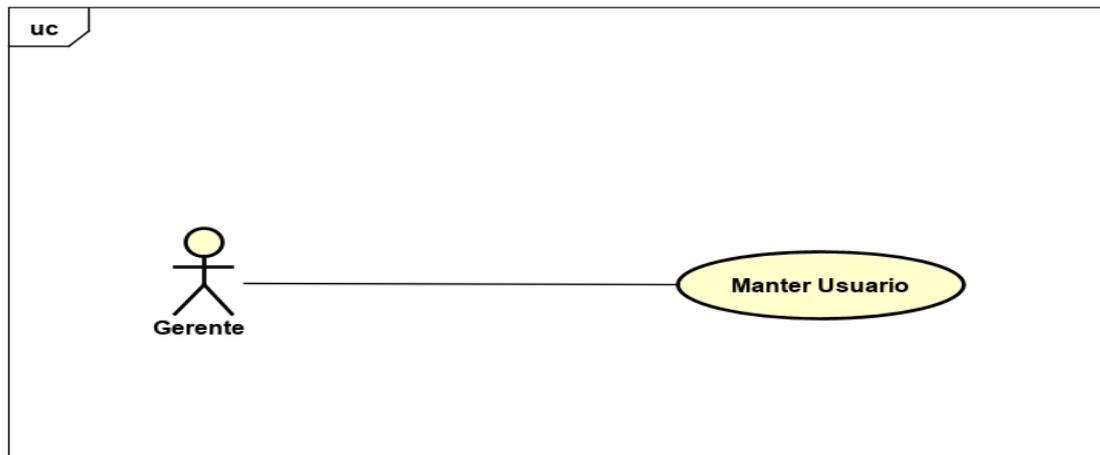
<b>1. Finalidade/Objetivo</b>	O Caso de Uso “Pagamento Avista” envolve o pagamento da mensalidade em dinheiro.
<b>2. Atores</b>	Funcionário
<b>3. Pré-condições</b>	Cliente realizar pagamento.
<b>4. Evento inicial</b>	O funcionário conclui a venda.
<b>5. Fluxo principal</b>	<p>a) O funcionário entra no sistema.</p> <p>b) O funcionário realiza a venda.</p> <p>c) O funcionário confirma o pagamento avista.</p>
<b>6. Fluxos Alternativos</b>	<p><b>A1 – Não confirma pagamento.</b></p> <p>a) O funcionário não confirma o pagamento.</p> <p>b) O sistema segue para o passo A do fluxo principal.</p>

### 3.3.11 Cartão de credito

<b>1. Finalidade/Objetivo</b>	No Caso de Uso “Cartão de Credito” envolve o pagamento com uso de cartão de credito.
<b>2. Atores</b>	Funcionário
<b>3. Pré-condições</b>	Cliente realiza pagamento.
<b>4. Evento inicial</b>	O funcionário conclui a venda.
<b>5. Fluxo principal</b>	<p>a) O funcionário entra no sistema.</p> <p>b) O funcionário realiza a venda.</p> <p>c) O funcionário confirma o pagamento com cartão de credito. A1.</p>
<b>6. Fluxos Alternativos</b>	<p><b>A1 – Parcelamento do pagamento.</b></p> <p>a) O funcionário tem opção de realizar o pagamento parcelado.</p>

	<b>b)</b> O sistema segue para o passo A do fluxo principal.
--	--

### 3.3.12 Manter usuário



**Figura 11:** UC Manter Usuário

<b>1. Finalidade/Objetivo</b>	No Caso de Uso “Manter Usuário” o Gerente fica responsável em cadastrar, excluir e atualizar cadastros de usuários no sistema.
<b>2. Atores</b>	Gerente
<b>3. Pré-condições</b>	
<b>4. Evento inicial</b>	O gerente realiza cadastro de usuários no sistema.
<b>5. Fluxo principal</b>	<p><b>a)</b> O Gerente entra no sistema.</p> <p><b>b)</b> O gerente realiza cadastro de novos usuários no sistema.</p> <p><b>c)</b> O gerente realiza atualizações em cadastros já existentes no sistema.</p> <p><b>d)</b> O ator gerencia todos os usuários do sistema.</p> <p>d) O gerente confirma o cadastro de usuários.</p> <p><b>A1.</b></p>

<b>6. Fluxos Alternativos</b>	<b>A1 – Não confirmando cadastro</b> <b>a)</b> O gerente não consegue realizar cadastro. <b>b)</b> O sistema segue para o passo A do fluxo principal.
-------------------------------	---

### 3.3.13 Manter funcionário

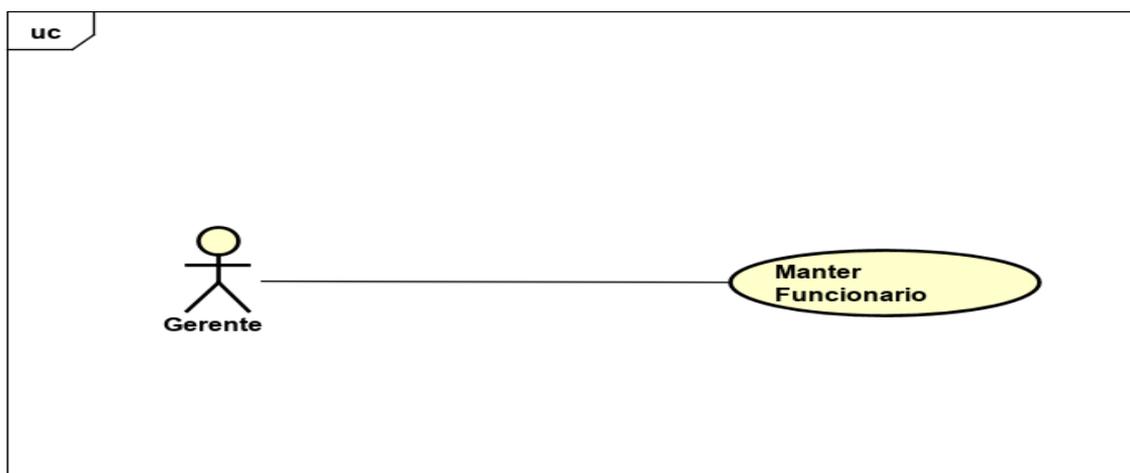


Figura 12: UC Manter Funcionário

<b>1. Finalidade/Objetivo</b>	No Caso de Uso “Manter Funcionário” o Gerente fica responsável em cadastrar, excluir e atualizar cadastros de funcionários no sistema.
<b>2. Atores</b>	Gerente
<b>3. Pré-condições</b>	
<b>4. Evento inicial</b>	O gerente realiza cadastro de funcionários no sistema.
<b>5. Fluxo principal</b>	<b>a)</b> O Gerente entra no sistema. <b>b)</b> O gerente realiza cadastro de novos funcionários no sistema. <b>c)</b> O gerente realiza atualizações em cadastros já existentes no sistema.

	<p><b>d)</b> O ator gerencia todos os cadastros do sistema.</p> <p>d) O gerente confirma o cadastro de funcionário. <b>A1.</b></p>
<b>6. Fluxos Alternativos</b>	<p><b>A1 – Não confirmando cadastro</b></p> <p><b>a)</b> O gerente não consegue realizar cadastro.</p> <p><b>b)</b> O sistema segue para o passo A do fluxo principal.</p>

### 3.3.14 Manter professores

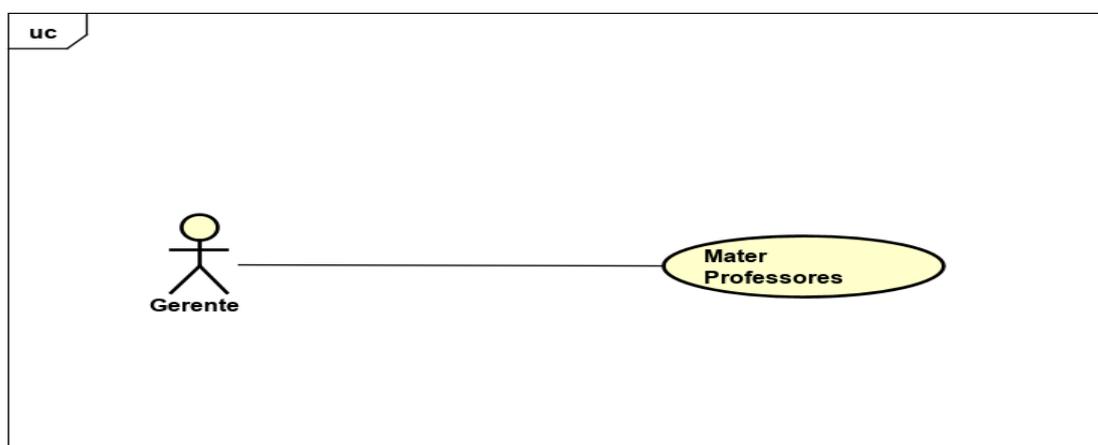


Figura 13: UC Manter Professores

<b>1. Finalidade/Objetivo</b>	No Caso de Uso “Professores” o Gerente fica responsável em cadastrar, excluir e atualizar cadastros de professores no sistema.
<b>2. Atores</b>	Gerente
<b>3. Pré-condições</b>	
<b>4. Evento inicial</b>	O gerente realiza cadastro de professores no sistema.
<b>5. Fluxo principal</b>	<p><b>a)</b> O Gerente entra no sistema.</p> <p><b>b)</b> O gerente realiza cadastro de novos professores no sistema.</p> <p><b>c)</b> O gerente realiza atualizações em cadastros já existentes no sistema.</p>

	<p><b>d)</b> O ator gerencia todos os cadastros do sistema.</p> <p>d) O gerente confirma o cadastro de professores. <b>A1.</b></p>
<b>6. Fluxos Alternativos</b>	<p><b>A1 – Não confirmando cadastro</b></p> <p><b>a)</b> O gerente não consegue realizar cadastro.</p> <p><b>b)</b> O sistema segue para o passo A do fluxo principal.</p>

### 3.3.15 Manter fornecedor

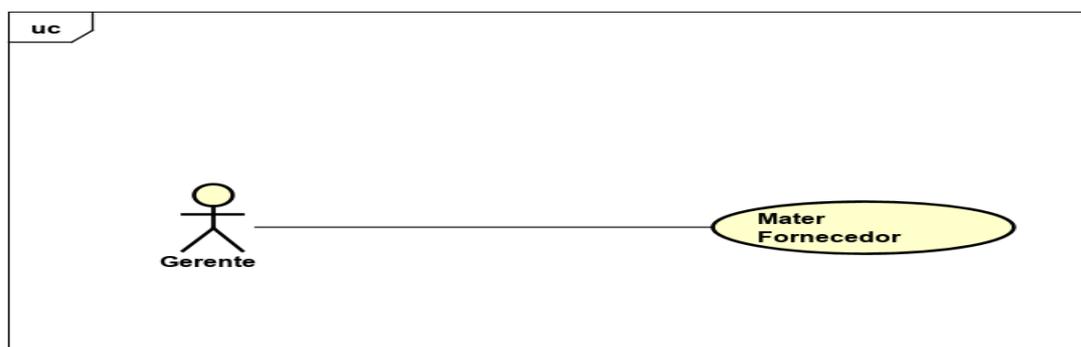


Figura 14: UC Manter Fornecedor

<b>1. Finalidade/Objetivo</b>	No Caso de Uso “Manter Fornecedor” o Gerente fica responsável em cadastrar, excluir e atualizar cadastros de fornecedores no sistema.
<b>2. Atores</b>	Gerente
<b>3. Pré-condições</b>	
<b>4. Evento inicial</b>	O gerente realiza cadastro de fornecedores.
<b>5. Fluxo principal</b>	<p><b>a)</b> O Gerente entra no sistema.</p> <p><b>b)</b> O gerente realiza cadastro de novos fornecedores.</p> <p><b>c)</b> O gerente realiza atualizações em cadastros já existentes no sistema.</p>

	<p><b>d)</b> O ator gerencia todos os cadastros do sistema.</p> <p>d) O gerente confirma o cadastro de fornecedores. <b>A1.</b></p>
<b>6. Fluxos Alternativos</b>	<p><b>A1 – Não confirmando cadastro</b></p> <p><b>a)</b> O gerente não consegue realizar cadastro.</p> <p><b>b)</b> O sistema segue para o passo A do fluxo principal.</p>

### 3.3.16 Acompanhar pagamento

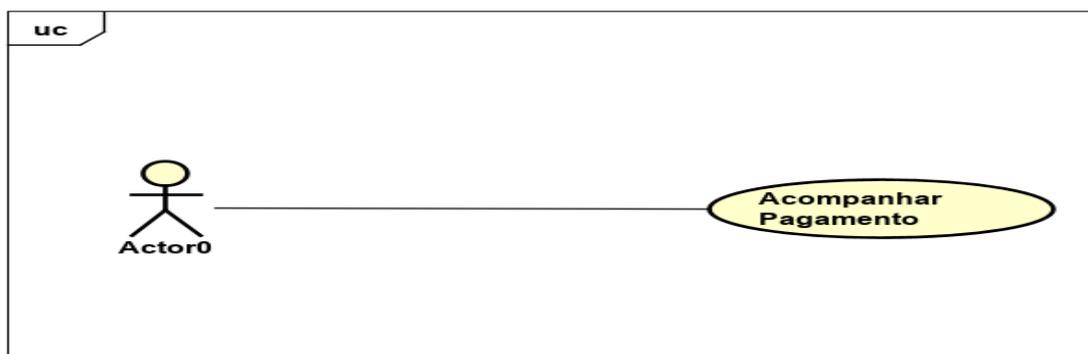


Figura 15: Acompanhar Pagamento

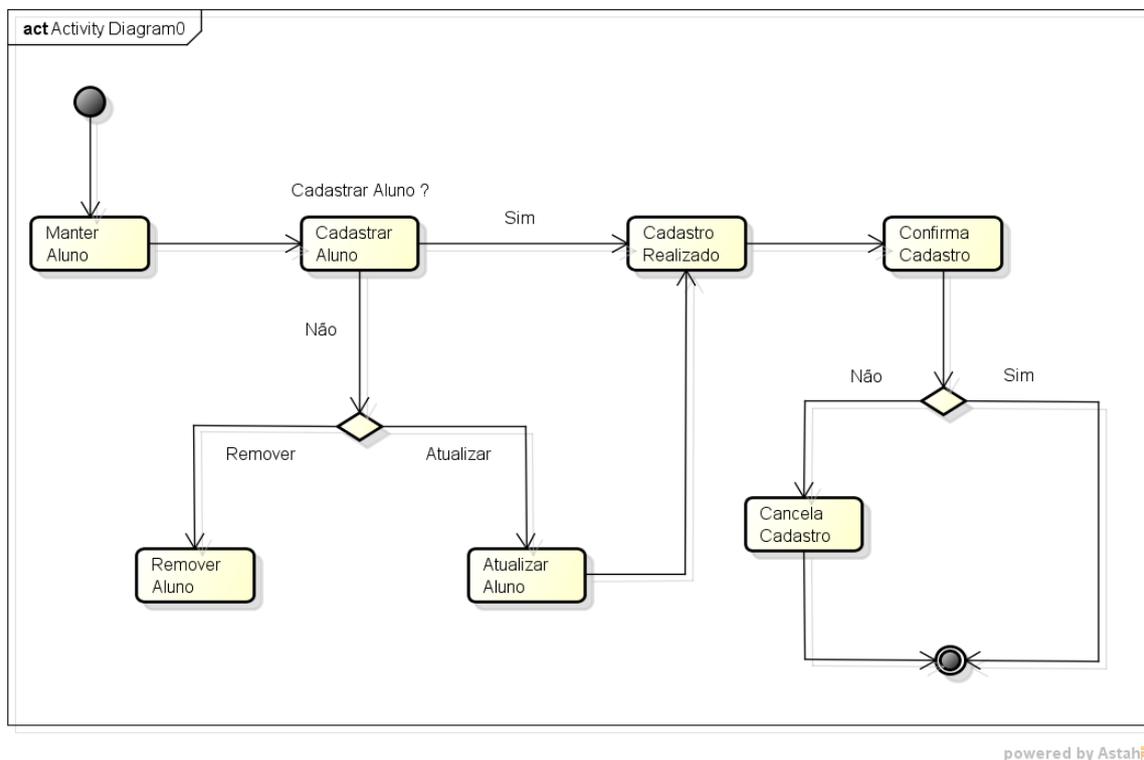
<b>1. Finalidade/Objetivo</b>	No Caso de Uso “Acompanhar Pagamento” o Gerente fica responsável em acompanhar pagamentos realizados.
<b>2. Atores</b>	Gerente
<b>3. Pré-condições</b>	Cliente realiza pagamento.
<b>4. Evento inicial</b>	O gerente acompanha pagamentos realizados.
<b>5. Fluxo principal</b>	<p><b>a)</b> O Gerente entra no sistema.</p> <p><b>b)</b> O gerente realiza acompanhamento dos pagamentos realizados.</p> <p><b>c)</b> O gerente pode realizar descontos ou alterações no valor a serem pagos.</p> <p><b>d)</b> Por fim, gerente confirma pagamento.</p>

<b>6. Fluxos Alternativos</b>	<b>A1 – Não confirmando pagamento.</b> <b>a)</b> O gerente não confirma o pagamento. <b>b)</b> O sistema segue para o passo A do fluxo principal.
-------------------------------	---

### 3.4 DIAGRAMA DE ATIVIDADES

O diagrama de Atividades se preocupa em registrar os passos percorridos para a conclusão de uma atividade específica, podendo esta ser representada por um método com certo grau de complexidade, um algoritmo, ou mesmo por um processo completo. Ele concentra-se na representação do fluxo de controle de uma atividade.

#### 3.4.1 diagrama de atividades manter aluno

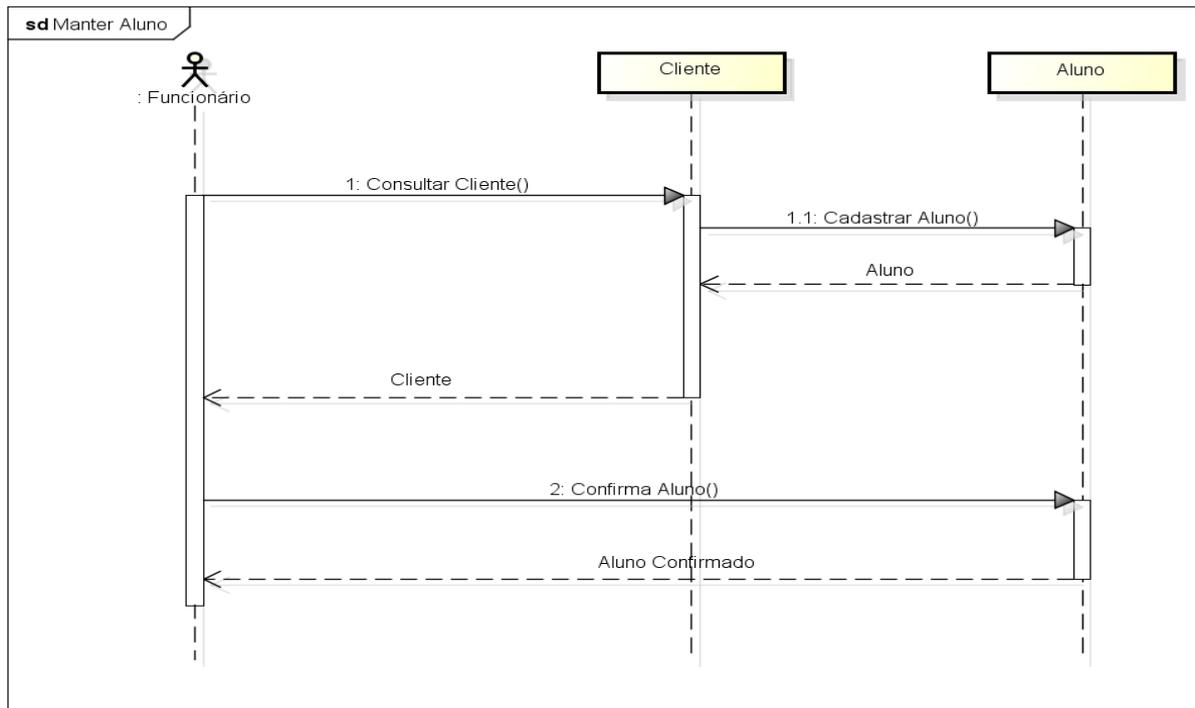


**Figura 16:** Diagrama de Atividade - Manter Aluno

## 3.5 DIAGRAMA DE SEQUENCIA

Diagrama de sequência é um diagrama que se preocupa em registrar a ordem em que as mensagens são trocadas entre os processos envolvidos. Ele determina como o processo deve se desenrolar e ser concluído por meio de chamada de métodos disparados por mensagens enviadas entre os objetos.

### 3.5.1 diagrama de sequência manter aluno



powered by Astah

Figura 27: Diagrama de Sequência - Manter Aluno

### 3.6 DIAGRAMA DE CLASSE

Esse diagrama é um dos mais importantes e mais utilizados da UML. Sua principal característica é permitir a visualização das classes do sistema com seus respectivos atributos e métodos, bem como demonstrar como as classes do diagrama se relacionam, complementam e transmitem informações entre si.

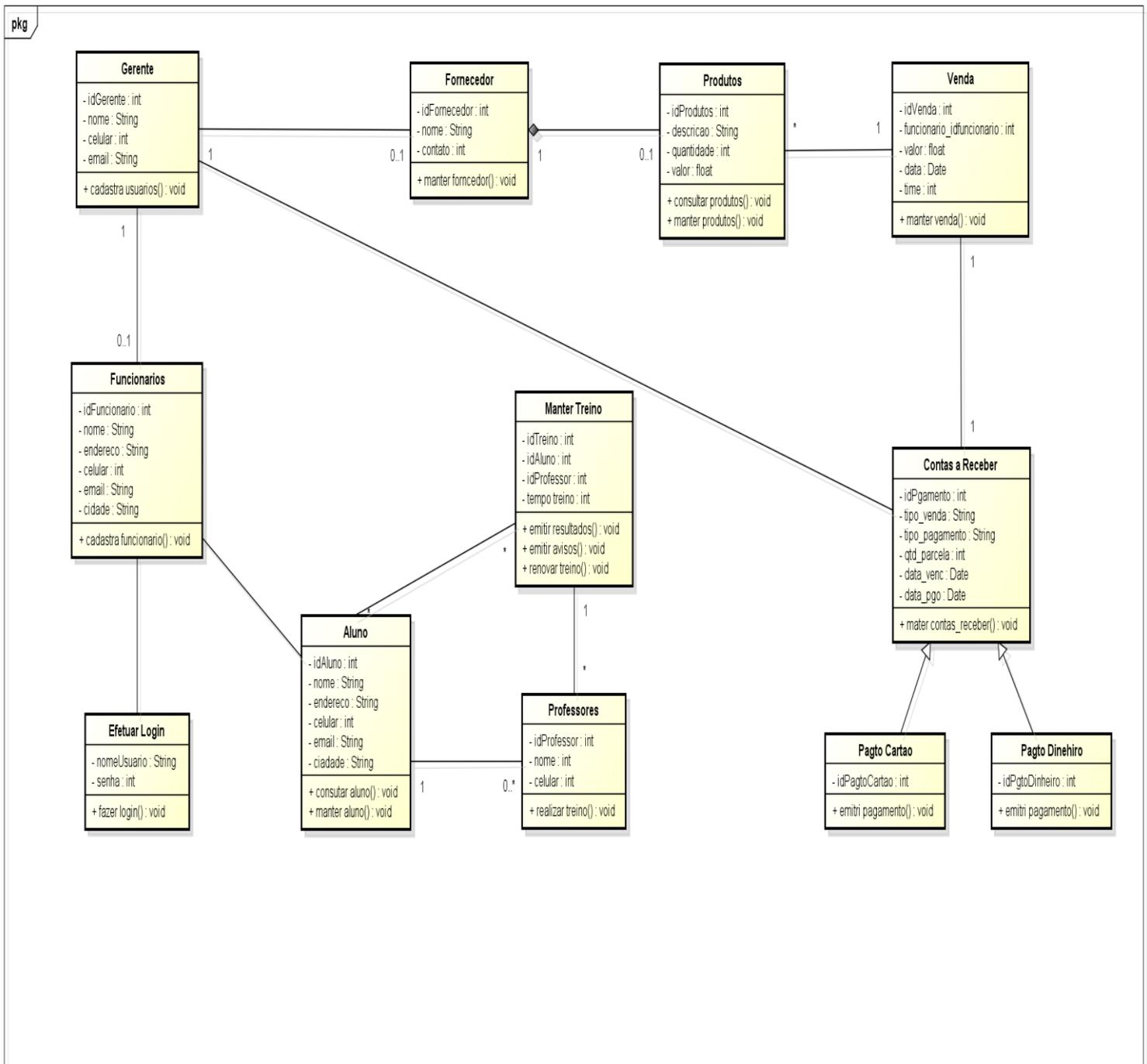


Figura 18: Diagrama de Classe

### 3.7 MODELAGEM ENTIDADE E RELACIONAMENTO

Em engenharia de software, um modelo entidade relacionamento (modelo ER) é um modelo de dados para descrever os dados ou aspectos de informação de um domínio de negócio ou seus requisitos de processo, de uma maneira abstrata que em última análise se presta a ser implementada em um banco de dados, como um banco de dados relacional.

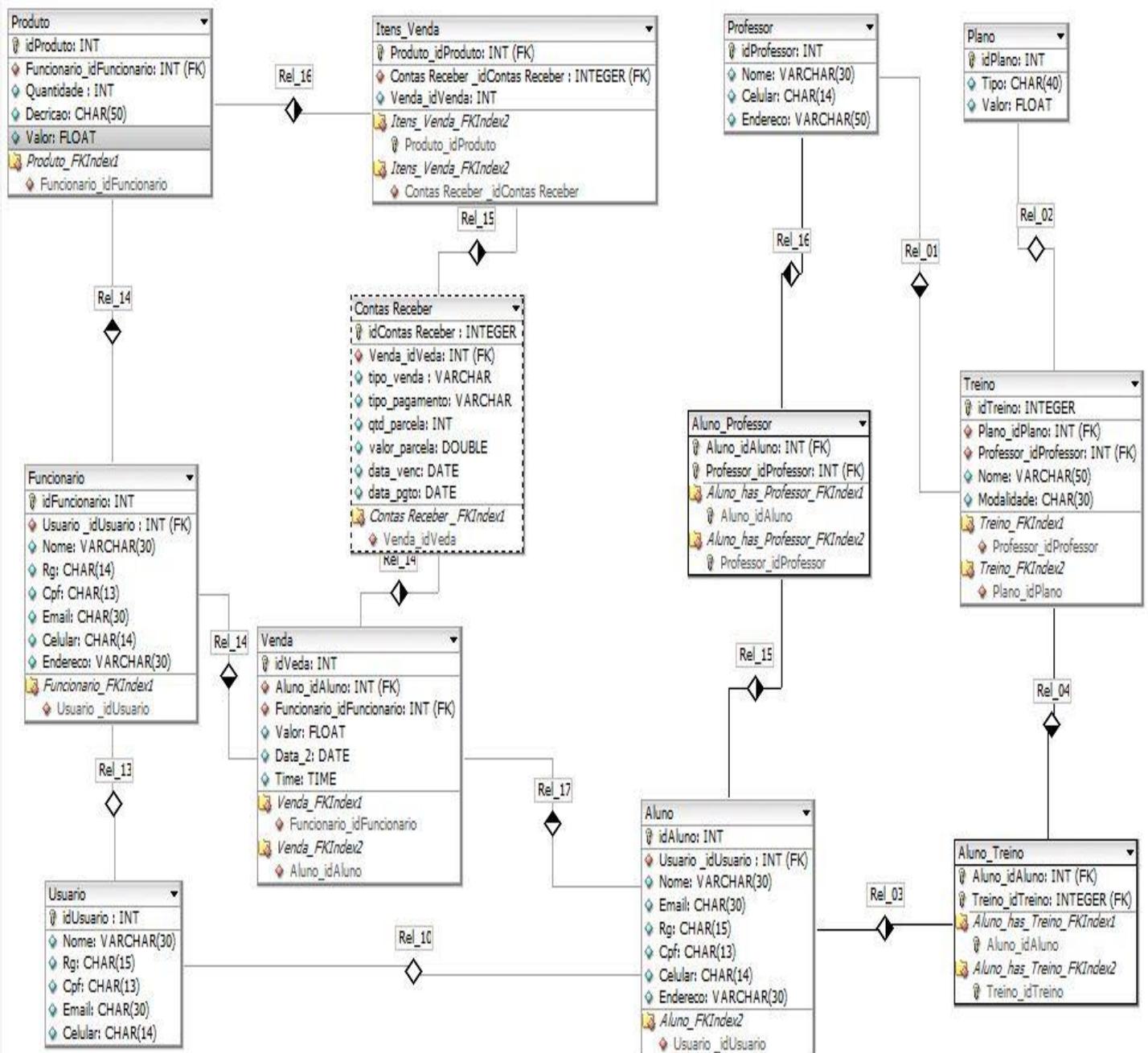
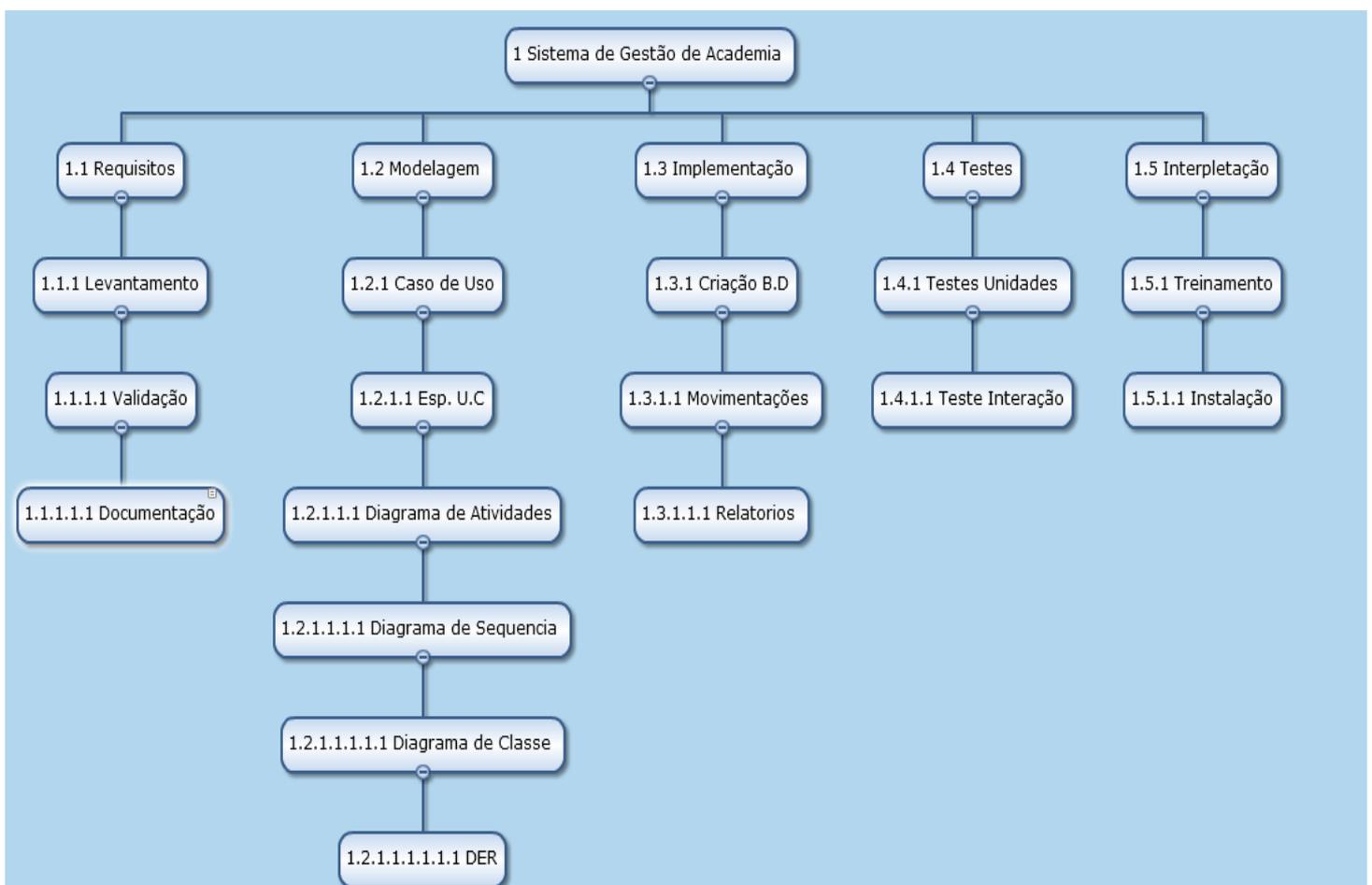


Figura 19: Modelagem Entidade e Relacionamento

## 4. ESTRUTURA ANALÍTICA DO PROJETO

Com a Estrutura Analítica, o desenvolvimento do projeto será mais prático, assim, organizando cada atividade que será realizada, contendo uma sequência e a quantidade de horas que será necessário para determinada atividade ser desenvolvida, em algumas situações aparecerá o caminho crítico, que é a sequência onde tem as atividades mais demoradas para ser desenvolvidas. No caso da Estrutura Analítica desse projeto, o caminho crítico está representado com linhas pontilhadas, veja a imagem a seguir:



**Figura 20:** Estrutura Analítica do Projeto

## 4.1 ORÇAMENTO

### 4.1.1 especificação de custo

Recursos Necessários ao Desenvolvimento

Serão necessários para o desenvolvimento do Projeto, os recursos descritos abaixo:

Equipamento:

01 Microcomputador;

Valor unitário = R\$2.000,00

Depreciação (2 anos) = R\$2.000,00 / 24 = R\$ 83,33 mês

Custo por dia = R\$83,33/ 30 (dias) = R\$2,77 (ao dia)

Custo do computador = R\$2,77 \* 300 = R\$ 831,00

Analista/Programador	Quantidade de Horas		Custo/hora (R\$)	Total (R\$)
<b>Rafael Bilche</b>	Análise	120	25,00	3.000,00
	Desenvolvimento	180	25,00	4.500,00
<b>Total Custo Pessoal</b>				<b>7.500,00</b>

**Custo total do Projeto = R\$ 831,00 + R\$ 7.500,00 = R\$ 8.331,00**

## 5. TELAS DO SISTEMA

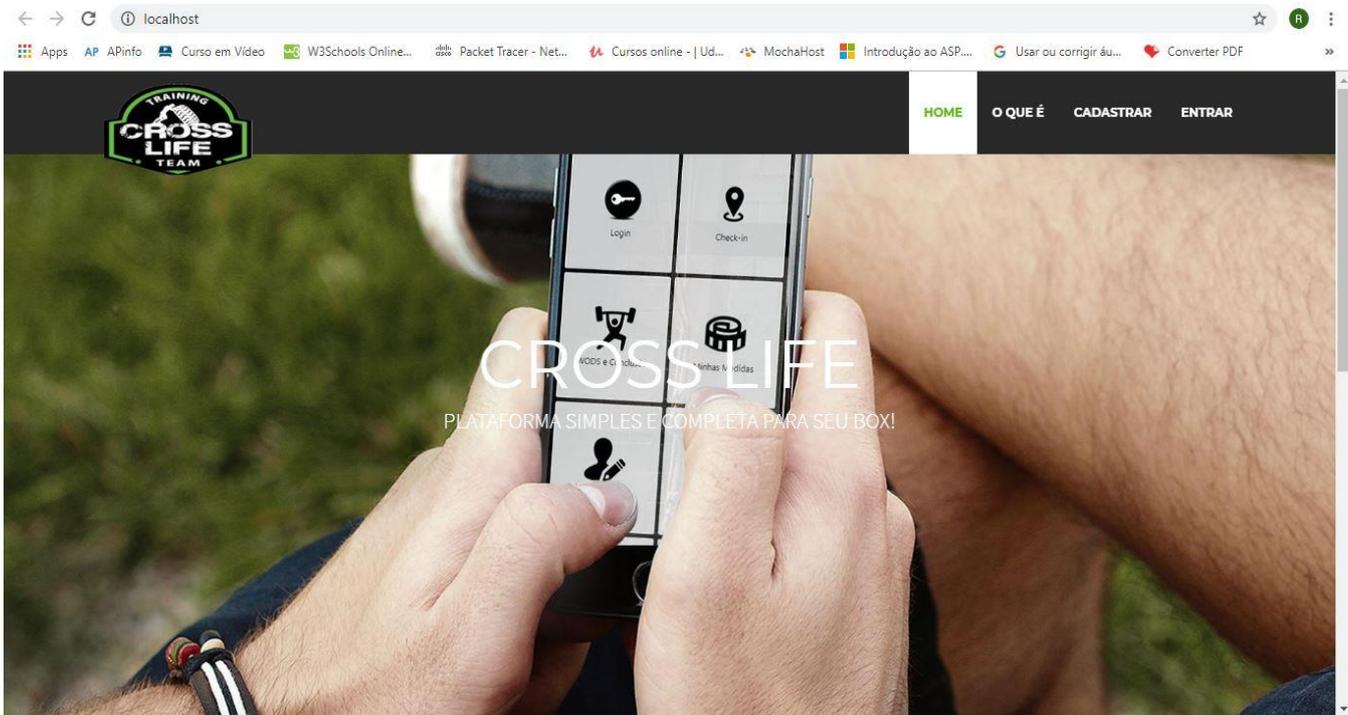


Figura 21: Tela Inicial do Sistema

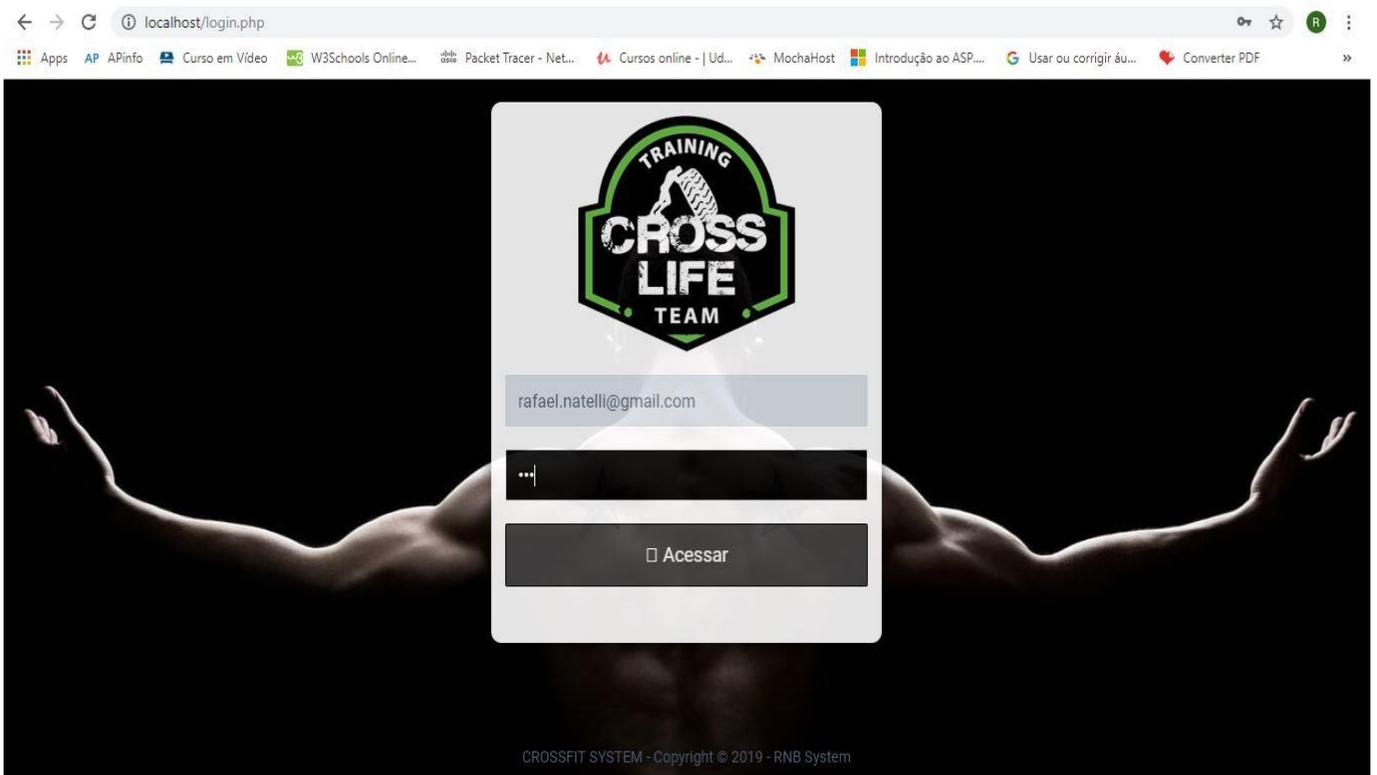


Figura 22: Tela Login

The screenshot shows a web browser at localhost/aluno.php. The application header includes 'Cross Life' and navigation menus for 'Cadastros', 'Produtos', 'Vendas', and 'Relatórios'. The user is logged in as 'Olá (Rafael)'. The main content is a 'Cadastro do Aluno' modal form with the following fields:

- Cadastro** (tab) and **Busca** (search icon)
- Nome do Aluno**: text input
- Data de Nascimento**: date picker
- Sexo**: dropdown menu (Selezione)
- Email**: text input with email icon
- RG**: text input
- CPF**: text input
- Altura**: text input (placeholder: altura)
- Celular**: text input with mobile phone icon
- Profissão**: text input
- Empresa**: text input
- Indicação**: text input
- Observações**: text input
- Dados de Endereço**: expandable section with a red heart icon
- Novo** and **Salvar** buttons at the bottom.

Figura 23: Tela Cadastro de Aluno

The screenshot shows a web browser at localhost/VendaProduto.php. The application header is identical to the previous figure. The main content is a 'Venda de Produto' modal form with the following fields:

- ID Aluno**: text input
- Aluno**: text input
- ID Produto**: text input
- Produto**: text input
- Quantidade Estoque**: text input
- Quantidade**: text input
- Valor**: text input (placeholder: 0,00)
- Desconto**: text input (placeholder: 0,00)
- Valor Total com Desconto**: text input
- Pagamento**: dropdown menu (Cartão)
- Parcelas**: dropdown menu (Selezione)
- Salvar** button at the bottom.

Figura 24: Tela Venda de Produto

## 6. CONCLUSÃO

Esse projeto teve com intuito de otimizar a gestão de academias assegurando que as informações sejam armazenadas com maior segurança. O planejamento inicial do projeto foi essencial para que o seu desenvolvimento pudesse ser realizado e assim atender os requisitos levantados, visando atingir a necessidades do cliente.

Após a análise e planejamento, durante o desenvolvimento do projeto tive a oportunidade de inserir novas ideias ao projeto inicial, com isso o estudo das possibilidades e oportunidades do projeto deve estar sempre em constante evolução a fim de que não haja divergências durante o desenvolvimento do projeto.

Com a realização do projeto foi possível ampliar meus conhecimentos em relação a programação e documentação da análise de software, assim estimulando o crescimento pessoal e profissional através dessa experiencia.

## REFERENCIAS

BOOCH, Grady; Jacobson, J. Rumbaugh, Uml - Guia do usuário, tradução da segunda edição. Ed. Elsevier Brasil. 2016

DALL'OGILIO, Pablo. Php Programando com Orientação a Objetos. Ed. Novatec, 2018.

DUCKETT, Jon. Javascript e Jquery desenvolvimento. Ed. Alta Books, 2016.

GONÇALVES, Edson. Dominando Netbeans, Ed. Ciência Moderna, 2006.

LACRUZ, Adonai José. Plano de Negócios Passo a Passo: Transformando Sonhos em Negócios. Rio de Janeiro: Qualitymark, 2008.

MILANI, Andre. MySQL - Guia do Programador. Ed. Novatec, 2007.

PRESSMAN, Roger S. Engenharia de Software. São Paulo: Makron Books, 1995.

SALOMÃO, Rodolfo; MAGALHÃES, Willian. NETBEANS IDE UTILIZAÇÃO DE RECURSOS PARA DESENVOLVIMENTO DE PROJETO JAVA WEB. 2015. Tese - Universidade Paranaense, Paranavaí, 2015.

SAMY, Maurício. Ajax com jQuery: Requisições AJAX com a simplicidade de jQuery. Ed. Novatec Editora, 2009.

SAMY, Maurício. Fundamentos de HTML5 e CSS3. Ed. Novatec Editora, 2015.

SOMMERVILLE, Ian. Engenharia de Software. São Paulo: Pearson, 2011.

SOUZA, Natan. Bootstrap 4: Conheça a biblioteca front-end mais utilizada no mundo. Ed. Editora Casa do Código, 2018.

VANSCONCELOS, Hugo. Aprenda Tudo sobre HTML5. 11-2015. Disponível em: <https://www.udemy.com/aprenda-tudo-sobre-html5/>. Acesso em 12/03/2019.

VENLIOLES, Fabio Motta. Como Melhorar a Gestão de Academia. Disponível em <<https://www.acadesystem.com.br/gestao-de-academia>>. Acesso em 07/11/2018.

VENLIOLES, Fabio Motta. Manual do Gestor de Academia. Ed. Sprint. 2005

VENLIOLES, Fabio Motta. Manual do Gestor de Academia. Ed. Sprint. 2005