

**CENTRO ESTADUAL DE EDUCAÇÃO TECNOLÓGICA PAULA SOUZA  
ETEC ZONA LESTE**

**Ensino Médio com Habilitação Profissional de Técnico em  
Desenvolvimento de Sistemas - AMS**

**Luma Mari Pereira Kido**

**Maria Eduarda de Carvalho Peixoto**

**Nathalia Lopes da Silva**

**Nayara Nasaria Soares da Silva**

**Yasmin Carvalho Mello**

**STREAMLINE: Busca e contratação de serviços autônomos**

**São Paulo**

**2022**

**Luma Mari Pereira Kido**

**Maria Eduarda de Carvalho Peixoto**

**Nathalia Lopes da Silva**

**Nayara Nasaria Soares da Silva**

**Yasmin Carvalho Mello**

**STREAMLINE: Busca e contratação de serviços autônomos**

Trabalho de Conclusão de Curso apresentado ao Curso do Ensino Médio com Habilitação Profissional de Técnico em Desenvolvimento de Sistemas AMS da Etec Zona Leste, orientado pelo Professor Ediney Ciasi Barreto, como requisito final para obtenção do título de Técnico em Desenvolvimento de Sistemas.

**São Paulo**

**2022**

## DEDICATÓRIA

Dedicamos este trabalho aos nossos professores que nos incentivam nos estudos e nos passam conhecimento para que sejamos ótimos profissionais.

## **AGRADECIMENTOS**

Agradecemos a Deus por ter nos dado saúde e força para superar as dificuldades;

Os nossos orientadores Carlos Alberto Pereira Da Silva, Ediney Ciasi Barreto, Jeferson Roberto De Lima, Maria Emília Gonçalves, Rogério Bezerra Costa, Vilma Cardoso Dos Santos, Wagner De Oliveira Lucca e Wagner França Marques pelo suporte no pouco tempo que lhes coube, pelas suas correções e incentivos;

Aos nossos pais, pelo amor, incentivo e apoio incondicional;

E a todos que direta ou indiretamente contribuíram com o projeto;

## EPÍGRAFE

“Você pode encarar um erro como uma besteira a ser esquecida ou como um resultado que aponta uma nova direção.”

STEVE JOBS

## RESUMO

Este projeto visa utilizar recursos tecnológicos consideravelmente acessíveis a fim de colaborar com a visibilidade e prestação de serviços informais, assim contribuir para a queda da taxa de desemprego no Brasil e proporcionar uma renda extra para a população. A partir de estudos relacionados à Engenharia de Software, *Unified Modeling Language* (UML) e Banco de Dados, o site desenvolvido reunirá informações dos profissionais e as mostrará a potenciais clientes, que poderão especificar, por meio de filtros por eles estabelecidos, o tipo de serviço desejado conforma as suas necessidades. A página fornecerá avaliações de ambas as partes além de estabelecer um orçamento base de acordo com o preço determinado pelo profissional, oferecendo confiabilidade e segurança.

**PALAVRAS-CHAVE:** Site. Contratação. Visibilidade. Emprego. Autônomo. Serviço.

## **ABSTRACT**

*This project aims to use considerably accessible technological resources in order to collaborate with the visibility and provision of informal services, thus contribute to the decline in the unemployment rate in Brazil and provide an extra income for the population. From studies related to Software Engineering, Unified Modeling Language (UML) and Database, the site developed will gather information from the professionals and show it to potential customers, who will be able to specify, through filters established by them, the type of service desired according to their needs. The page will provide feedbacks from both parties besides establishing a base budget according to the price determined by the professional, offering reliability and security.*

**Keywords:** *Site. Hiring. Visibility. Employment. Self-employed. Service.*

## LISTA DE ILUSTRAÇÕES

Figura 1 - Interface criada no Figma .....	13
Figura 2 - Modo editor Figma .....	14
Figura 3 - Comunidade Figma.....	15
Figura 4 - Componentes Figma.....	15
Figura 5 - Barra de tarefas Figma .....	16
Figura 6 - Posicionamento no Figma.....	16
Figura 7 - Estrutura básica HTML .....	17
Figura 8 - HTML da página web .....	19
Figura 9 - Resultado no Navegador .....	20
Figura 10 - Estrutura básica CSS.....	21
Figura 11 - Conexão entre HTML e CSS .....	21
Figura 12 - CSS da página web .....	22
Figura 13 - Resultado no Navegador .....	23
Figura 14 - Responsividade no navegador web .....	24
Figura 15 - Responsividade em tela mobile .....	25
Figura 16 - Conexão com o CSS do Bootstrap .....	26
Figura 17 - Conexão com o Javascript.....	26
Figura 18 - Código Dropdown com tag col .....	27
Figura 19 - Resultado Dropdown com tag col .....	27
Figura 20 - Sintaxe Var .....	28
Figura 21 - Sintaxe Function .....	28
Figura 22 - Alerts no JavaScript .....	29
Figura 23 - Conexão com o Banco de Dados.....	30
Figura 24 - Estrutura básica PHP.....	30
Figura 25 - Representação gráfica do modelo relacional .....	33
Figura 26 - Estrutura básica Dicionário de Dados.....	34
Figura 27 - Estrutura básica Diagrama de Caso de Uso .....	37
Figura 28 - Estrutura básica Diagrama de Atividades .....	38
Figura 29 - Nós de controle .....	39
Figura 30 - Diagrama de Classe “Pessoa” .....	40
Figura 31 - Relação entre as classes Pessoa e Empresa.....	41
Figura 32 - Representação de Associação .....	41



Figura 33 - Estrutura básica Diagrama de Sequência.....	43
Figura 34 - Exemplo de Estado .....	43
Figura 35 - Diagrama de estados para Automóvel, tendo como foco o movimento ..	44
Figura 36 - Diagrama de estados para Automóvel no contexto de uma oficina mecânica .....	44
Figura 37 - (a) Estado inicial (b) Estado final.....	44
Figura 38 - Diagrama de estados de Venda.....	45
Figura 39 - Modelagem de estados para Automóvel, com estado composto .....	46
Figura 40 - Modelagem alternativa para Automóvel, com estado composto .....	46
Figura 41 - Modelagem de Automóvel com estado submáquina.....	47
Figura 42 - Atributos do objeto Gato .....	48
Figura 43 - Métodos do objeto Gato.....	48
Figura 44 - Der: Histórico, Orçamento e Serviços.....	51
Figura 45 - Der: Denúncia .....	52
Figura 46 - Der: Avaliação.....	53
Figura 47 - Tabela cadastrocliente StreamLine.....	54
Figura 48 - Tabela cadastroprofissional StreamLine .....	55
Figura 49 - Tabela servico StreamLine .....	56
Figura 50 - Tabela orcamento StreamLine.....	56
Figura 51 - Tabela historico StreamLine .....	57
Figura 52 - Tabela avaliacoesciente StreamLine.....	57
Figura 53 - Tabela avaliacoeprofissional StreamLine.....	58
Figura 54 - Tabela denuncia_cli StreamLine.....	58
Figura 55 - Tabela denuncia_prof StreamLine .....	59
Figura 56 - Regras de Negócio StreamLine .....	60
Figura 57 - Requisitos Funcionais StreamLine.....	61
Figura 58 - Requisitos Funcionais StreamLine Continuação.....	62
Figura 59 - Requisitos Não Funcionais StreamLine .....	63
Figura 60 - Diagrama de Caso de Uso StreamLine.....	64
Figura 61 - Diagrama de Atividades Inteiro StreamLine.....	65
Figura 62 - Diagrama de Atividade parte 1: Início do Sistema .....	66
Figura 63 - Diagrama de Atividade parte 2: Contratação .....	67
Figura 64 - Diagrama de Atividade parte 3: Finalização do Serviço.....	68
Figura 65 - Diagrama de Estados Principal.....	69

Figura 66 - Diagrama de Estados: Submáquina Login.....	70
Figura 67 - Diagrama de Estados: Submáquina Cadastro .....	71
Figura 68 - Diagrama de Estados: Submáquina Contratação Cliente .....	72
Figura 69 - Diagrama de Estados: Submáquina Serviço Autônomo .....	73
Figura 70 - Diagrama de Estados: Submáquina Verificação Denúncia.....	74
Figura 71 - Logo StreamLine: Ely Neutra .....	75
Figura 72 - Logo StreamLine: Ely Surpresa .....	76
Figura 73 - Logo StreamLine: Ely Aborrecida.....	77
Figura 74 - Página Inicial Completa StreamLine .....	78
Figura 75 - Página Inicial: Intuito .....	79
Figura 76 - Página Inicial: Funcionamento .....	80
Figura 77 - Página Inicial: Cadastro e Login.....	80
Figura 78 - Página Inicial: Categoria Diarista .....	81
Figura 79 - Página Inicial: Categoria Estética.....	81
Figura 80 - Página Inicial: Categoria Aulas .....	82
Figura 81 - Página Inicial: Categoria Fotografia .....	82
Figura 82 - Página Inicial: Categoria Babá.....	83
Figura 83 - Cadastro como Cliente: Informações Básicas .....	84
Figura 84 - Cadastro como Cliente: Endereço .....	85
Figura 85 - Cadastro como Autônomo: Informações Básicas .....	86
Figura 86 - Cadastro como Autônomo: Endereço .....	87
Figura 87 - Cadastro como Autônomo: Detalhes .....	88
Figura 88 - Login Cliente .....	89
Figura 89 - Página Principal Cliente: Profissionais Pendentes.....	89
Figura 90 - Página Principal Cliente: Principais Autônomos .....	90
Figura 91 - Página Principal Cliente: Funcionamento e Instruções.....	91
Figura 92 - Login Autônomo.....	91
Figura 93 - Página Principal Autônomo: Clientes Pendentes.....	92
Figura 94 - Página Principal Autônomo: Perfis de Clientes e Instruções .....	93
Figura 95 - Perfil Cliente: Dados Pessoais.....	94
Figura 96 - Perfil Cliente: Dados Públicos.....	94
Figura 97 - Perfil Autônomo: Dados Pessoais.....	95
Figura 98 - Perfil Autônomo: Dados Público .....	95
Figura 99 - Trabalhos Pendentes: Visão como Autônomo .....	96

Figura 100 - Trabalhos Realizados: Visão como Autônomo .....	97
Figura 101 - Trabalhos Realizados: Visão como Cliente.....	97
Figura 102 - Trabalhos Pendentes: Visão como Cliente .....	98
Figura 103 - Dados do Autônomo Parte 1: Visão como Cliente .....	99
Figura 104 - Dados do Autônomo Parte 2: Visão como Cliente .....	99
Figura 105 - Dados da Avaliação do Autônomo: Visão como Cliente .....	100
Figura 106 - Avaliar Autônomo: Visão como Cliente .....	101
Figura 107 - Denunciar Autônomo: Visão como Cliente.....	101
Figura 108 - Avaliar Cliente: Visão como Autônomo .....	102
Figura 109 - Denunciar Cliente: Visão como Autônomo.....	103

## LISTA DE TABELAS

Tabela 1 -Tabela de Caso de Uso.....	38
Tabela 2 - Símbolos de associação .....	42
Tabela 3 - Conceitos essenciais do diagrama de sequência .....	42
Tabela 4 - Significado dos conceitos do paradigma de OO .....	48
Tabela 5 - Os pilares do paradigma de OO.....	49

## LISTA DE ABREVIATURAS E SIGLAS

Articulado Médio Superior (AMS)

Banco de Dados (BD)

*Cascading Style Sheet* (CSS)

*Database Administrator* (DBA)

Dicionário de Dados (DD)

*HyperText Markup Language* (HTML)

Instituto Brasileiro de Geografia e Estatística (IBGE)

*Internal Revenue Service* (IRS)

Orientado a Objetos (OO)

*Personal Home Page* (PHP)

*Primary Key* (PK)

Projeto de Banco de Dados (PBD)

Sistema de Banco de Dados (SBD)

Sistema de Gerenciamento de Bancos de Dados (SGBD)

Trabalho de Conclusão de Curso (TCC)

*Unified Modeling Language* (UML)

# SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO .....</b>	<b>12</b>
<b>2</b>	<b>REFERENCIAL TEÓRICO .....</b>	<b>13</b>
2.1	Figma.....	13
2.1.1	Compartilhamento .....	14
2.1.2	Comunidade .....	14
2.1.3	Funcionamento .....	15
2.2	HTML .....	17
2.3	CSS.....	20
2.4	Bootstrap .....	23
2.5	JavaScript.....	27
2.6	PHP.....	29
2.7	Banco de Dados .....	31
2.7.1	Abordagem relacional.....	32
2.7.2	Normalização .....	33
2.7.3	Dicionário de dados .....	33
2.8	UML .....	35
2.8.1	Levantamento de Análise e Requisitos .....	36
2.8.2	Diagrama de Caso de Uso .....	37
2.8.3	Diagrama de Atividades .....	38
2.8.4	Diagrama de Classes.....	39
2.8.5	Diagrama de Sequência .....	42
2.8.6	Diagrama de Estados .....	43
2.9	Paradigma de Orientação a Objetos.....	47
<b>3</b>	<b>DESENVOLVIMENTO.....</b>	<b>50</b>
3.1	Banco de Dados .....	50

3.1.1	Abordagem Relacional.....	50
3.1.2	Dicionário de Dados .....	53
3.2	UML .....	59
3.2.1	Levantamento de Análise e Requisitos .....	59
3.2.2	Diagrama de Caso de Uso .....	64
3.2.3	Diagrama de Atividades .....	65
3.2.4	Diagrama de Estados .....	68
3.3	Logo StreamLine.....	74
3.4	Telas do Site StreamLine.....	77
3.4.1	Página Inicial.....	78
3.4.2	Cadastros .....	83
3.4.3	Login.....	88
3.4.4	Perfil.....	93
3.4.5	Trabalhos .....	96
4	CONCLUSÃO.....	104
	REFERÊNCIAS.....	105

## 1 INTRODUÇÃO

Em detrimento da demanda de trabalho, o mundo atual descarta profissionais por exigir um currículo abrangente e perfil social atualizado, o que ocasiona a falta de emprego. Segundo o portal de notícias brasileiras G1, dados do Instituto Brasileiro de Geografia e Estatística (IBGE) mostram que a taxa de desemprego atingiu cerca de 12 milhões de brasileiros no segundo trimestre de 2022 (maio, junho e julho). Atualmente, um indivíduo precisa se atualizar de acordo com as redes sociais e os avanços tecnológicos para que se torne possível a conexão entre contratante e profissional. Mas como se conectar quando não sabe por onde começar?

Ao adquirir informações por meio de pesquisar relacionadas, artigos e livros chegamos à conclusão sobre o que cada autônomo precisa para divulgar o seu perfil de forma acessível se adequando ao que o cliente busca atualmente usando ferramentas tecnológicas que o auxiliam efetivamente.

A proposta de trabalho é um site que atenda os critérios básicos de propaganda, conexão interpessoal e segurança, buscando diminuir o desemprego brasileiro e possibilitando que mais pessoas possam obter uma fonte de renda segura através de serviços autônomos.

Partindo do auxílio de pesquisas bibliográficas e através da linguagem de programação *Personal Home Page* (PHP), *HyperText Markup Language* (HTML), Javascript, *Cascading Style sheet* (CSS), Bootstrap, Figma, estudos de *Unified Modeling Language* (UML) e Banco de Dados (BD), foi realizado um site que cadastra informações de trabalhadores com vista às suas competências e perfis profissionais para mostrar ao cliente, através de filtros, quem melhor se enquadra às suas necessidades, ao mesmo tempo que garante a segurança do contratante e do trabalhador ao passar a confiabilidade através de seus dados pessoais.



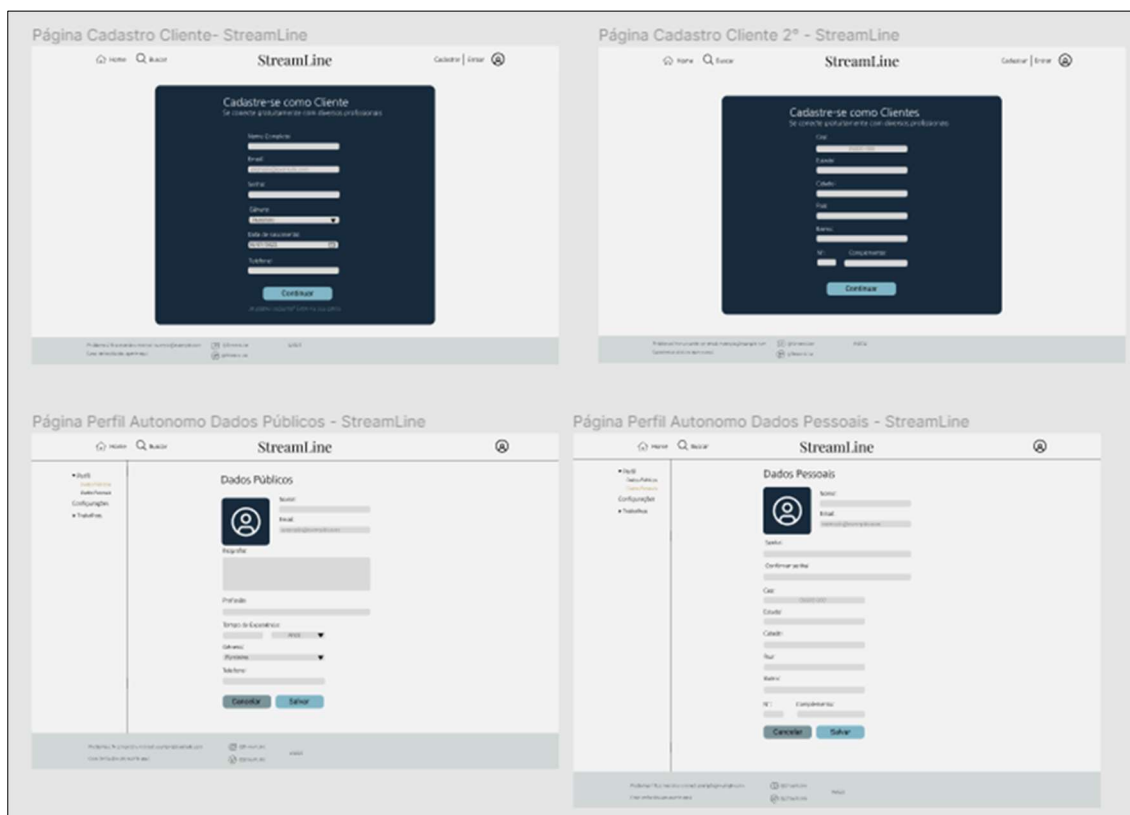
## 2 REFERENCIAL TEÓRICO

Nessa seção, será retratada a fundamentação teórica sobre as tecnologias que serão utilizadas para o desenvolvimento do projeto de TCC StreamLine, um site de busca e contratação de serviços autônomos.

### 2.1 Figma

De acordo com Nascimento (2020) o Figma é um software de criação de protótipos gratuito que é utilizável em todos os sistemas operacionais que tenham navegadores da web. Através do armazenamento na nuvem, ele torna possível o compartilhamento dos arquivos entre os usuários mantendo a opção de autorização para modificar ou visualizar o projeto para que, desta forma, auxilie no desenvolvimento e no julgamento entre diversos usuários de forma síncrona ou assíncrona criando projetos por meio de protótipos de designs fieis aos seus requisitos que é uma parte fundamental da aplicação, auxiliando nesse quesito da interface como mostrado na Figura 1.

Figura 1 - Interface criada no Figma



Fonte: Autoria própria, 2022.

### 2.1.1 Compartilhamento

Conforme dito por Oliveira (2022) existe na plataforma o modo de editor, que consiste em poder modificar, adicionar e apagar, tendo todo o acesso do arquivo, e o de visualizador que se resume em analisar as informações da interface e usufruir dos modelos interativos como demonstrado na Figura 2, tornando possível e prática a interação entre o designer e desenvolvedor em um mesmo arquivo, contribuindo na fluência de trabalho e nas atribuições das linhas de produção.

**Figura 2 - Modo editor Figma**

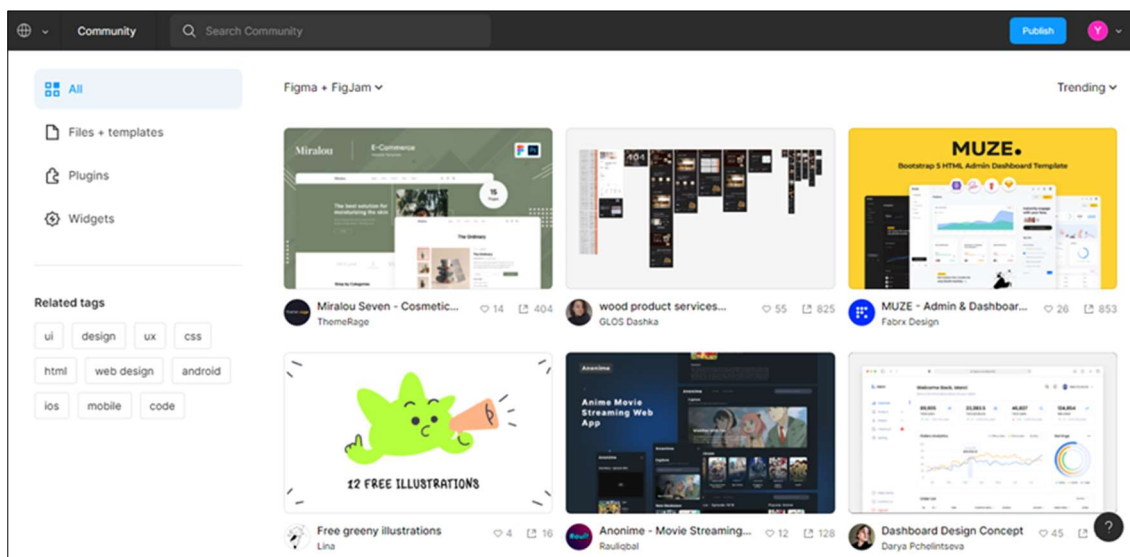
A imagem mostra a interface de usuário da plataforma StreamLine, especificamente o modo editor Figma. No topo, há uma barra de navegação com ícones para Home e Buscar, o nome da plataforma 'StreamLine' e um ícone de perfil. À esquerda, há um menu lateral com opções: Perfil (com subitens Dados Públicos, Dados Pessoais, Configurações e Trabalhos). O conteúdo principal exibe o formulário 'Dados Públicos' com campos para Nome, Email (exemplo@exemplo.com), Biografia, Profissão, Tempo de Experiência (com uma barra deslizante e um menu suspenso 'Anos'), Gênero (com um menu suspenso 'Feminino') e Telefone.

Fonte: Autoria própria, 2022.

### 2.1.2 Comunidade

A plataforma também possui uma comunidade, Figura 3, que segundo Oliveira (2022) surgiu através da possibilidade de um ambiente de compartilhamento de publicações comunitárias entre usuários, que desenvolvem diversos designs de interfaces como guias de estilo, plugins, manuais de uso das ferramentas entre outros.

**Figura 3 - Comunidade Figma**

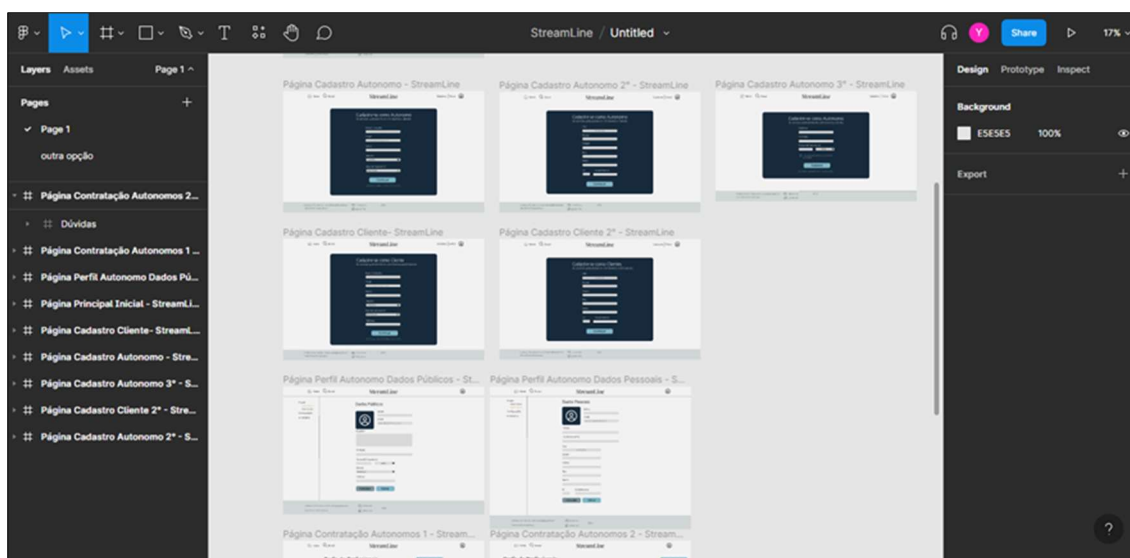


Fonte: Autoria própria, 2022.

### 2.1.3 Funcionamento

O lado esquerdo da Figura 4 dispõe da organização dos componentes, onde ficam colocados através de tópicos e subtópicos, para facilitar na visualização do layout.

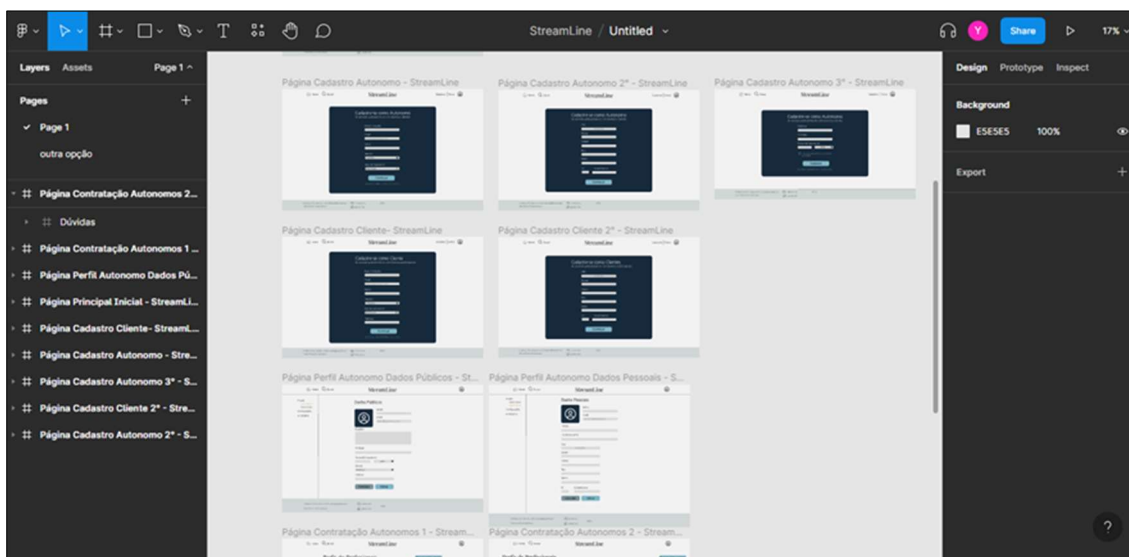
**Figura 4 - Componentes Figma**



Fonte: Autoria própria, 2022.

No canto superior esquerdo da Figura 5, temos a barra de tarefas, que possui as opções de criarmos novos arquivos, frames que facilitam na organização do layout, formas geométricas, canetas, textos, componentes, opção de arrastar a tela e de comentários.

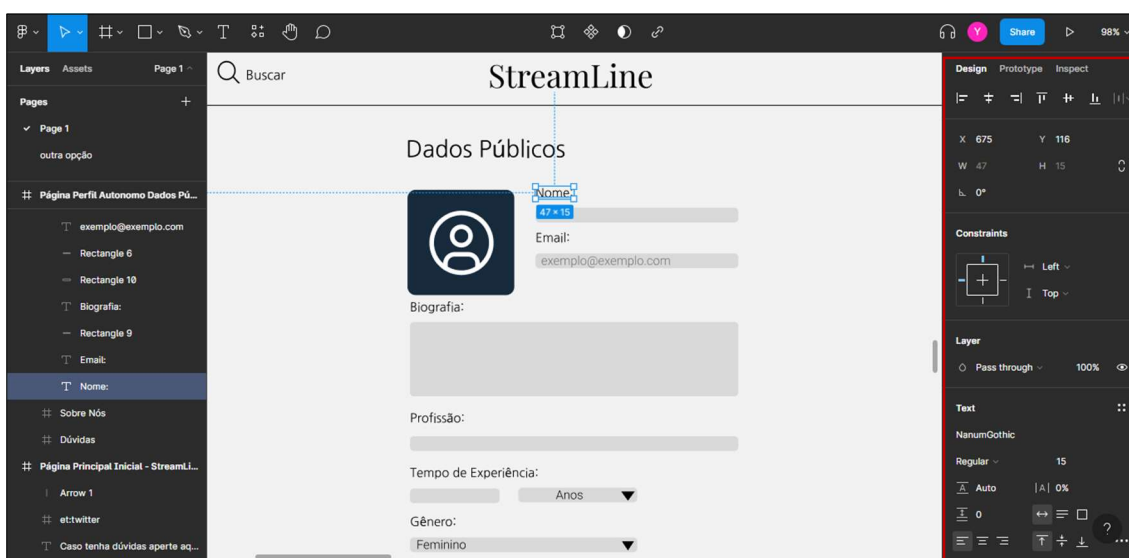
Figura 5 - Barra de tarefas Figma



Fonte: Autoria própria, 2022.

O lado direito da Figura 6 contém alternativas, arrastar os componentes para suas posições com o mouse, através dos eixos X e Y, ou das possibilidades de centro ou laterais para colocá-los em locais específicos, tendo como modificar suas cores e efeitos. Possui também formatações de textos como tamanho, fonte, espaçamentos entre outros.

Figura 6 - Posicionamento no Figma



Fonte: Autoria própria, 2022.

## 2.2 HTML

Em congruência com Silva (2015), o *HyperText Markup Language* (HTML) traduzido para o português significa Linguagem para Marcação de Hipertexto. O hipertexto é estabelecido como textos que tem como finalidade estarem interligados a documentos para a internet, sendo uma de suas funções primordiais, um encadeamento com outros documentos web através de links.

O HTML representa a primeira camada para o desenvolvimento de um site, chamada de camada de informação. “O HTML marca a informação dando-lhe significado” (EIS; FERREIRA, 2012, p. 11).

Conforme Eis e Ferreira (2012), no decorrer dos anos houveram diversas versões do HTML, todavia, sua estrutura básica demonstrou mudanças pouco relevantes, como mostra a Figura 7:

**Figura 7 - Estrutura básica HTML**

```
<!DOCTYPE html>
<html lang="pt-br">
  <head>
    <meta charset="UTF-8"/>
    <title>Título da página</title>
    <link rel="stylesheet" type="text/css" href="css/interface.css" />
  </head>
  <body>
    <!-- Corpo da Página web -->
  </body>
</html>
```

Fonte: Autoria própria, 2022.

Torres (2018) afirma que o Doctype tem como finalidade designar a especificação de código que terá que ser empregada no navegador de internet e possível outros recursos definindo como o meio de acesso que irá se submeter ao código instruído. Ele funciona como uma orientação de como executar o documento no navegador contribuindo com uma exibição do site mais favorável.

A tag <html> é considerada um dos elementos de maior importância do código, pois em seu interior estará todo o documento que será inicializado.

Junto a ele está o <lang = "pt-br"> que de acordo com Silva (2015, p. 43) “[...] é o que denominamos par atributo/valor sendo lang o atributo e pt-br o seu valor.” Indica que

a linguagem utilizada na escrita dos códigos será em português brasileiro contribuindo em uma melhor eficiência dos leitores de tela e outras tecnologias assistivas.

A tag `<head>` determina um cabeçalho para o documento possibilitando o reconhecimento de outros elementos chamados de metadados. Nele apresenta informações essenciais sobre como será a página tendo que indicar pelo menos duas tags: `<meta charset>` e `<title>`.

A tag `<meta charset>` define os caracteres padrão a serem seguidos na composição do documento dispondo de uma tabela nomeada como Unicode que possui milhares de caracteres e suas singularidades dos diversos idiomas do mundo, diz Eis e Ferreira (2012).

Em concordância com Silva (2015), a tag `<title>` tem como função indicar resumidamente o conteúdo retratado na página. Sendo de suma importância para pesquisas em ferramentas de busca, pois as informações instruídas pelo usuário serão filtradas pelo título dos documentos.

A tag `<body>` como o nome já fala, é caracterizada por constituir todo o corpo do documento, segundo Silva (2015) todos os elementos nele introduzidos corretamente serão refletidos no navegador conforme descrito por meio dos códigos.

A Figura 8 traz um formulário básico feito apenas de HTML no editor de código-fonte Notepad++ para uma melhor compreensão:

Figura 8 - HTML da página web

```

<!DOCTYPE html>
<html lang="pt-br">
  <head>
    <meta charset="UTF-8" />
    <title>Fomulário Básico</title>
    <link rel="stylesheet" type="text/css" href="css/interface.css" />
  </head>
  <body>
    <form action="form.php" method="post">
      <div id="elementos">
        <h1>Cadastro</h1>
        <p>Informe seus dados:</p>
        <div>
          <label for="nome">Nome:</label>
          <input type="text" name="nome" size="40" maxlength="100">
        </div>
        <div>
          <label for="sexo">Sexo:</label>
          <select name="sexo">
            <option>Escolha o sexo
            <option>Feminino
            <option>Masculino
            <option>Outros
          </select>
        </div>
        <div>
          <label for="endereco">Endereço:</label>
          <input type="text" name="endereco" size="40">
          <select name="uf">
            <option>RO<option>AC<option>AM<option>RR<option>PA<option>AP<option>TO
            <option>MA<option>PI<option>CE<option>RN<option>PB<option>PE<option>AL
            <option>SE<option>BA<option>MG<option>ES<option>RJ<option>SP<option>PR
            <option>SC<option>RS<option>MS<option>MT<option>GO<option>DF
          </select>
        </div>
        <div>
          <label for="escolaridade">Escolaridade:</label>
          <input type="radio" name="escolaridade" value="fundamental">Fundamental
          <input type="radio" name="escolaridade" value="medio">Médio
          <input type="radio" name="escolaridade" value="graduacao" checked>Graduação
        </div>
        <div id="button">
          <input type="submit" name="enviar" value="Enviar">
          <input type="reset" name="limpa" value="Limpar">
        </div>
      </div>
    </form>
  </body>
</html>

```

Fonte: Autoria própria, 2022.

Logo depois, como demonstrado na Figura 9, o resultado visual do código sendo executado em um navegador:

Figura 9 - Resultado no Navegador

**Cadastro**

Informe seus dados:

Nome:

Sexo:

Endereço:

Escolaridade:  Fundamental  Médio  Graduação

Fonte: Autoria própria, 2022.

## 2.3 CSS

Em conformidade com Silva (2015) *Cascading Style Sheet (CSS)* em português equivale a folha de estilo em cascata com o objetivo de servir como uma ferramenta em documentos web para a implementação de variedades de fontes, espaçamentos, cores entre muitos outros nomeados como estilos para a página.

De acordo com Eis e Ferreira (2012) essa linguagem de marcação corresponde ao controle visual das informações que estão sendo passadas para o internauta através do HTML também conhecida com a camada da formatação.

Eis e Ferreira (2012) agregam dizendo que uma de suas principais finalidades está em fazer com que as informações da página web possam ser acessadas facilmente por qualquer meio de acesso.

Segundo Silva (2015) para ser desenvolvido as linhas de código a estilização do documento web, é necessário levar em conta uma regra essencial com dois componentes o seletor e a declaração, que seria a junção entre a propriedade e o valor, como explicado:

- Seletor: seria o foco da declaração, tudo posto entre as chaves ({}), será atribuído a ele.
- Declaração: ela estabelece quais serão os critérios para a estilização. Contendo a propriedade e o valor.



- Propriedade: seria qual aspecto do seletor que será implementado o atributo de estilo.
- Valor: é a determinação de mensura ou competências da propriedade.

Demonstração de um componente na Figura 10:

**Figura 10 - Estrutura básica CSS**

```
seletor{  
    propriedade: valor;  
    /*declaração*/  
}
```

Fonte: Autoria própria, 2022.

Segundo Mazza (2014), para que seja feito um CSS próprio para um determinado documento web, se faz necessário criar um arquivo de CSS, como na Figura 11, correspondente e adicioná-lo ao HTML em específico por meio da tag <link> inserido na tag <head>.

**Figura 11 - Conexão entre HTML e CSS**

```
<!DOCTYPE html>  
<html lang="pt-br">  
  <head>  
    <meta charset="UTF-8" />  
    <title>Fomulário Básico</title>  
    <link rel="stylesheet" type="text/css" href="css/interface.css" />  
  </head>
```

Fonte: Autoria própria, 2022.

Na Figura 12 traz um código de CSS básico para o formulário em HTML anterior:

Figura 12 - CSS da página web

```
@import url('https://fonts.googleapis.com/css2?family=Nunito:wght@200&display=swap');

body {
  background-color: #008080;
  font-family: 'Nunito', sans-serif;
  font-size: 12pt;
  margin-top: 30px;
}

div#button input{
  background-color: #008080;
  padding: 8px;
  margin-top: 20px;
  transition: background-color 1s;
  border-radius: 0.5em;
  border: none;
  float: right;
  margin-right: 30px;
}

div#button input:hover {
  background-color: #008080;
}

div#button input{
  color: #FFFFFF;
  text-decoration: none;
}

form{
  background-color: #FFFFFF;
  /*Apenas para centralizar o form na página*/
  margin: 0 auto;
  width: fit-content;
  /*Para ver as bordas do formulário*/
  padding: 3em;
  border: 0.5px solid #CCC;
  border-radius: 0.5em;
}

div#elementos{
  line-height: 50px;
  margin-bottom: 40px;
}
```

Fonte: Autoria própria, 2022.

Feita a estilização do site com CSS, é possível notar as mudanças atribuídas na página web quando executado no navegador logo depois, como na Figura 13:

**Figura 13 - Resultado no Navegador**

Cadastro

Informe seus dados:

Nome:

Sexo:

Endereço:

Escolaridade:  Fundamental  Médio  Graduação

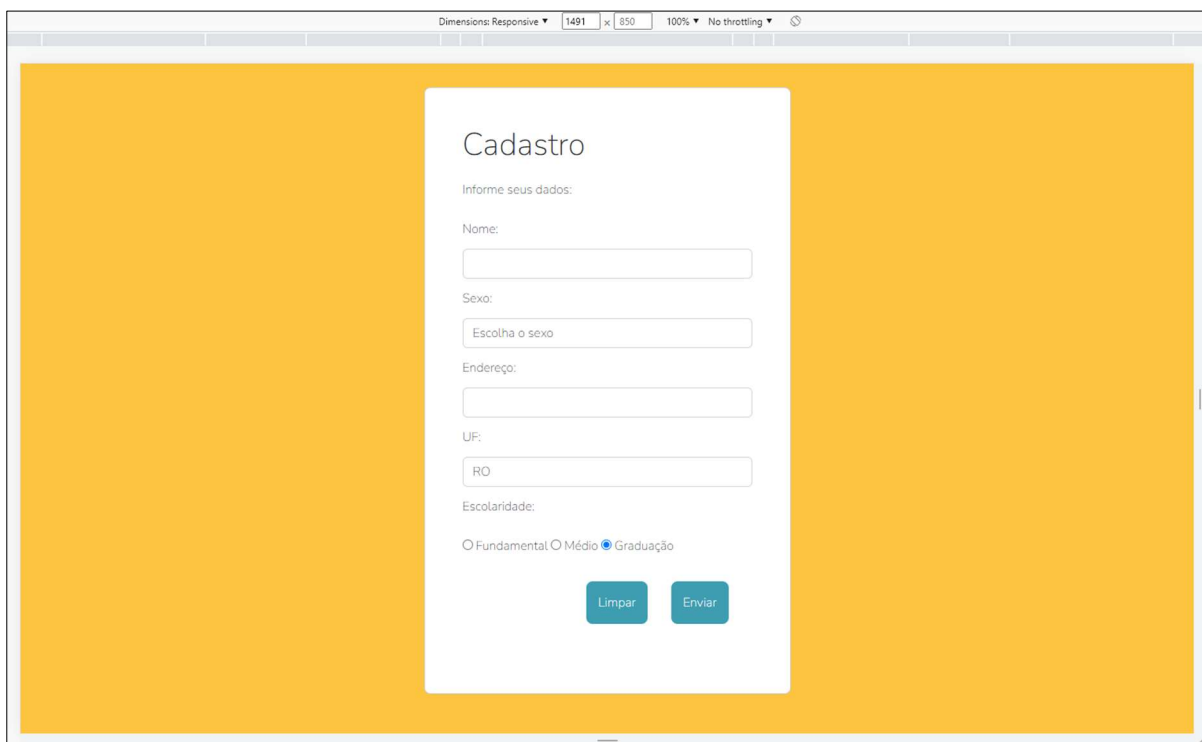
Fonte: Autoria própria, 2022.

## 2.4 Bootstrap

Tendo compatibilidade com HTML5 e CSS3 o Bootstrap é um *framework* desenvolvido por Mark Otto e Jacob Thornton, que possui componentes padrões customizáveis.

Conforme dito por Albino (2015) a estrutura do Bootstrap é inteiramente feita em HTML, facilitando na criação de sites complexos sem necessitar de grandes conhecimentos em Javascript e CSS.

O Bootstrap, de acordo com Silva (2014), foi criado para auxiliar desenvolvedores na criação de web sites com responsividade, tendo foco em ser compatível com telas menores como aparelhos móveis, até telas maiores como computadores, criando dessa maneira uma estrutura padrão para os sites, tornando o processo de criação ágil e rápido. Como exibido na Figura 14 representando a responsividade no desktop:

**Figura 14 - Responsividade no navegador web**

Dimensions: Responsive | 1491 x 850 | 100% | No throttling

### Cadastro

Informe seus dados:

Nome:

Sexo:

Endereço:

UF:

Escolaridade:  
 Fundamental  Médio  Graduação

Fonte: Autoria própria, 2022.

Podendo exemplificar a responsividade em telas mobile também, como na Figura 15:

**Figura 15 - Responsividade em tela mobile**

Dimensions: Samsung Galaxy S20 Ultra 412 x 915 92% No throttling

### Cadastro

Informe seus dados:

Nome:

Sexo:

Endereço:

UF:

Escolaridade:  
 Fundamental  Médio  Graduação

Fonte: Autoria própria, 2022.

Para ser introduzido ao código deve ser declarado a tag `<meta name="viewport">` para que ele seja responsivo da maneira correta. O CSS deve ser conectado através do *stylesheet*, por meio da pasta onde o arquivo baixado se encontra como mostra a Figura 16, ou através do link do site:

**Figura 16 - Conexão com o CSS do Bootstrap**

```
<!DOCTYPE html>
<html lang="pt-br">
  <head>
    <meta charset="UTF-8" />
    <title>Fomulário Básico</title>
    <meta name="viewport" content="width=device-width"/>
    <link rel="stylesheet" type="text/css" href="bootstrap/css/bootstrap.min.css">
    <link rel="stylesheet" type="text/css" href="css/interface.css" />
  </head>
```

Fonte: Autoria própria, 2022

Para que muitos dos componentes funcionem é necessário utilizar o JavaScript que emprega o jQuery e o Popper.js, ambos devem ser conectados como mostrado na Figura 17, ou através da pasta onde o arquivo baixado está situado:

**Figura 17 - Conexão com o Javascript**

```
<body>
  <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js"
  integrity="sha384-q8i/X+965Dz00rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo"
  crossorigin="anonymous"></script>
  <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.3/umd/popper.min.js"
  integrity="sha384-ZMP7rVo3mIykV+2+9J3UJ46jBk0WLaUAdn689aCwoqbBJiSnjAK/l8WvCWPIpM49"
  crossorigin="anonymous"></script>
  <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/js/bootstrap.min.js"
  integrity="sha384-ChfqqxuZUCnJSK3+MXmPNIyE6ZbWh2IMqE241rYiqJxyMiZ6OW/JmZQ5stwEULTy"
  crossorigin="anonymous"></script>
```

Fonte: Autoria própria, 2022

Segundo Albino (2015) O Bootstrap funciona por meio de um *Grid System* (sistema de grade), que é dividida em doze colunas, podendo ser variada em versões menores, sendo modificada por entre *containers* (classes) que definirão os espaços que o conteúdo ocupara na tela, através de *rows* (linhas) e *cols* (colunas).

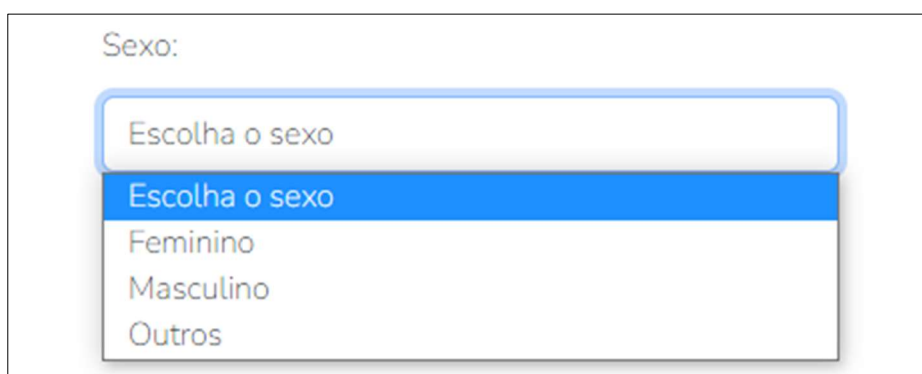
Para a melhor responsividade é utilizado a *tag* <col>, que se modifica a partir da tela que é desejada, sendo *col-xs-* para telas muito pequenas, *col-sm-* para pequenas, *col-md-* para médias, *col-lg-* para grandes, conforme explicado por Albino (2015), como mostrado na Figura 18:

**Figura 18 - Código Dropdown com tag col**

```
<div>
  <label for="sexo">Sexo:</label>
  <select name="sexo" class="form-control col-lg-6" id="sexo">
    <option>Escolha o sexo
    <option>Feminino
    <option>Masculino
    <option>Outros
  </select>
</div>
```

Fonte: Autoria própria, 2022

Resultando na Figura 19:

**Figura 19 - Resultado Dropdown com tag col**

Fonte: Autoria própria, 2022

## 2.5 JavaScript

De acordo com Silva (2010), o JavaScript é uma linguagem que opera ao lado do cliente, ou seja, para que seja funcional deve ocorrer interação no navegador do usuário. Isso é possível por conta de um interpretador JavaScript hospedado no navegador.

Dimes (2015) diz que a linguagem faz com que os desenvolvedores tenham mais facilidade em interagirem com as páginas web utilizando o próprio navegador. O JavaScript se baseia na linguagem de programação Java, mas é uma linguagem própria, mesmo que suas metodologias e sintaxes de programação sejam idênticas, ele não pode ser considerado uma versão simples do Java.

Conforme Prescott (2016), usando essa linguagem o programador ajuda os usuários a interagirem bem com as páginas Web. Com ele é possível controlar o navegador, comunicar com o servidor de forma assíncrona, ou seja, que não ocorre ao mesmo

tempo, alterar de forma criativa o conteúdo de uma página e criar jogos e aplicações tanto móveis quanto para desktops.

Como diz Silva (2010) a declaração é uma linha ou bloco de comandos que consiste em um script. Com isso um programa é definido por uma sequência de declarações. A tag <var> tem como objetivo declarar evidentemente uma ou mais variáveis. A declaração contém ao menos o nome de uma variável, sendo capaz de declarar valores e múltiplas variáveis em uma lista, isoladas por vírgula. Exemplificando com a sintaxe geral para declaração de variáveis na Figura 20:

**Figura 20 - Sintaxe Var**

```
function enviado(){
    var nome = document.forms["cadastro"]["nome"].value;
    var email = document.forms["cadastro"]["email"].value;
    var senha = document.forms["cadastro"]["senha"].value;
```

Fonte: Autoria própria, 2022

A tag <function> tem como objetivo declarar uma função Javascript que pode aderir zero ou mais argumentos. De forma geral, uma função remete-se a definir um bloco de códigos a ser executado como mostrado na Figura 21:

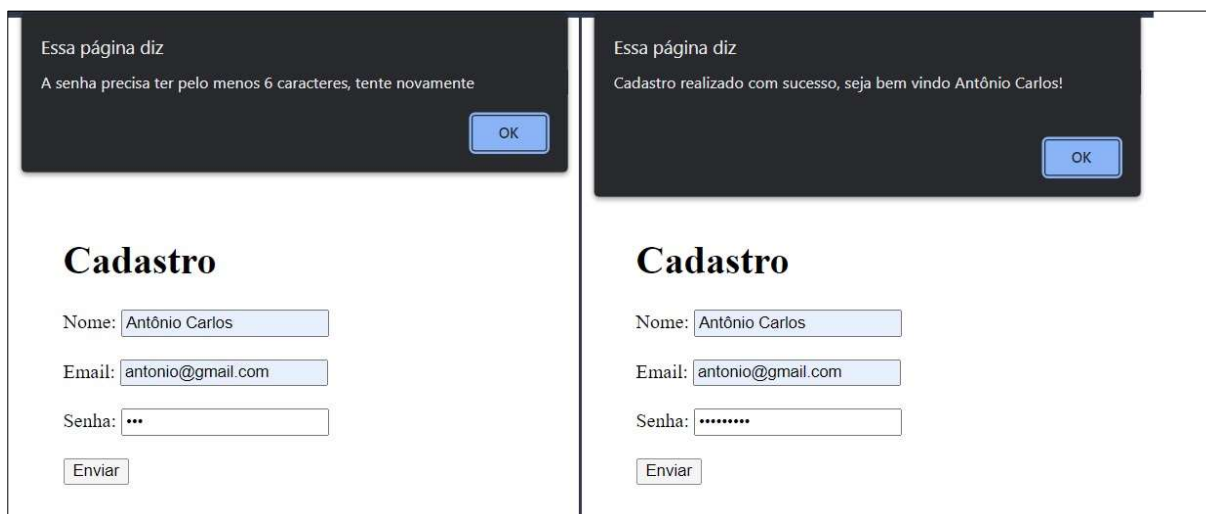
**Figura 21 - Sintaxe Function**

```
<script>
    function enviado(){
        var nome = document.forms["cadastro"]["nome"].value;
        var email = document.forms["cadastro"]["email"].value;
        var senha = document.forms["cadastro"]["senha"].value;
        if (senha.length < 6) {
            alert("A senha precisa ter pelo menos 6 caracteres, tente novamente");
            return false;
        }
        else{
            alert("Cadastro realizado com sucesso, seja bem vindo " +nome+ "!");
            return true;
        }
    }
</script>
```

Fonte: Autoria própria, 2022

Ao preencher os requisitos e enviá-lo dependendo da quantidade de caracteres da senha do usuário, ele poderá ser cadastrado ou não emitindo alertas para informar a situação do cadastro como mostra a Figura 22:



**Figura 22 - Alerts no JavaScript**

Fonte: Autoria própria, 2022.

## 2.6 PHP

De acordo com Dall'Oglio (2018), O PHP é uma das linguagens mais utilizadas no mundo. Segundo o que ele apresenta, é de se concluir que a popularidade dessa linguagem foi conquistada devido a flexibilidade e a um conjunto abrangente de classes e funções que permitem desde a criação de simples aplicações, até complexas advindas de negócios corporativos.

A linguagem PHP provém do script, que embutida no HTML permite que seu código seja encapsulado em tags especiais de início e fim. Porém, esse código só pode ser executado no servidor, ou seja, precisa-se de uma interpretação dessa linguagem para que sua execução seja processada.

Conforme Gonzaga e Birckan (2000), talvez a maior e mais significativa característica em PHP é seu suporte a uma faixa muito ampla de bancos de dados. Para eles, escrever uma página Web baseada em um banco de dados é muito mais simples, como representado na conexão da Figura 23.

**Figura 23 - Conexão com o Banco de Dados**

```
<?php
    $host = "127.0.0.1";
    $usuario = "root";
    $senha = "";
    $bd = "streamline";
    $con = mysqli_connect($host , $usuario, $senha , $bd);
?>
```

Fonte: Autoria própria, 2022.

O PHP sendo uma linguagem de script popularmente conhecida, é especialmente adequado para o desenvolvimento web. Sua potencialização é rápida e flexível, e através de tags básicas é possível criar uma estrutura de página simples, assim como mostra a Figura 24:

**Figura 24 - Estrutura básica PHP**

```
<html lang="pt-br">
    <head>
        <title>Estrutura básica de PHP</title>
        <meta charset="utf-8">
    </head>
    <body>
        <!--Corpo da página -->
        <?php
            const ConstantePHP = "Definir um valor que não será alterado";
            $variavel = "Criar variável";
            echo "Mensagem na Tela; $variavel"; //Mensagem na Tela; Criar variável
            echo ConstantePHP; //Definir um valor que não será alterado
        ?>
    </body>
</html>
```

Fonte: Autoria própria, 2022

O HTML representando a primeira camada para o desenvolvimento de uma página no navegador, é essencial para o encapsulamento do PHP, pois permite que a aplicação seja executada dentro do corpo da página, de modo que não interfira no layout. Assim, sendo salvo com sua própria extensão para distinguir sua propriedade das demais.

Todos os comandos PHP devem ser renderizados por suas respectivas tags de abertura e fechamento, <?php?>, para que a interpretação do código seja determinada a partir delas. Além de que, todas as resoluções de saída para a tela devem ser lideradas pelos comandos “print” ou “echo”, incluídas por aspas.

A sintaxe de declaração de variáveis pertencentes à essa linguagem, é localizada pelo sinal \$ como o primeiro caractere, seguido do nome atribuído e do tipo determinado, podendo ser definidas como variáveis escalares (retrabalhadas) ou não-escalares (matrizes e objetos).

Constantes atuam como identificadores para valores simples que não podem mudar durante a execução dos scripts. Utilizando a declaração “const”, somente valores dos tipos boolean, integer, float, string e arrays, indicados por possuírem apenas valores escalares são aceitos. O valor da constante é acessado facilmente com a informação de seu nome, mas também pode-se utilizar de uma função, sendo ela a “constant()” utilizada para ler seu valor caso for obtida dinamicamente.

## **2.7 Banco de Dados**

Date (2004) diz que um sistema de banco de dados é fundamentalmente um sistema computadorizado de manutenção de registros, ou seja, é um sistema cuja a função geral é recolher informações e permitir que os usuários realizem buscas e atualizem essas informações quando solicitadas. As informações podem ser algo que se relacione com um indivíduo ou organização em que o sistema será implantado e desempenhará a ajuda necessária no procedimento geral das atividades desse indivíduo ou organização.

Um banco de dados tem seu tamanho e sua complexidade variável, diz Elmasri (2005), por exemplo, uma lista que inclui nomes e endereços, pode conter algumas centenas de registros, todas com estruturas simples. Porém, o cadastro computadorizado de uma biblioteca de dados pode ter 500 mil entradas estruturadas em diferentes variedades — pelo sobrenome principal do autor, pelo assunto, pelo título —, sendo cada variedade ordenada em ordem alfabética. O *Internal Revenue Service* (IRS), órgão responsável pelo controle dos formulários de impostos preenchidos pelos contribuintes dos Estados Unidos, mantém um banco de dados mais capaz e complexo.

De acordo com Rob (2011) existe conjunto de programas que controla a estrutura do banco de dados e gerencia o acesso aos dados armazenados, e o responsável por isso é o sistema de gerenciamento de bancos de dados (SGBD). O banco de dados é como um arquivo eletrônico com uma organização ideal e o sistema de gerenciamento de banco de dados é o software que ajuda nessa organização. O conceito de banco

de dados se entende por especificar os tipos de dados, as estruturas e as restrições para aqueles dados que serão registrados em um banco de dados.

Para Elmasri (2005) a montagem de um banco de dados acontece quando é armazenado dados em alguma mídia adequada, gerenciada pelo SGBD. O manuseio contém algumas funcionalidades, como pesquisas dentro do banco de dados para recuperar e visualizar dados específicos, atualizações do banco de dados para manter as informações corretas e fornece relatórios dos dados. O compartilhamento possibilita que usuários e programas acessem o banco de dados de forma simultânea.

A proteção e manutenção do banco de dados também são funções importantes do SGBD, Ele protege o sistema contra o funcionamento inadequado ou erros e falhas (cashes) no hardware e software, a segurança é realizada em combate a acessos não autorizados ou maliciosos. Um banco de dados deve durar por muitos anos, com isso, os SGBD devem ser eficientes para manter um sistema de banco de dados moderno, com a evolução de requisitos que se alteram ao longo do tempo.

### **2.7.1 Abordagem relacional**

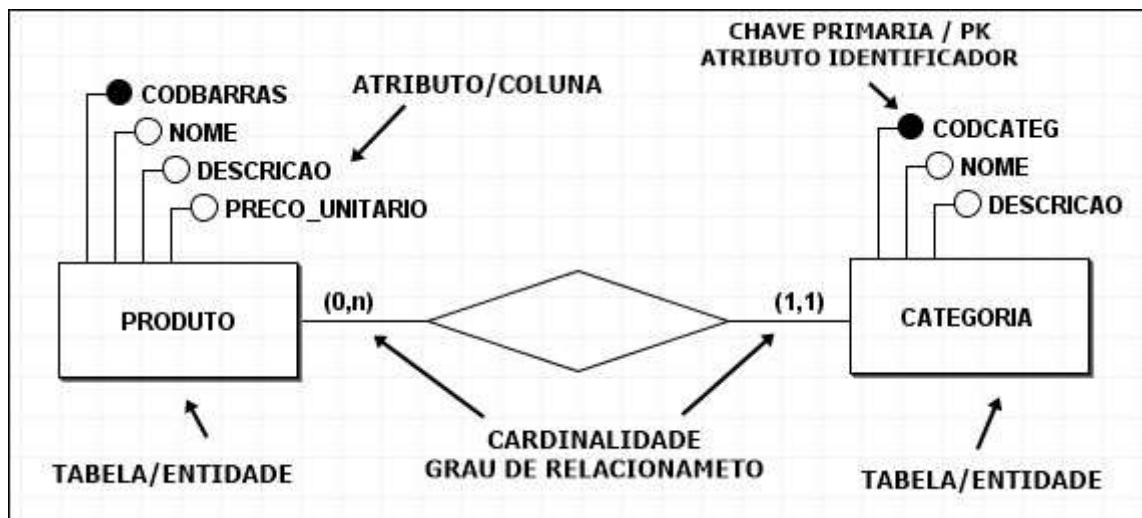
Lisboa Filho (2001) declara que é possível citar principais exemplos de modelos de banco de dados clássicos como: Rede, Hierárquico e Relacional. O relacional é o modelo mais utilizado nos SGBD, estando presente em 90% das organizações atualmente. Esse modelo proporcionou o conceito de operadores de consulta aos dados sendo procedente de uma base matemática.

Fernandes (2017) afirma que o Modelo Relacional tem como objetivo permitir uma independência entre os dados. Assim, as modificações na representação internas dos dados não afetam as aplicações, seja no gerenciamento dos arquivos, ordem dos registros ou caminho de acessos. O Modelo Relacional é composto por uma relação estabelecida em uma tabela bidimensional formada por colunas e linhas, caracterizadas por um nome. Dessa forma é armazenado no banco de dados essa relação que representa um objeto do mundo real.

Marcário (2005) discursa que uma relação consiste de uma estrutura de uma instância. Nessa estrutura é especificado o nome da relação e o nome e a atribuição de cada coluna, também denominada atributo ou campo da relação, como demonstrado na Figura 25. O entendimento do atributo é declarado na estrutura por seu nome e serve para delimitar os valores que este atributo pode assumir. A estrutura

só pode ser modificada por comando específicos, por conta de ser uma relação imutável ao longo do tempo.

Figura 25 - Representação gráfica do modelo relacional



Fonte: (SANTANA, 2019)

## 2.7.2 Normalização

Santanchè (2013) e Áлива (2007) explica que a normalização de dados em um banco de dados é uma boa modelagem que os documentos devem garantir para que nenhum dado seja repetido, causando inconsistências. É uma técnica para eliminar redundâncias de informações e evitar anomalias causadas pela inserção, atualização e remoção de tuplas, ou seja, listas ordenadas. Uma vantagem da normalização em bancos de dados nativos é que eles suportam propriedades multivaloradas, enquanto que a maioria dos relacionais não suporta.

França (2014) relata que uma forma normal é uma regra que deve ser seguida para que uma tabela seja bem avaliada. A forma normal sujeita o esquema de relação a uma cadeia de avaliação para garantir que ele satisfaz a forma normal. Esse processo de avaliação segue o estilo *top-down*, onde cada relação é avaliada sob os critérios das formas normais.

## 2.7.3 Dicionário de dados

Coelho (2006) menciona que o dicionário de dados tem a função de mostrar as descrições de características e atributos de um projeto específico, através da estrutura do banco de dados e seus relacionamentos entre as tabelas, possibilitando dessa forma a visão da estrutura do dicionário.

O dicionário é composto pela lista que controla os objetos e suas características, como mostrado na Figura 26, que possui uma lista de atributos que os representam a serem coletados, armazenando e ordenando as informações conforme o levantamento do trabalho é realizado.

**Figura 26 - Estrutura básica Dicionário de Dados**

<b>Cadastro Cliente</b>			
<b>Atributo</b>	<b>Tipo</b>	<b>Tamanho</b>	<b>outros</b>
Id_cli	INT	10	PRIMARY KEY
nome_cli	VARCHAR	130	
genero	VARCHAR	50	
email_cli	VARCHAR	250	
senha	VARCHAR	20	
data_nascimento	DATE		
Data_cadCli	DATE		
contato	INT	9	
cep	int	10	
numero_casa	VARCHAR	50	
complemento	VARCHAR	200	
rua_cliente	VARCHAR	100	
bairro_cliente	VARCHAR	100	

Fonte: Autoria própria, 2022

Conforme dito por Barbosa a importância do dicionário de dados se baseia no fato dos diagramas não mostrarem diversas informações importantes sobre os dados que serão tratados durante a atividade de modelagem de um Projeto de Banco de Dados (PBD), criando muitas dificuldades durante a implementação. Algumas das complicações na criação de DD pelos usuários de Sistema de Banco de Dados (SBD) são as dificuldades na manutenção e na organização do DD e complicações na leitura dos dados.

Podendo ser chamado de repositório de informações, como mencionado por Elmasri (2005) o dicionário de dados é semelhante a um catálogo do SGBD, entretanto possuindo mais diversidade de informações acessadas pelo software do SGBD e pelos usuários. Armazenando também as resoluções de trabalhos, os modelos de utilização, as explicações dos projetos das aplicações e os dados dos usuários. Através dos usuários ou do administrador do banco de dados chamado de Database

Administrator (DBA) é possível acessar, quando preciso, diretamente as informações guardadas.

## 2.8 UML

Em concordância com Booch (2006) a Unified Modeling Language (UML) está diretamente ligada a assuntos práticos e teóricos referente a linguagem gráfica contribuindo para uma melhor visualização e precisão na construção de sistemas de software. Traduzida para Linguagem Unificada de Modelagem em português, a UML proporciona modelos de estruturas padronizadas que colaboram no planejamento da arquitetura dos projetos dispondo dos processos organizacionais de elaboração até o desenvolvimento material em si.

Fowler (2014) explica que essa linguagem é comum de ser encontrada em sistemas que compõe o estilo orientado a objetos (OO) para projetos mais eficazes no processo de engenharia de software, podendo ser empregado de três formas diferentes: UML como esboço, projeto ou linguagem de programação.

- UML como esboço: é utilizado como a teoria do projeto, muitas vezes feito para esclarecer conceitos e ideias necessárias para a criação do sistema, tendo em vista que os esboços são caracterizados pelo desenho de diagramas UML antes ou depois do código de programação de acordo com o modelo de engenharia empregado. Na engenharia reversa por exemplo, essa forma de UML está ligada a explicação do funcionamento do projeto em decorrência do código já definido, ao contrário da engenharia direta que consiste em mostrar o que está para ser feito estando aberto para discursões de como prosseguir.
- UML como projeto: muitas vezes desenvolvido por projetistas ligados à área de programação, essa UML tem como objetivo a construção de projetos que descrevam detalhadamente as funções e objetivos a serem alcançados com esse sistema, tornando possível que os programadores que codifiquem essa estrutura compreendam facilmente e façam de forma literal seguindo o que está projetado, podendo ser de modo geral para o desenvolvimento do software ou apenas em uma etapa específica
- UML como linguagem de programação: consiste na especificação do sistema como um todo, desenhando diagramas UML que são agregados de forma direta para o código executável tornando a UML o código-fonte.

### 2.8.1 Levantamento de Análise e Requisitos

Machado (2018) define os requisitos de um software como a visão do usuário em relação ao projeto, eles evidenciam os atributos e as limitações que o cliente deseja para o sistema em questão sendo considerada para os programadores como metas e etapas a serem cumpridas para que satisfaçam o solicitante e aqueles que vão usufruir daquele trabalho. Considerando que os requisitos sejam os desejos empregados no software em virtude aos usuários, estes são separados em dois grupos distintos para manter a organização em seu desenvolvimento, sendo eles os requisitos funcionais e os não funcionais.

- Requisitos funcionais: são todos aqueles que descrevem o foco do projeto, o que o sistema foi feito para fazer.
- Requisitos não funcionais: estão ligados a parte de qualidade do sistema, são funções que não estão como o objetivo do sistema, mas sim descrevem o que será executado para que os objetivos funcionais sejam alcançados, como o programador fará com que o projeto tenha uma melhor performance para a execução dos seus objetivos.

Levando em consideração os conceitos de requisitos impostos por Machado (2018) vale ressaltar que Sommerville (2011 apud CHICHINELLI, 2017) conclui que se faz necessário 4 etapas para a elaboração do levantamento de requisitos: estudo de viabilidade, elicitação e análise de requisitos, especificação de requisitos e validade de requisitos.

- Estudo de viabilidade: tem como base analisar se os requisitos propostos pelo cliente estão acessíveis para desenvolvimento levando em conta o orçamento proposto e as tecnologias a disposição para o feito concluindo se será possível obter avanços nessa etapa.
- Elicitação e análise de requisitos: consiste no procedimento de organização dos requisitos, aqui é imposto problemas existentes e com a possibilidade de existir, sendo necessário uma discussão sobre como prosseguir, é onde ocorre a elaboração de mais requisitos que contribuem para a execução do projeto e então são classificados de acordo com a ordem de prioridade e agrupado por suas semelhanças, resolve os conflitos entre requisitos e documentam o processo.



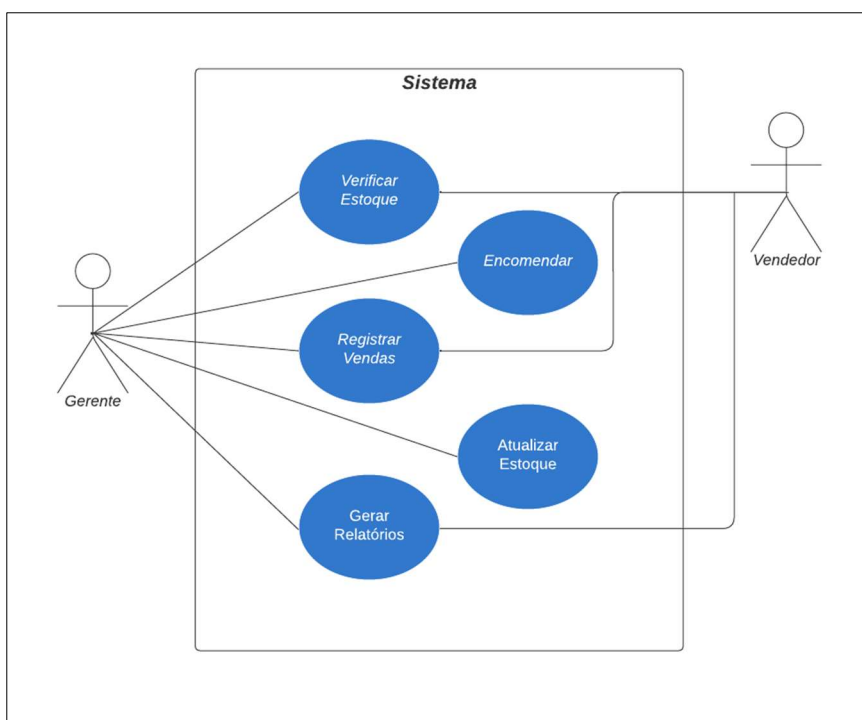
- Especificação de requisitos: é utilizada para a identificação dos requisitos analisados na etapa anterior, organiza os registros de forma que seja separado os requisitos feitos para o sistema e os feitos para o usuário.
- Validade de requisitos: será analisado a consistência dos requisitos selecionados, se estão de acordo com a realidade como uma revisão da documentação para que eles sejam completos e bem estabelecidos.

### 2.8.2 Diagrama de Caso de Uso

Larman (2000) afirma que diagramas de caso de uso estão diretamente ligados a análise de requisitos de Projetos OO sendo considerado uma das técnicas de levantamento destes. Guedes (2018) completa dizendo que esses requisitos são necessariamente interações do usuário com o software, que são chamados de atores.

Guedes (2018) explica que as funcionalidades são definidas em textos diretos baseando-se em verbos de ação dentro de balões interligados a atores que representam aqueles que irão interagir com o sistema, como mostra o diagrama da Figura 27, básico de um comércio:

Figura 27 - Estrutura básica Diagrama de Caso de Uso



Fonte: Autoria própria, 2022.

Aqueles que de fato fazem a compra são os clientes em questão, no entanto, regularmente não são eles que tem o devido acesso ao sistema de cobrança, logo,

não podem ser considerados atores. Guedes (2018) acrescenta que, esse método de levantamento de requisitos costuma obter uma linguagem mais simples, tornando-se de fácil interpretação para aqueles que o analisarem, sendo de grande ajuda em documentações que buscam explicar o projeto podendo ter diversas formas de transcrever e documentar esse diagrama como por meio de tabelas, como mostra a Tabela 1:

**Tabela 1 -Tabela de Caso de Uso**

Comércio (nome do Caso de Uso)	UC01 – Atualizar Estoque
Caso de Uso Geral	
Ator Principal	Gerente

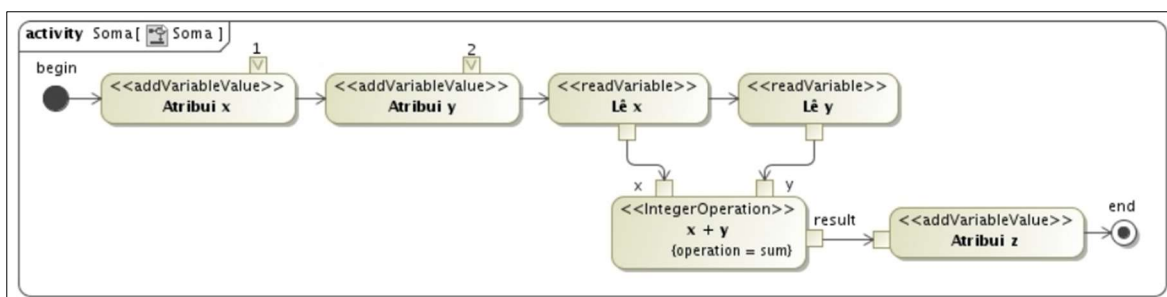
Fonte: Autoria própria, 2022.

### 2.8.3 Diagrama de Atividades

Conforme Ferreira (2010), diagramas de atividade tem como objetivo principal proporcionar de forma clara e intuitiva exemplificações de sequências de atividades feitas para mostrar um procedimento computacional através de características dinâmicas.

Ferreira (2010) complementa dizendo que sua estrutura é composta atividades e suas ações, sendo assim, as atividades seriam a descrições do que será feito e suas ações representado na Figura 28:

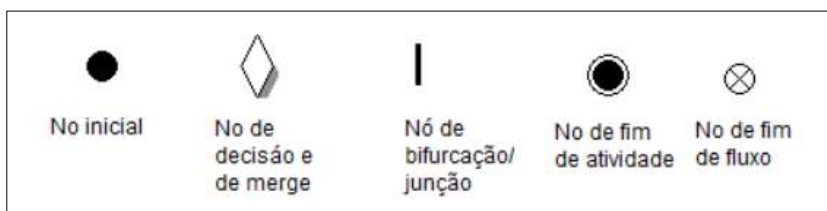
**Figura 28 - Estrutura básica Diagrama de Atividades**



Fonte: (FERREIRA, 2010)

De Mello (2011) afirma que é fundamental o uso de nós de controle para a elaboração dos diagramas pois conduzem tráfego de controle e de dados, possibilitando uma melhor compreensão das ações realizadas, podendo serem representadas como mostra a Figura 29:

**Figura 29 - Nós de controle**



Fonte: (DE MELLO, 2011)

- **Nó inicial:** como o nome já diz, consiste na marcação do começo da ação orientando o fluxo de controle, sendo possível aparecer apenas uma vez por atividade.
- **Nó de decisão e de *merge*:** é possível haver mais fluxos no decorrer do diagrama e este nó proporciona a escolha, podendo deliberar em qual prosseguir dentre as diversas alternativas possíveis de acordo com suas condições particulares preestabelecidas sendo interligados por um nó *merge*.
- **Nó de bifurcação/junção:** oferece a possibilidade de mais fluxos sendo executados simultaneamente em uma atividade, sendo possível condições de restrição para sua efetuação ou utilizando o nó junção (*join*) para uni-los de forma espontânea ou através de especificação com o *JoinSpec*.
- **Nó de fim de atividade:** tem como finalidade mostrar o final de uma atividade, mas ao contrário do nó inicial, é possível que ele aparece mais de uma vez na atividade, pois indica que uma atividade foi finalizada, no entanto, o fluxo de controle permanece, podendo haver outras atividades.
- **Nó de fim de fluxo:** permite um desencaminhamento de atividade possibilitando um laço de repetição para uma condição específica, ou seja, reproduz uma mesma situação inúmeras vezes até que a condição seja cumprida sem necessariamente finalizar aquela atividade, podendo dar continuidade a ela depois.

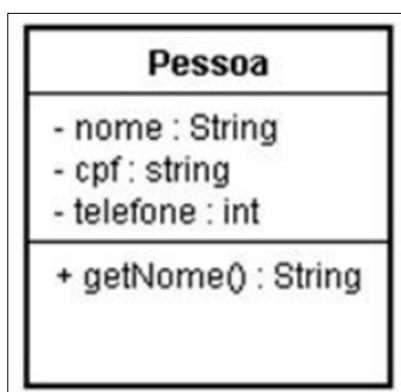
#### 2.8.4 Diagrama de Classes

De acordo com Almeida (2006) o diagrama de classe mostra de maneira estática a estruturas dos sistemas através de classes, interfaces e seus relacionamentos, sendo assimilado mais facilmente do que o código fonte, pois auxiliam os desenvolvedores a captar as implementações dos programas, arquiteturas, comportamentos, entre outros, se tornando desta maneira um dos diagramas de modelagem mais importantes

e mais compreendidos da UML. Referenciando Almeida (2006) desenvolvedores ao decorrer do projeto usufruem dos diagramas de classes para descrever, compreender e encontrar locais que precisam de modificações na arquitetura de softwares orientados por objetos.

Conforme dito por Neves (2017) A importância dos diagramas de classe se baseia na definição fundamental de um projeto e seus relacionamentos. Ele afirma que em quase toda classe, que são um conjunto de objetos que possuem estrutura e comportamentos iguais, é possível encontrar: nome, atributos e operações, como é possível ver na Figura 30 que representa uma classe com esses mesmos compartimentos.

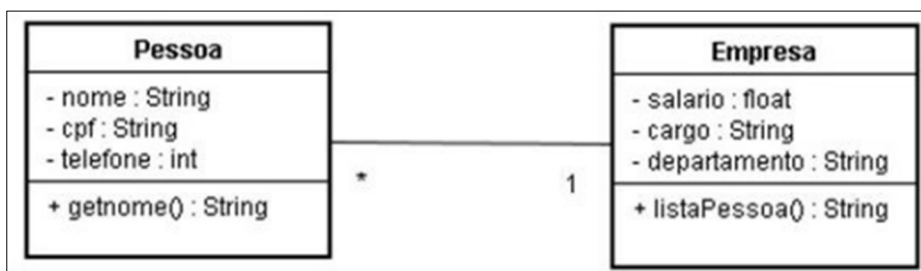
**Figura 30 - Diagrama de Classe “Pessoa”**



Fonte: (NEVES, 2017)

Como aponta Neves (2017) Pessoa, ou seja, o nome, é o compartimento que deve ser visto obrigatoriamente quando o diagrama for apresentado, os atributos e as operações são exibidos opcionalmente. Como Neves (2017) declara a Figura 31 mostra o relacionamento entre duas classes, sendo elas a classe Pessoa e a classe Empresa, tornando assim possível que a classe Empresa liste os funcionários que estão dentro da classe Pessoa.

**Figura 31 - Relação entre as classes Pessoa e Empresa**



Fonte: (NEVES, 2017)

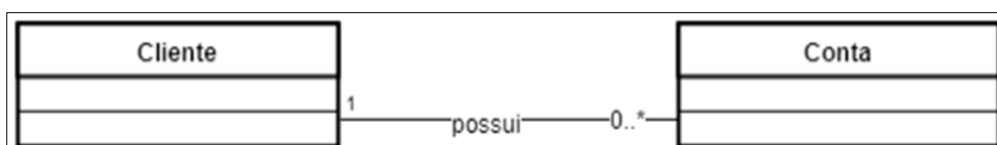
Nas palavras de Souza (2018) cada atributo possui a caracterização da informação que ele irá conter, sendo os principais: O número inteiro representado por *Integer* ou *number*; Números reais chamado de *Double*; Textos denominado como *String*; Data como *Date*; Valor lógico verdadeiro ou falso conhecido como *Boolean*.

Souza (2018) afirma que a visibilidade garante se os atributos irão ser visíveis ou não para as outras classes, possuindo três níveis: *Public* (Público) onde todas as classes relacionadas têm acesso; *Protect* (Protegido) apenas descendentes das classes podem ter acesso; *Private* (Privado) nenhuma classe tem acesso, apenas ela própria.

Como aponta Souza (2018) o objeto é a entidade que representa algo do mundo real, sendo identificado por diversas propriedades que definem o objeto, comportamentos que são as operações que os objetos podem efetuar e a identidade que faz com que um objeto seja identificado como único entre diversos objetos iguais.

Souza (2018) declara que as associações são as linhas que ligam as classes, simbolizando possíveis ligações que uma classe pode ter com a outra, na Figura 32 se pode observar as classes Cliente e Conta, que podem acessar objetos para uma possível troca de mensagens e dessa forma ajudar na execução das funções do sistema, podendo ter títulos para facilitar o entendimento.

**Figura 32 - Representação de Associação**



Fonte: (SOUZA, 2018)

As associações permitem a representação de multiplicidades, que significam o limite para quantidade de objetos que podem estar associados a outros objetos em uma

classe como dito por Souza (2018). Cada associação possui uma multiplicidade em cada extremidade da linha, como mostrado na Tabela 2, sendo representada pelos símbolos:

**Tabela 2 - Símbolos de associação**

Nome	Simbologia
Apenas Um	1 ou 1..1
Zero ou Muitos	0..* Ou *
Um ou Muitos	1..*
Zero ou Um	0..1

Fonte: Adaptado de Souza (2018)

### 2.8.5 Diagrama de Sequência

Segundo Silva (2009) o diagrama de sequência representa a sequência global de métodos em classes diferentes de maneira simples e lógica. Descrevendo o comportamento de um caso de uso e mostrando como o conjunto de objetos interagem através das mensagens passadas entre os objetos. Esse diagrama se baseia nas informações dos casos de uso, se tornando uma complementação com mais informações acrescentadas e detalhadas. Conforme mencionado por Silva (2009), na Tabela 3, são os conceitos essenciais do diagrama de sequência:

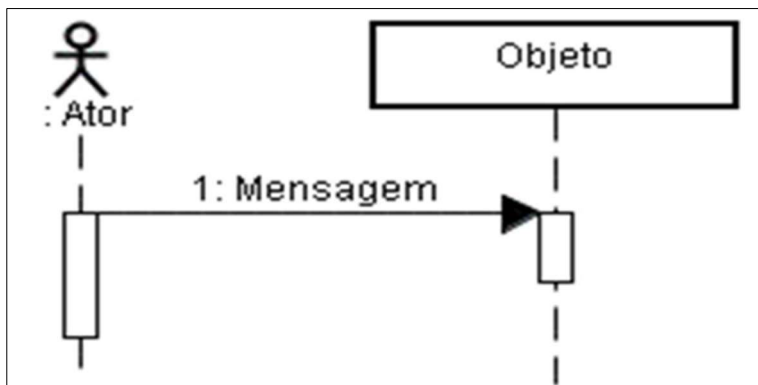
**Tabela 3 - Conceitos essenciais do diagrama de sequência**

Conceitos	Significado
Atores	Quem solicita serviços, interage com o sistema, que inicia os processos através de ações.
Objetos	Instancias das classes, por exemplo as páginas Web visitadas. Retratados como retângulos.
Gate	Ponto onde a mensagem pode ser passada para fora ou dentro do fragmento de interação.
Fragmento de interação	Exemplos: <i>Alt</i> (Alternativa), <i>Opt</i> (Opcional), <i>Break</i> (Parar), <i>Loop</i> (Repetição) entre outras.
Linha de vida	Ilustra a vida do objeto na troca de mensagens do sistema.
Mensagem	Refletem as atividades de navegação que equivalem a uma seta que indica a direção da mensagem entre dois objetos.

Fonte: Autoria própria, 2022

Tendo em vista também, na Figura 33, a estrutura base fornecida para a criação dos diagramas:

**Figura 33 - Estrutura básica Diagrama de Sequência**



Fonte: (STEIN JÚNIOR, 2009)

### 2.8.6 Diagrama de Estados

Os diagramas de estado, também conhecidos como Diagrama de Máquina de Estados, conforme mencionado por Nakazato (2013) descrevem aspectos dinâmicos e comportamentais de um sistema, demonstrando graficamente o ciclo de vida dos objetos, ou como explicado por Macário (2003) tudo que um objeto pode realizar em um software, deixando a ele saber como contrapor-se a um evento, demonstrando as ações que causam a mudança de um estado para outro.

De acordo com Nakazato (2013) o estado é a situação em que o objeto se localiza no decorrer do seu ciclo de vida, podendo ser uma circunstância onde o objeto fica até outro evento acontecer, o forçando a sair dele ou a execução de uma tarefa, o final da execução provocaria a saída da situação. Sendo representado como mostrado na Figura 34, tendo um identificador que corresponde ao nome do estado.

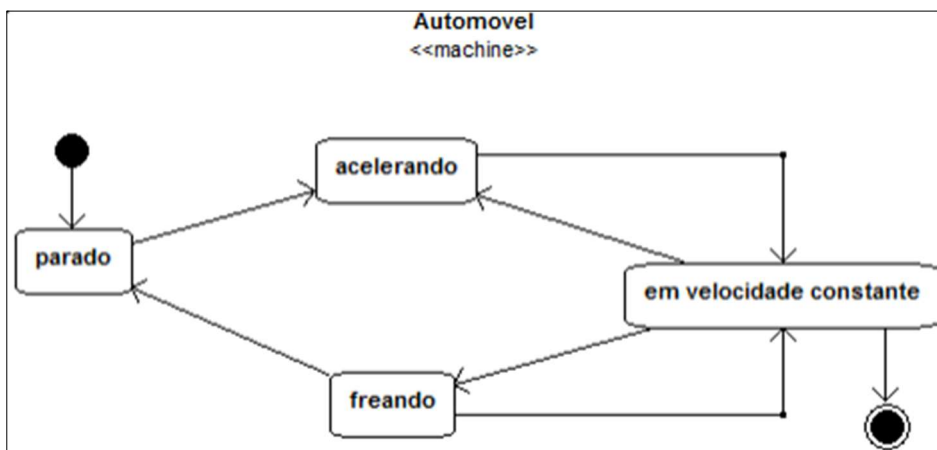
**Figura 34 - Exemplo de Estado**



Fonte: Autoria própria, 2022.

Nakazato (2013) diz que conforme o contexto modelado o estado de um objeto pode ser diferente, como ilustrado na Figura 35, onde o contexto é o movimento do automóvel.

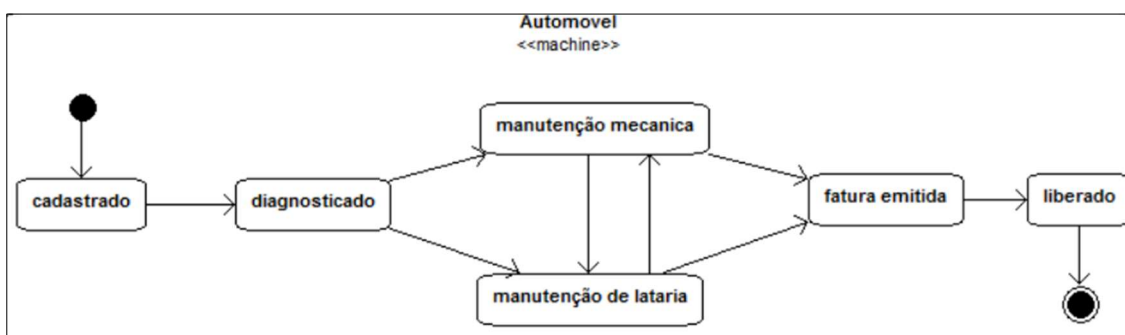
**Figura 35 - Diagrama de estados para Automóvel, tendo como foco o movimento**



Fonte: (Nakazato, 2013)

Na Figura 36 onde o contexto é uma oficina mecânica, o objeto permanece o mesmo, entretanto seu estado é completamente diferente.

**Figura 36 - Diagrama de estados para Automóvel no contexto de uma oficina mecânica**



Fonte: (Nakazato, 2013)

Nakazato (2013) declara que o estado inicial é ilustrado por um círculo fechado, como demonstrado na Figura 37 (a), sendo obrigatório ter apenas um, e o estado final é demonstrado através de um círculo fechado com uma borda dupla, como mostra a Figura 37 (b), sendo possível ter mais de um.

**Figura 37 - (a) Estado inicial (b) Estado final**



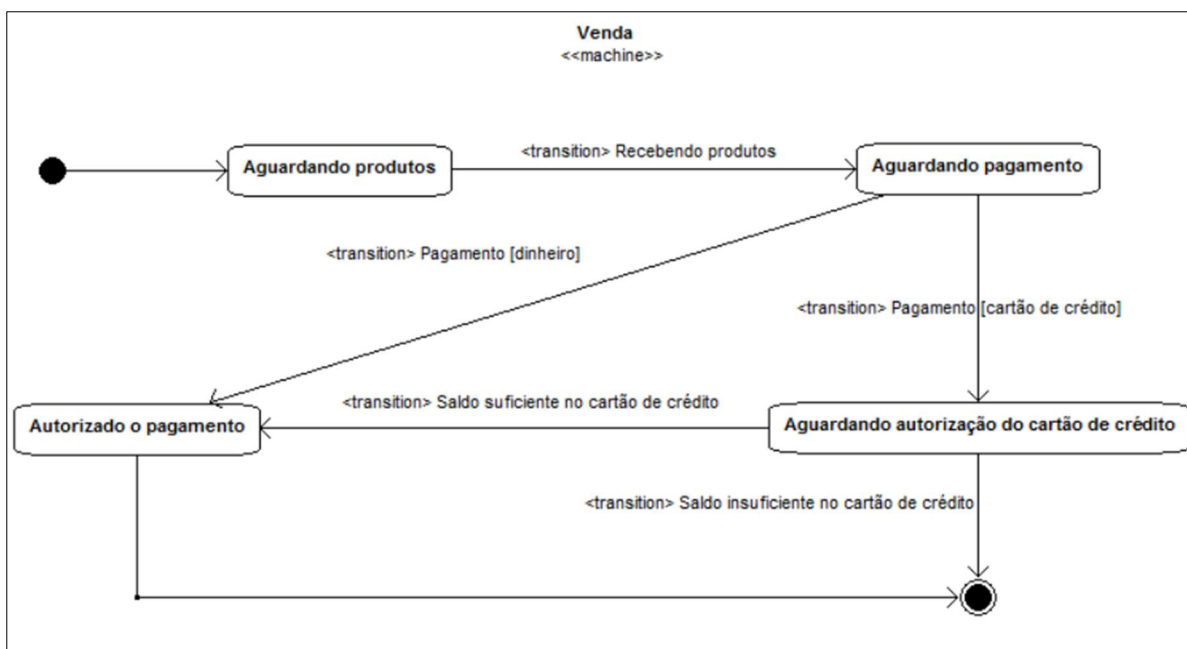
Fonte: (Nakazato, 2013)



Para Nakazato (2013) as transições são as associações de um estado ao outro, sendo ilustradas por uma linha que os conecta, sendo possível uma transição retornar para o mesmo estado. Referenciando Nakazato (2013) uma transição contém um evento agregado, que ocasionam a transição dos estados, a Figura 38 demonstra essas transições a partir dos eventos "Recebendo produtos", "Pagamento [dinheiro]", "Pagamento [cartão de crédito]", "Saldo insuficiente no cartão de crédito" e "Saldo suficiente no cartão de crédito".

Conforme mencionado por Nakazato (2013) uma condição de guarda se refere a uma expressão que se converte em um valor lógico, ou seja, verdadeiro ou falso, fazendo com que o evento ligado a ele só aconteça caso verdadeiro, sendo sempre apresentada entre colchetes, como é possível ver na Figura 38 através das condições de guarda "[dinheiro]" e "[cartão de crédito]".

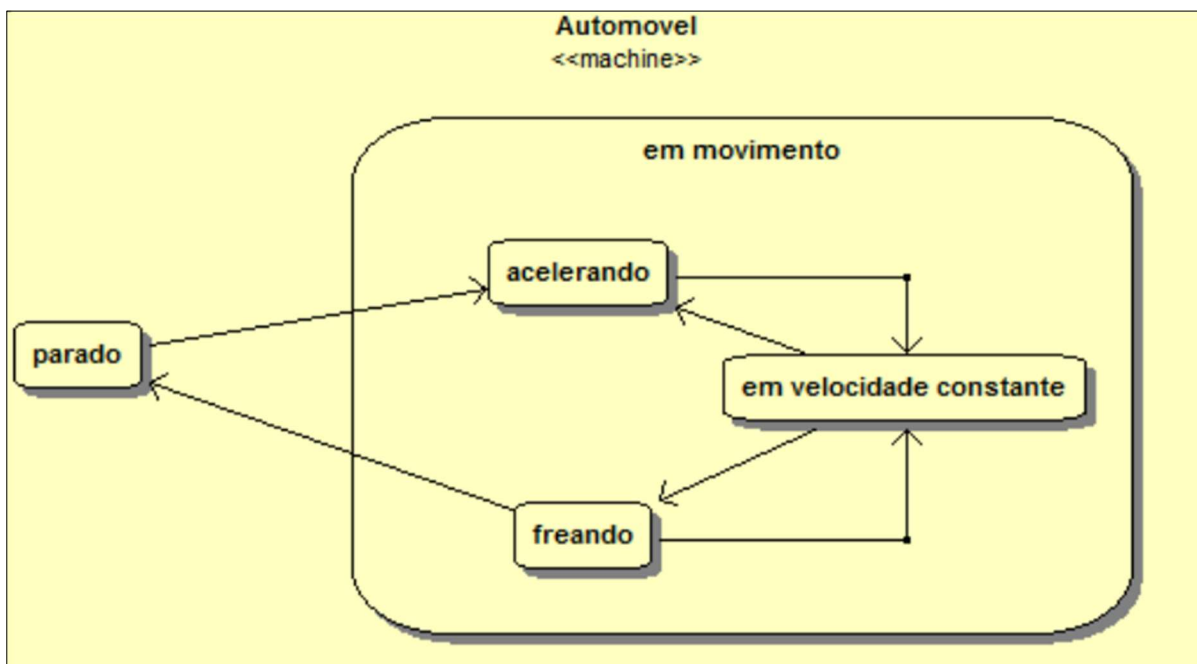
**Figura 38 - Diagrama de estados de Venda**



Fonte: (Nakazato, 2013)

Como aponta Nakazato (2013) em diagramas complexos e detalhados é utilizado os estados compostos, tendo relação com a descrição dos diagramas. Através da Figura 39 é possível ver o superestado "em movimento" que possui os subestados "acelerando", "freando" e "em velocidade constante", que acaba que por se tornar a versão mais detalhada da Figura 35.

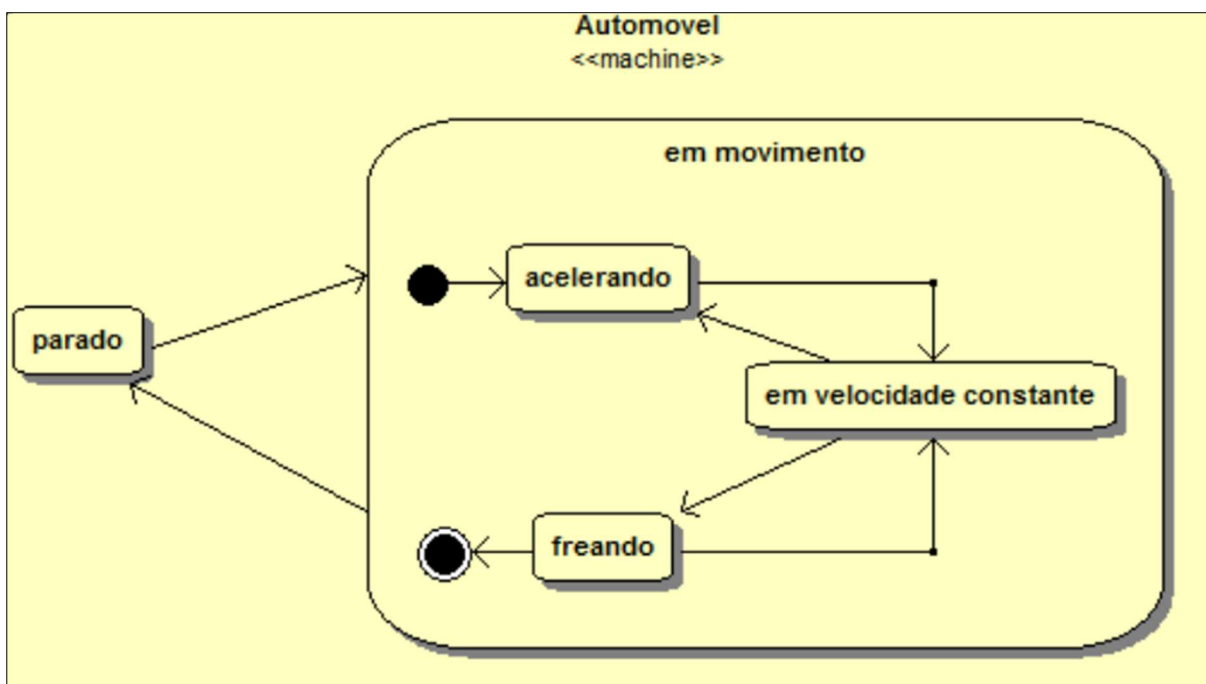
Figura 39 - Modelagem de estados para Automóvel, com estado composto



Fonte: (Nakazato, 2013)

Tendo também a possibilidade de ser modelado conforme ilustra a Figura 40, que ainda permanece com o mesmo significado, porém com o diferencial de possuir um estado final e um pseudo-estado inicial no superestado.

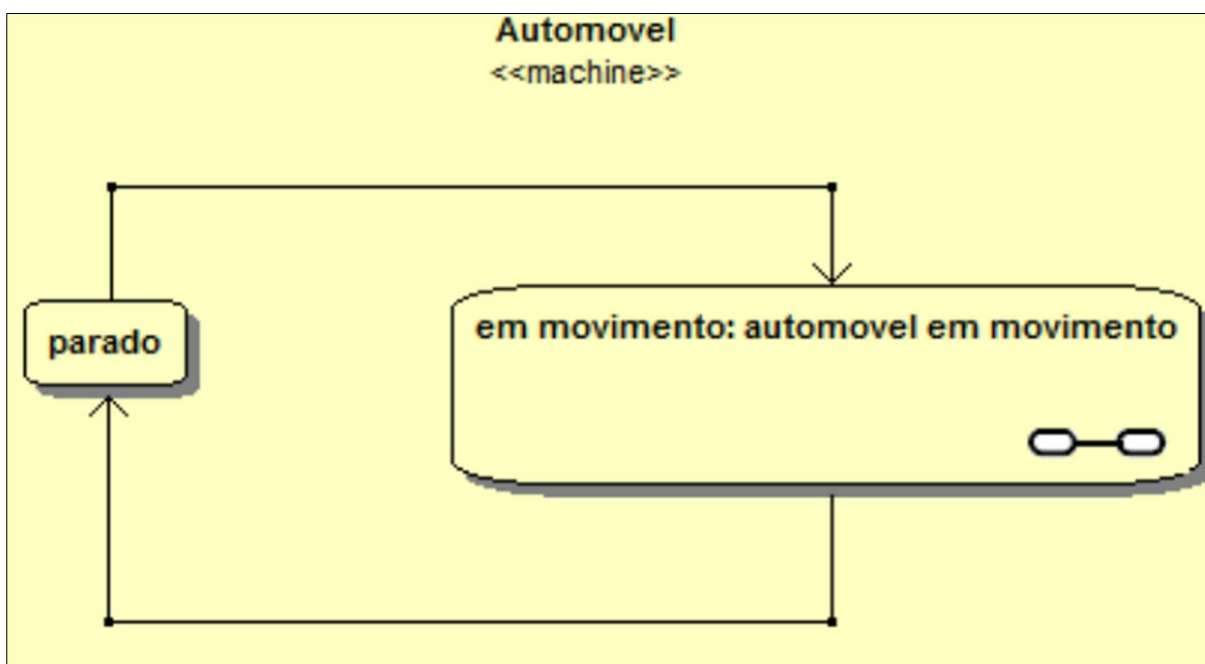
Figura 40 - Modelagem alternativa para Automóvel, com estado composto



Fonte: (Nakazato, 2013)

Nakazato (2013) afirma que em diagramas mais complexos, onde o uso de muitos estados compostos decorre em um diagrama com muitos elementos e pouca legibilidade, é utilizado submáquinas, onde os subestados existem em outro diagrama e o superestado equivale a uma submáquina. Como é possível ver na Figura 41 a submáquina "em movimento" deduz que existe outro diagrama de estados, no qual o conteúdo deve ser precisamente o grupo de subestados presentes em "em movimento" da Figura 40.

**Figura 41 - Modelagem de Automóvel com estado submáquina**



Fonte: (Nakazato, 2013)

## 2.9 Paradigma de Orientação a Objetos

Para Durães (2021) o paradigma de Orientação a Objeto é uma simulação do mundo real através dos princípios computacionais que definem as principais características dos sistemas Orientados a Objeto, tendo o diferencial de usar conceitos de herança e polimorfismo.

Na Tabela 4, conforme Durães (2021), esses são os conceitos do paradigma de Orientação a Objeto são:

**Tabela 4 - Significado dos conceitos do paradigma de OO**

Conceitos	Significado
Objeto	Representa algo do mundo real.
Atributos	São variáveis que guardam as características de cada objeto, como demonstrado na Figura 34.
Métodos	Funções das ações que cada objeto poderá realizar, tornando possível visualizar o que é possível executar com os objetos.
Classe	Modelo de conjuntos de objetos que possuem comportamentos e características semelhante. A classe descreve as propriedades, métodos, atributos e menciona as funcionalidades existentes dos objetos.

Fonte: Autoria própria, 2022

Podendo pôr em prática seus conceitos de acordo com a Figura 42:

**Figura 42 - Atributos do objeto Gato**

Gato	
Nome:	Tufo
Idade:	2 anos
Tamanho dos pelos:	Curtos
Cor dos olhos:	Amarelo
Peso:	5 kg

Fonte: (DURÃES, 2021)

Dado os atributos, é possível identificar, através da Figura 43, as ações que o objeto em questão é capaz de executar:

**Figura 43 - Métodos do objeto Gato**

Gato
Nome Idade Tamanho dos pelos Cor dos olhos Peso
Comer() Dormir() Correr() Miar()

Fonte: (DURÃES, 2021)

Conforme Durães (2021) podemos ver na Tabela 5 os pilares do paradigma de Orientação a Objeto:

**Tabela 5 - Os pilares do paradigma de OO**

<b>Pilares</b>	<b>Significado</b>
Abstração	A concentração apenas nos aspectos fundamentais dos objetos.
Encapsulamento	Visualização apenas do resultado, ocultando informações irrelevantes como a forma que algo é realizado, por exemplo.
Herança	Relacionamento entre classes onde a classe filha é a extensão de outra classe chamada mãe ou superclasse, fazendo com que a classe filha herde da classe mãe as características, como métodos e atributos, tendo a possibilidade de adicionar características próprias além das herdadas.
Polimorfismo	Quando duas ou mais classes chamam um método em comum, entretanto tem reações diferentes.

Fonte: Autoria própria, 2022.

### 3 DESENVOLVIMENTO

Essa seção terá como objetivo a documentação do processo de desenvolvimento do site StreamLine a partir dos dados recolhidos através de estudos de UML a construção do banco de dados MYSQL e as tecnologias utilizadas para a sua elaboração.

#### 3.1 Banco de Dados

Nosso site é organizado por meio de uma abordagem relacional e um dicionário de dados, que facilitaram a composição e a compreensão de nossas tabelas no banco de dados StreamLine.

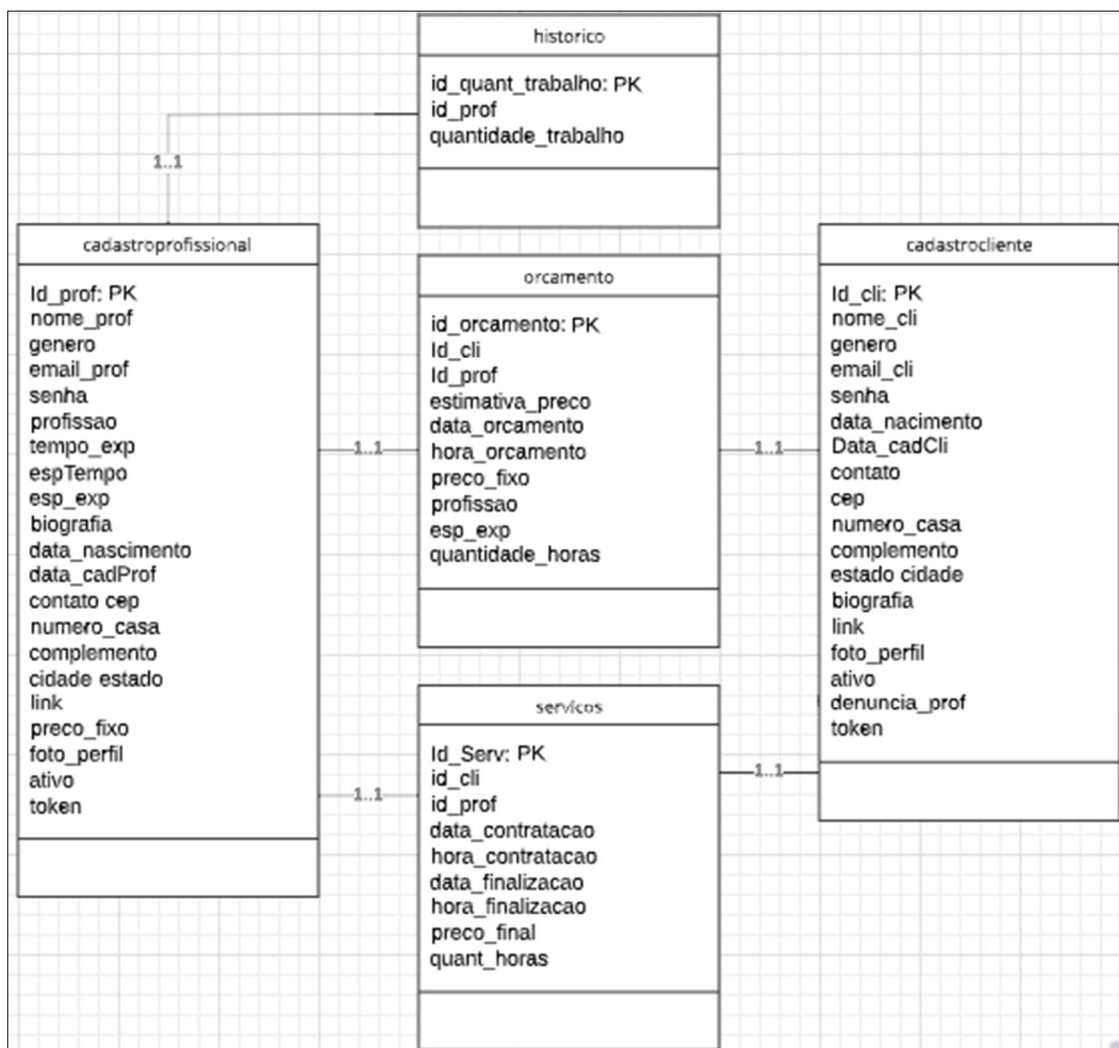
##### 3.1.1 Abordagem Relacional

Para explicar graficamente as relações e cardinalidades entre as entidades encontradas no banco de dados StreamLine, temos as Figuras 44, 45 e 46, que demonstram de forma mais ilustrativa as entidades e seus dados, tendo a sigla PK, que significa *Primary key* (Chave Primária), para representar o seu identificador único.

Na Figura 44, é possível ver a entidade cadastroprofissional, que armazena todos os autônomos cadastrados no site, possuindo relacionamento com a entidade historico, onde se armazena a quantidade de trabalho realizado pelo profissional, com a cardinalidade de um profissional para um histórico. Com a entidade orcamento, lugar que é guardado o valor calculado pelo site através do preço fixo do trabalhador, temos a cardinalidade de um profissional para um orçamento. Com a entidade servicos, no qual guarda as informações do trabalho, obtemos a cardinalidade de um profissional para um serviço.

Também temos as relações com a entidade cadastrocliente, que possui relacionamento com a entidade orcamento, assim adquirindo a cardinalidade de um cliente para um orçamento. E com a entidade servicos, de um cliente para um serviço.

Figura 44 - Der: Histórico, Orçamento e Serviços

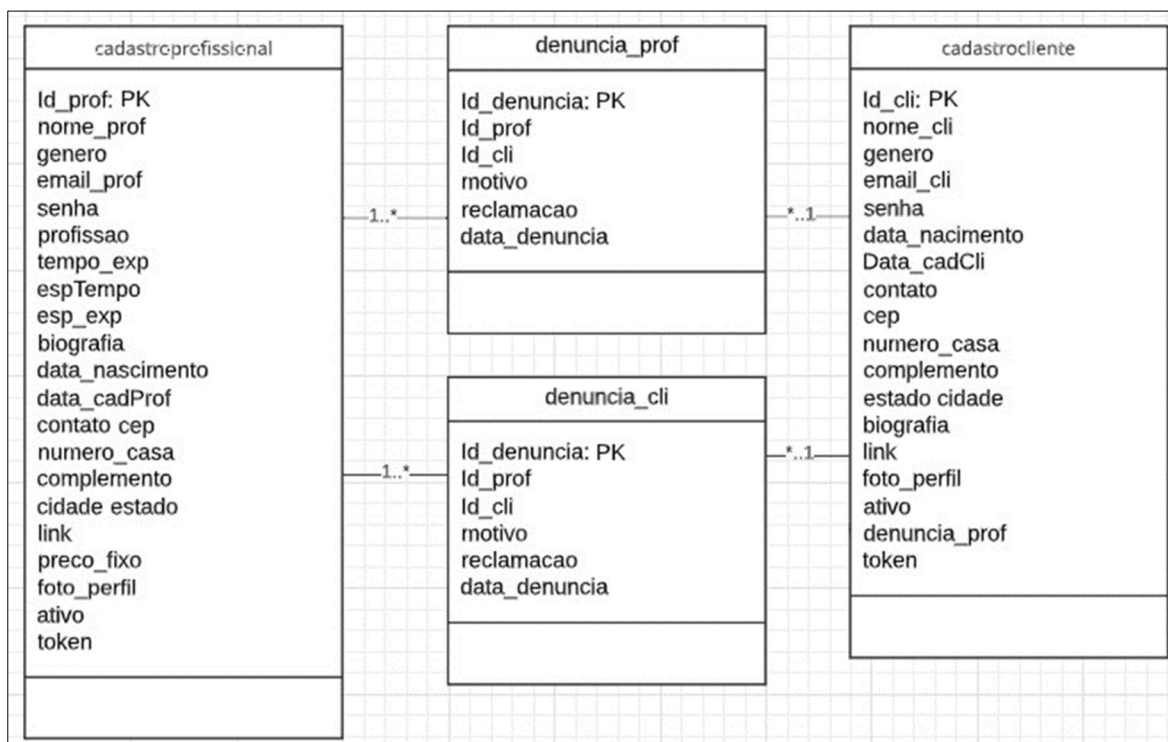


Fonte: Autoria própria, 2022.

É encontrado na Figura 45, a entidade cadastroprofissional que possui relacionamento com a entidade denuncia\_prof, onde se armazena as denúncias que o profissional recebeu do cliente, sendo sua cardinalidade de um profissional para muitas denúncias. A relação entre a entidade cadastroprofissional e a entidade denuncia\_cli, é guardado as denúncias que o profissional realizou para o cliente, tendo a cardinalidade de um profissional para muitas denúncias.

A entidade cadastrocliente possui relacionamento com a entidade denuncia\_prof, sendo nesse caso a denúncia que o cliente realiza para o profissional e sua cardinalidade é de um cliente para muitas denúncias. Já na relação entre as entidades cadastrocliente e denuncia\_cli, sendo nessa situação a denúncia que o cliente recebe do profissional, sua cardinalidade é de um cliente para muitas denúncias.

Figura 45 - Der: Denúncia



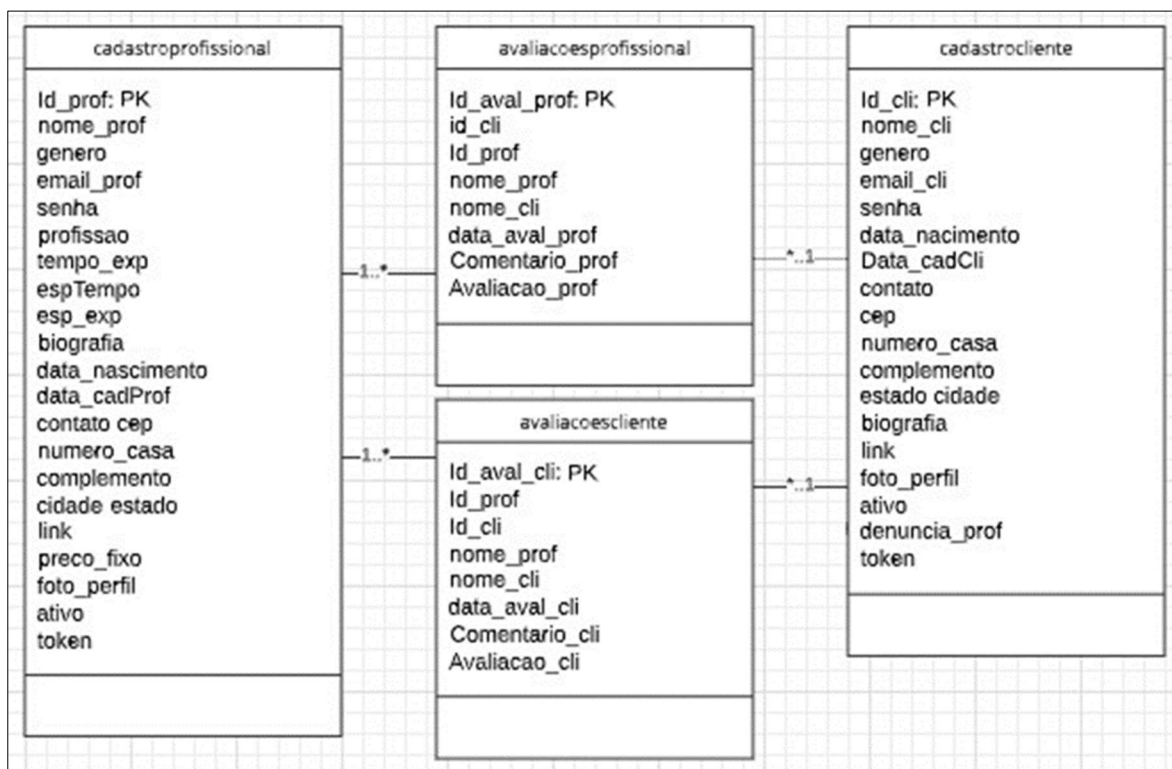
Fonte: Autoria própria, 2022.

É possível observar na Figura 46, a entidade cadastroprofissional no qual tem relacionamento com a entidade avaliacoeprofissionais, onde as avaliações que o profissional recebeu do cliente são armazenadas, e tendo como cardinalidade de um profissional para muitas avaliações. Na relação entre as entidades cadastroprofissional e avaliacoesciente, no qual é guardado as avaliações que o profissional realizou para o cliente, a cardinalidade é de um profissional para muitas avaliações.

Já a entidade cadastrocliente possui relacionamento com a entidade avaliacoeprofissionais, que nesse caso é a avaliação que o cliente realiza para o profissional, tendo a cardinalidade de um profissional para muitas avaliações. Na relação entre as entidades cadastrocliente e avaliacoesciente, sendo nessa situação a avaliação que o cliente recebe do profissional, sua cardinalidade é de um profissional para muitas avaliações.



Figura 46 - Der: Avaliação



Fonte: Autoria própria, 2022.

### 3.1.2 Dicionário de Dados

Organizamos em tabelas simples as descrições das características dos atributos encontrados em nossas tabelas do banco de dados StreamLine, para melhor entendimento de como ocorre a troca de dados e como esses dados são resgatados por outras tabelas em nosso banco. Na Figura 47 temos os atributos para o cadastro de clientes ou contratantes de profissionais autônomos, sendo descrito esses atributos em tipo tamanho e outros, tratando-se o termo “outros” como a descrição do tipo de chave existente na tabela do banco de dados.

Figura 47 - Tabela cadastrocliente StreamLine

<b>cadastrocliente</b>			
<b>Atributo</b>	<b>Tipo</b>	<b>Tamanho</b>	<b>outros</b>
Id_cli	INT	11	PRIMARY KEY
nome_cli	VARCHAR	130	
genero	VARCHAR	50	
email_cli	VARCHAR	250	
senha	VARCHAR	20	
data_nascimento	DATE		
Data_cadCli	DATE		
contato	INT	9	
cep	INT	10	
numero_casa	VARCHAR	50	
complemento	VARCHAR	200	
estado	VARCHAR	200	
cidade	VARCHAR	200	
biografia	TEXT		
link	VARCHAR	100	
foto_perfil	VARCHAR	200	
ativo	TINYINT	1	
token	VARCHAR	200	

Fonte: Autoria própria, 2022.

Na Figura 48 temos a tabela organizada dos atributos necessários para o cadastro de prestadores de serviços, os profissionais autônomos.

Figura 48 - Tabela cadastroprofissional StreamLine

<b>cadastroprofissional</b>			
<b>Atributo</b>	<b>Tipo</b>	<b>Tamanho</b>	<b>outros</b>
Id_prof	INT	10	PRIMARY KEY
nome_prof	VARCHAR	130	
genero	VARCHAR	50	
email_prof	VARCHAR	250	
senha	VARCHAR	20	
profissao	VARCHAR	180	
tempo_exp	INT	10	
espTempo	VARCHAR	200	
esp_exp	VARCHAR	120	
biografia	TEXT		
data_nascimento	DATE		
data_cadProf	DATE		
contato	VARCHAR	20	
cep	VARCHAR	10	
numero_casa	VARCHAR	50	
complemento	VARCHAR	200	
cidade	VARCHAR	200	
estado	VARCHAR	200	
link	VARCHAR	100	
preco_fixo	DECIMAL	10,2	
foto_perfil	VARCHAR	200	
ativo	TINYINT	1	
token	VARCHAR	200	

Fonte: Autoria própria, 2022.

Criamos uma tabela para a organização dos atributos relacionados ao serviço, com atributos do profissional e do cliente, informações sobre o orçamento, detalhes da contratação e tempo de finalização desses serviços, como na Figura 49:

Figura 49 - Tabela servico StreamLine

<b>servico</b>			
Atributo	TIPO	Tamanho	outros
Id_Serv	INT	11	AUTO_INCREMENT/PRIMARY KEY
Id_prof	INT	10	
Id_cli	INT	11	
data_contratacao	DATE		
hora_contratacao	TIME		
data_finalizacao	DATE		
hora_finalizacao	TIME		
preco_final	DECIMAL	10,2	
quant_horas	DECIMAL	10,2	

Fonte: Autoria própria, 2022.

A Figura 50 apresenta a tabela com atributos em que armazenam detalhes sobre o orçamento.

Figura 50 - Tabela orcamento StreamLine

<b>orcamento</b>			
Atributo	TIPO	Tamanho	outros
id_orcamento	INT	11	AUTO_INCREMENT/PRIMARY KEY
Id_cli	INT	11	
Id_prof	INT	10	
estimativa_preco	DECIMAL	10,2	
data_orcamento	DATE		
hora_orcamento	TIME		
preco_fixo	DECIMAL	10,2	
profissao	VARCHAR	180	
Esp_exp	VARCHAR	120	
quantidade_horas	DECIMAL	10,2	

Fonte: Autoria própria, 2022.

Na Figura 51 que contém a tabela pertinente ao histórico, nela é armazenada a quantidade de trabalho que o profissional prestou.

**Figura 51 - Tabela historico StreamLine**

<b>historico</b>			
Atributo	TIPO	Tamanho	outros
id_quant_trabalho	INT	11	AUTO_INCREMENT/PRIMARY KEY
id_prof	INT	11	
quantidade_trabalho	INT	10	

Fonte: Autoria própria, 2022.

Nas seguintes duas Figuras 52 e 53, temos as tabelas avaliações em que é armazenado todos os dados sobre a avaliação feita pelo profissional e contratante, como dados públicos dos mesmos, data exata de avaliação, comentário e avaliação de desempenho.

**Figura 52 - Tabela avaliacoesciente StreamLine**

<b>avaliacoesciente</b>			
Atributo	Tipo	Tamanho	outros
Id_aval_cli	INT	10	AUTO_INCREMENT/PRIMARY KEY
Id_prof	INT	10	
Id_cli	INT	10	
nome_prof	VARCHAR	130	
nome_cli	VARCHAR	130	
data_aval_cli	DATE		
Comentario_cli	TEXT		
Avaliacao_cli	INT	5	

Fonte: Autoria própria, 2022.

Na Figura 53 são apresentados os mesmos atributos da tabela correspondente a avaliação do cliente, porém direcionado a avaliação do profissional.

**Figura 53 - Tabela avaliacoeprofissional StreamLine**

<b>avaliacoeprofissional</b>			
Atributo	Tipo	Tamanho	outros
Id_aval_prof	INT	10	AUTO_INCREMENT/PRIMARY KEY
Id_prof	INT	10	
Id_cli	INT	10	
nome_prof	VARCHAR	130	
nome_cli	VARCHAR	130	
data_aval_prof	DATE		
Comentario_prof	TEXT		
Avaliação_prof	INT	5	

Fonte: Autoria própria, 2022.

As Figuras 54 e 55 exibem as tabelas de denúncias em que é armazenado informações sobre a denúncia feita pelo profissional e contratante, como dados públicos dos mesmos, data exata de denúncia, motivo da denúncia como furto, pagamento não realizado, agressões verbais ou físicas, entre outros. Assim como é armazenado a informação sobre a reclamação a ser feita.

**Figura 54 - Tabela denuncia\_cli StreamLine**

<b>denuncia_cli</b>			
Atributo	TIPO	Tamanho	outros
id_denuncia	INT	11	AUTO_INCREMENT/PRIMARY KEY
Id_prof	INT	10	
Id_cli	INT	11	
motivo	TEXT		
reclamacao	TEXT		
data_denuncia	DATE		

Fonte: Autoria própria, 2022.

Já na Figura 55 que contém a tabela associada com a denúncia do profissional, são espelhados os mesmos atributos da tabela referente a denúncia do cliente, porém direcionado para denúncias feitas pelo próprio profissional.

Figura 55 - Tabela denuncia\_prof StreamLine

<b>denuncia_prof</b>			
Atributo	TIPO	Tamanho	outros
id_denuncia	INT	11	AUTO_INCREMENT/PRIMARY KEY
Id_prof	INT	10	
Id_cli	INT	11	
motivo	TEXT		
reclamacao	TEXT		
data_denuncia	DATE		

Fonte: Autoria própria, 2022.

### 3.2 UML

A construção lógica de nosso site surgiu com levantamentos de análise e requisitos, a partir deles foram desenvolvidos diagramas de Caso de Uso, Atividades e Estado para melhor compreensão do funcionamento de todo o site juntamente com o banco de dados MYSQL.

#### 3.2.1 Levantamento de Análise e Requisitos

Como regras de negócio temos orientações e restrições, mostradas na Figura 56, que regulam todas as operações dos usuários e do nosso sistema, para colaborar com o funcionamento.

Figura 56 - Regras de Negócio StreamLine

<b>Regras de negócio</b>	
Código	Regra
RN1	O usuário poderá avaliar entre 1 e 5.
RN2	O usuário só poderá avaliar após o serviço ser prestado.
RN3	O usuário deve colocar as informações solicitadas no cadastro.
RN4	O usuário deve colocar os dados solicitados no login.
RN5	O cliente/usuário só poderá realizar outra contratação após a avaliação ou denúncia do profissional anterior.
RN6	O usuário não poderá fazer mais de um cadastro com o mesmo e-mail na mesma categoria.
RN7	O serviço ficará com pendência quando ainda estiver sendo realizado.
RN8	Cada serviço deve estar associado a dois IDs diferentes.
RN9	Os usuários devem estar registrados no site.
RN10	Os usuários do sexo biológico feminino deverão estar na faixa etária de idade entre 18 e 60 anos.
RN11	Os usuários do sexo biológico masculino deverão estar na faixa etária de idade entre 18 e 65 anos.
RN12	Enviar e-mail de alerta de exclusão de conta após 3 meses de inatividade do usuário profissional.
RN13	A conta do usuário com mais de três denúncias será suspensa.
RN14	O usuário só poderá denunciar após o serviço ser prestado.
RN15	O usuário só poderá realizar uma denúncia por cliente/profissional.
RN16	Os dados fornecidos no login precisam estar cadastrados.
RN17	Para finalização de cadastro é necessário a confirmação do e-mail.

Fonte: Autoria própria, 2022.

Nossos requisitos funcionais, como mostrado na Figura 57, descreve como nosso sistema se comporta. Define todas as ações que nosso site contém para cumprir a necessidade e expectativa dos usuários.



Figura 57 - Requisitos Funcionais StreamLine

<b>Requisitos funcionais</b>	
<b>RF1</b>	<b>Cadastrar usuários</b>
<b>Descrição</b>	Os dados informados pelo usuário deverão ser devidamente preenchidos e confirmados para o êxito do cadastro no banco de dados.
<b>Regras de negócio associadas</b>	RN3, RN4, RN6, RN8, RN9, RN10, RN11, RN16, RN17.
<b>Requisitos funcionais associados</b>	RF3, RF7.
<b>RF2</b>	<b>Excluir conta</b>
<b>Descrição</b>	A conta do usuário será excluída automaticamente caso não esteja de acordo ou viole alguma regra.
<b>Regras de negócio associadas</b>	RN12.
<b>Requisitos funcionais associados</b>	RF4, RF5.
<b>RF3</b>	<b>Gerenciar Dados dos usuários</b>
<b>Descrição</b>	O usuário deve poder consultar, editar e excluir suas informações
<b>Regras de negócio associadas</b>	RN3, RN6, RN10, RN11.
<b>Requisitos funcionais associados</b>	RF1.
<b>RF4</b>	<b>Registrar Data e Hora</b>
<b>Descrição</b>	Os cadastros dos usuários devem obter junto de seus dados a data e hora correspondentes ao momento em que foi gerado o registro.
<b>Regras de negócio associadas</b>	RN3, RN9, RN16, RN17.
<b>Requisitos funcionais associados</b>	RF1

Fonte: Autoria própria, 2022.

Na Figura 58, apresenta a continuação dos requisitos funcionais do nosso site de busca e contratação, StreamLine.

**Figura 58 - Requisitos Funcionais StreamLine Continuação**

<b>RF5</b>	<b>Gerenciar Denúncias</b>
<b>Descrição</b>	O usuário poderá registrar denúncias sobre outros perfis e comentários
<b>Regras de negócio associadas</b>	RN13, RN14, RN15.
<b>Requisitos funcionais associados</b>	RF2, RF6.
<b>RF6</b>	<b>Gerenciar Avaliações</b>
<b>Descrição</b>	Será registrado dados avaliativos referente aos usuários (cliente e profissional) e o serviço prestado como método de qualificação.
<b>Regras de negócio associadas</b>	RN1, RN2, RN5 e RN13.
<b>Requisitos funcionais associados</b>	RF5, RF7.
<b>RF7</b>	<b>Gerenciar Contratações</b>
<b>Descrição</b>	O software analisa se está sendo cumprida as diretrizes para a contratação de serviços.
<b>Regras de negócio associadas</b>	RN5 e RN8.
<b>Requisitos funcionais associados</b>	RF1, RF6.
<b>RF8</b>	<b>Suspender conta</b>
<b>Descrição</b>	A conta do usuário será suspensa automaticamente caso não esteja de acordo ou viole alguma regra.
<b>Regras de negócio associadas</b>	RN13, RN14.
<b>Requisitos funcionais associados</b>	RF4, RF5

Fonte: Autoria própria, 2022.

Os requisitos não funcionais de nosso site encaminham as funcionalidades que são entregues para os usuários, esses requisitos são relacionados com o uso do site em termos de desempenho, usabilidade, confiabilidade, segurança, disponibilidade, manutenção e tecnologias envolvidas. Assim como são apresentados na Figura 59:

**Figura 59 - Requisitos Não Funcionais StreamLine**

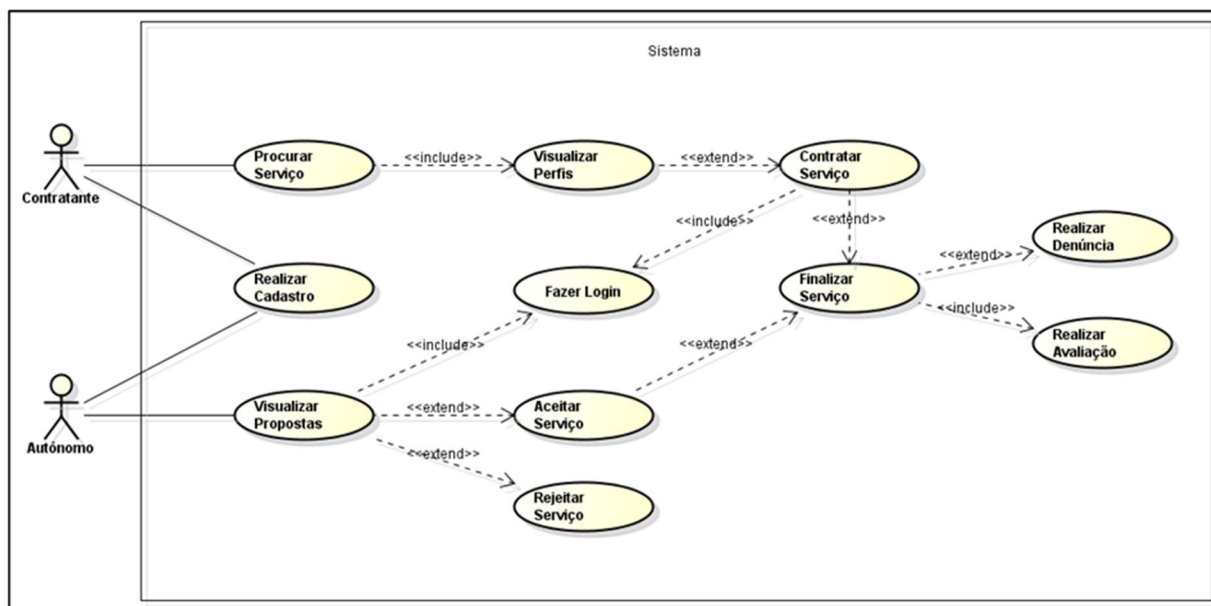
<b>Requisitos não funcionais</b>		
<b>Código</b>	<b>Descrição do requisito não funcional</b>	<b>Classificação</b>
RNF1	A interface do sistema deve ser responsiva aos dispositivos.	Suportabilidade
RNF2	O sistema deve ser prático e fácil de executar suas principais funcionalidades.	Usabilidade
RNF3	Atualização automática das informações para todos os usuários conectados no sistema online.	Performance
RNF4	O sistema deve alertar o usuário para que evite ofensas e palavras de baixo calão em sua avaliação.	Legais
RNF5	O sistema deve ser separado por categoria de profissões.	Usabilidade
RNF6	O site deve conter cores visualmente confortáveis.	Usabilidade
RNF7	O site deve ter dicas para aprimoração de perfil.	Facilidade
RNF8	O site deve conter instruções de utilização.	Facilidade
RNF9	O site deve ser minimalista para evitar poluição visual.	Usabilidade
RNF10	O site deve possuir filtragem para facilitar as buscas (perfil e serviço).	Usabilidade
RNF11	Os dados dos usuários devem ser separados em públicos e pessoais.	Éticos
RNF12	Os usuários podem se cadastrar com a conta Google.	Interoperabilidade
RNF13	Os usuários devem ter acesso as avaliações para garantir confiabilidade.	Usabilidade
RNF14	O sistema deve calcular o orçamento de acordo com os dados fornecidos pelos usuários.	Usabilidade
RNF15	O site deve conter um aviso de copyright.	Legais

Fonte: Autoria própria, 2022.

### 3.2.2 Diagrama de Caso de Uso

Com um diagrama de Caso de Uso podemos entender as ações dos usuários em nosso site, como representado na Figura 60.

Figura 60 - Diagrama de Caso de Uso StreamLine



Fonte: Autoria própria, 2022.

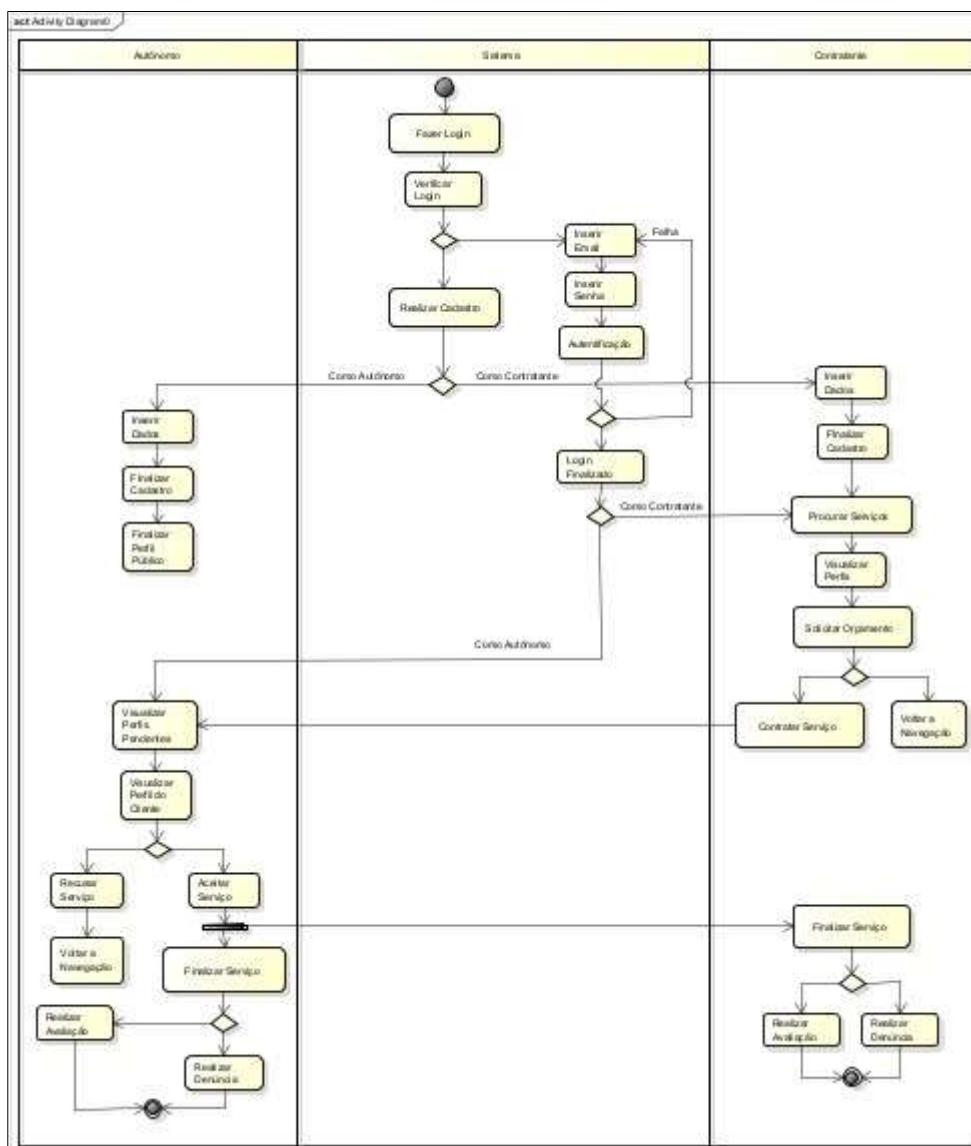
Sendo contratante, o usuário pode realizar cadastro e procurar serviços de acordo com a categoria de trabalho que necessitar, após selecionar a categoria é possível visualizar perfis de profissionais que atuam na área escolhida, escolhendo o profissional que cumprir suas exigências, o contratante pode solicitar a contratação de tal profissional, porém é preciso estar logado para passar para a próxima etapa, após o contrato ser realizado, o contratante deve definir o serviço como finalizado e para isso é obrigatório que o mesmo faça uma avaliação do profissional, a denúncia se necessário fica a critério do contratante.

Sendo profissional autônomo, o usuário pode realizar cadastro e visualizar as propostas de serviços, mas para que isso seja possível o profissional deve realizar o login, após estar logado, o mesmo pode decidir entre aceitar ou recusar o serviço proposto, caso seja recusado ele continua sua busca pelas propostas, caso seja aceito o usuário segue com o contrato e após o serviço realizado, o profissional deve definir o serviço como finalizado e para isso é obrigatório que o mesmo também faça uma avaliação do contratante, a denúncia se demandar fica a critério do profissional.

### 3.2.3 Diagrama de Atividades

O diagrama de atividade ilustra todas as atividades realizadas pelo sistema, como é demonstrado na Figura 61, onde o diagrama está como um todo. Para a melhor visualização contaremos com o diagrama dividido nas Figuras 63 e 64, onde temos a contratação e a finalização do serviço. O diagrama das Figuras 62, 63 e 64 possuem o fluxo do autônomo encontrado na parte esquerda, do sistema localizado no meio e do contratante na parte direita.

Figura 61 - Diagrama de Atividades Inteiro StreamLine

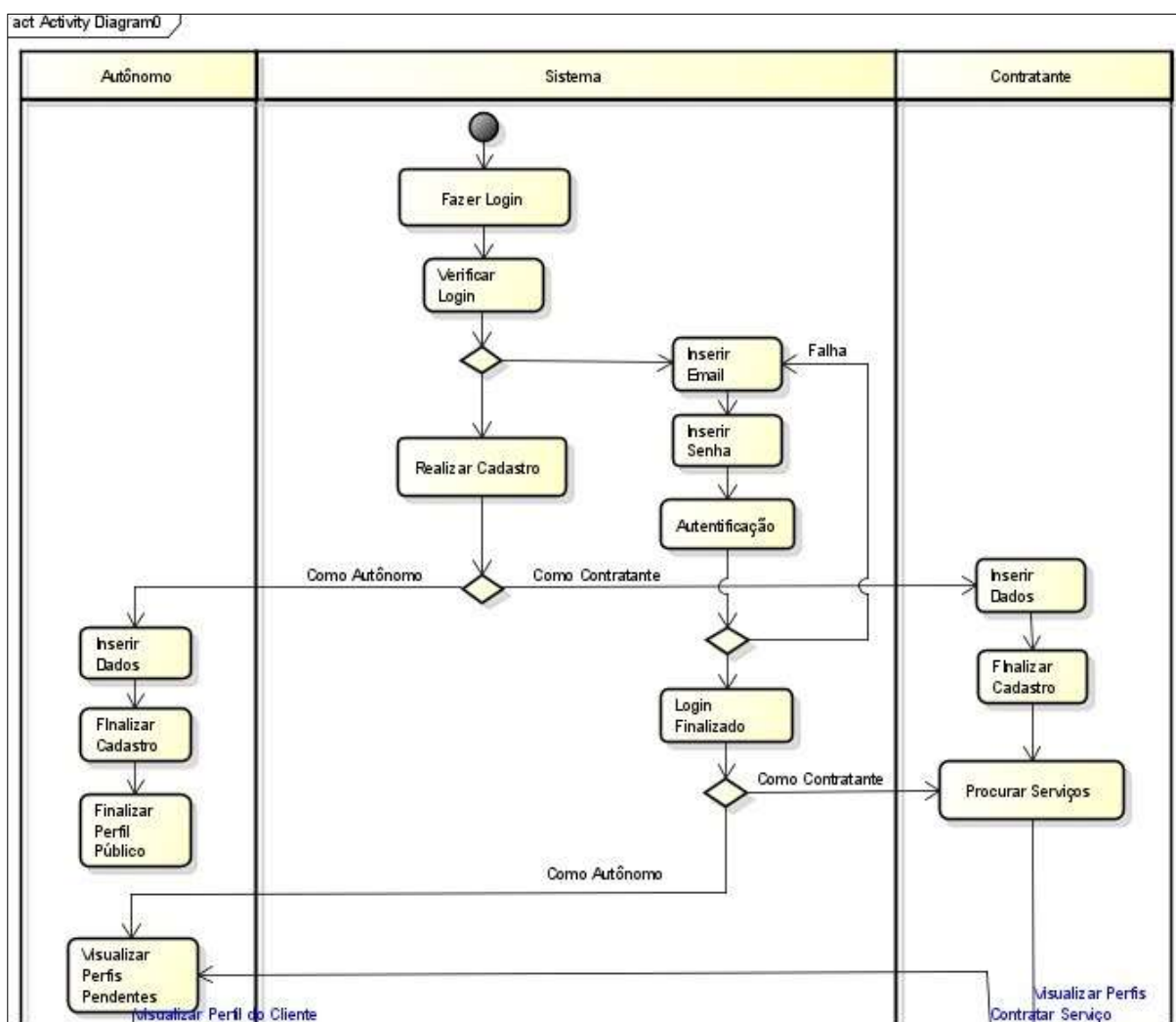


Fonte: Autoria própria, 2022.

Como mencionado posteriormente, a Figura 62 ilustra o fluxo do início do sistema, onde se tem a verificação do login, podendo encaminhar o usuário para o restante da

parte do login, onde o sistema analisa se ele está na entidade de contratante ou de cliente, o redirecionando para a visualização de perfis pendentes, caso seja autônomo, ou para a procura de serviços, caso contratante, e em caso de não ter o login o usuário irá para a realização do cadastro, que possui as opções de autônomo e contratante, que ao ser finalizado encaminha o usuário, se cadastrado como autônomo, para a finalização do perfil público.

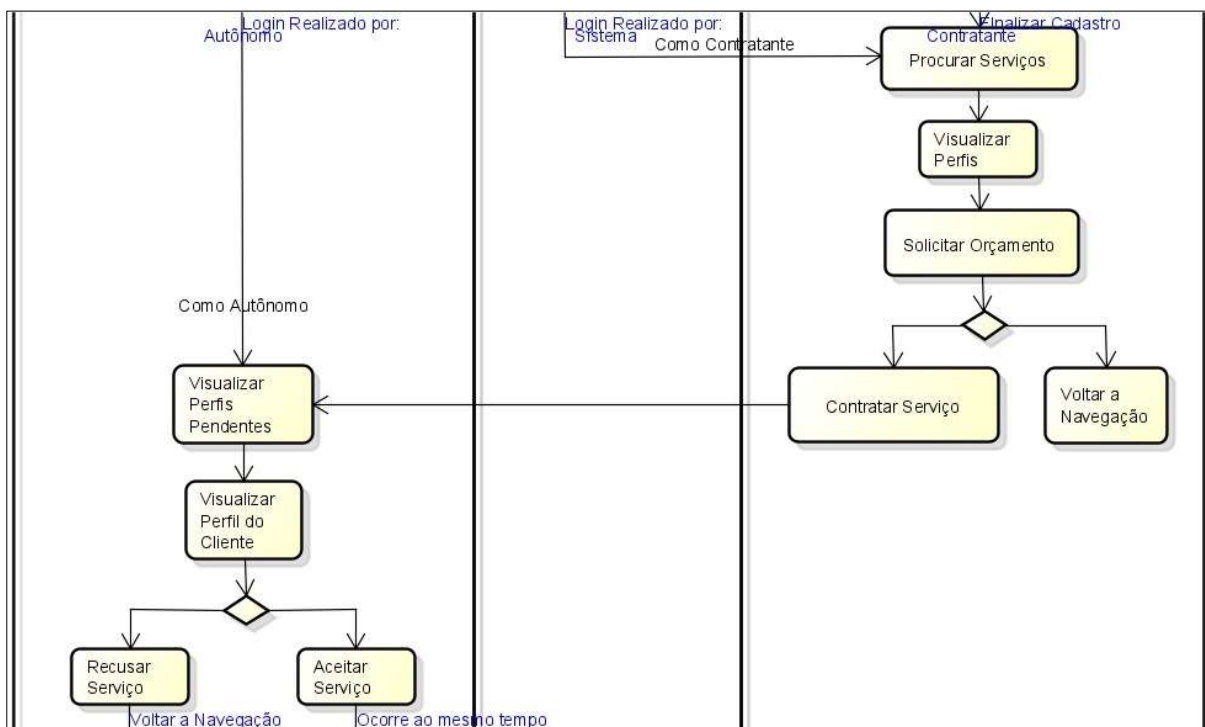
**Figura 62 - Diagrama de Atividade parte 1: Início do Sistema**



Fonte: Autoria própria, 2022.

O fluxo na parte de contratação, mostrado na Figura 63, é iniciado pelo contratante, onde é procurado um serviço e visualizado os perfis dos autônomos, selecionado um que o agrada, solicita-se o orçamento, tendo a opção de voltar a navegação, ou contratar o serviço, no qual é enviado a tela de perfis pendentes, em que o autônomo visualiza os perfis, até selecionar um e recusar e voltar a navegação, ou aceitar o serviço.

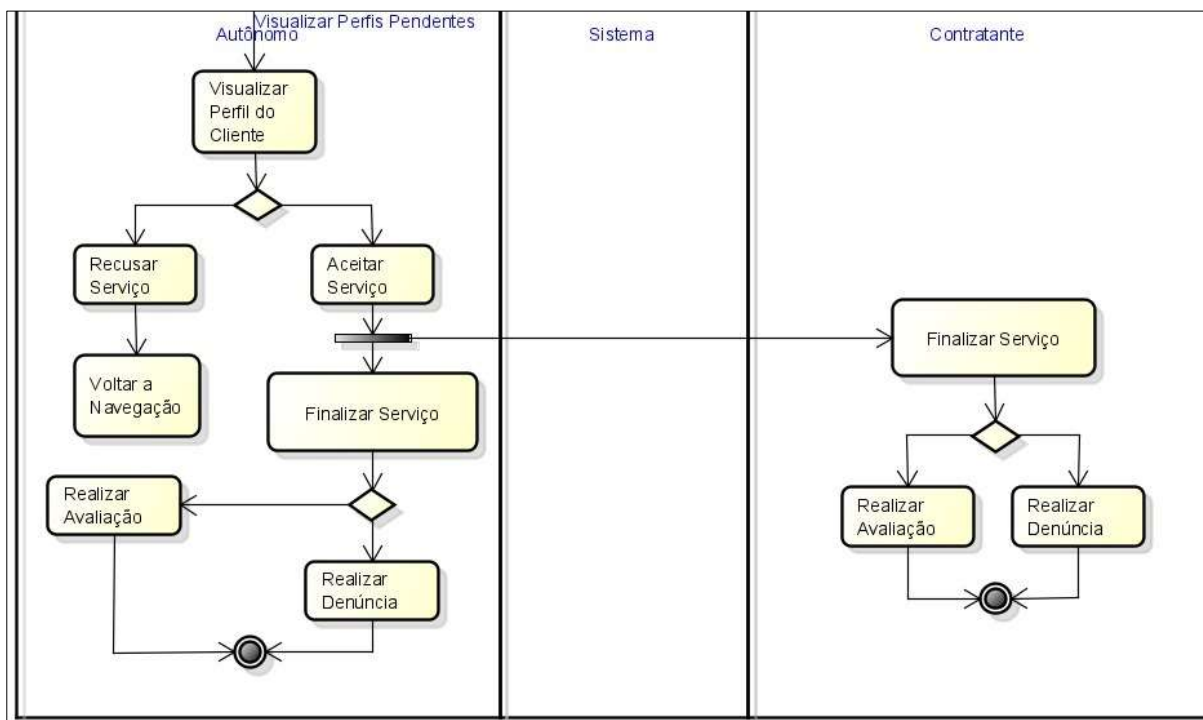
Figura 63 - Diagrama de Atividade parte 2: Contratação



Fonte: Autoria própria, 2022.

O fluxo na parte de finalização do serviço, ilustrado na Figura 64, ocorre ao autônomo aceitar o serviço, onde ele é computado como terminado depois de ambos os usuários finalizarem o serviço, podendo denunciar ou realizar a avaliação, levando ao final do fluxo do sistema.

**Figura 64 - Diagrama de Atividade parte 3: Finalização do Serviço**



Fonte: Autoria própria, 2022.

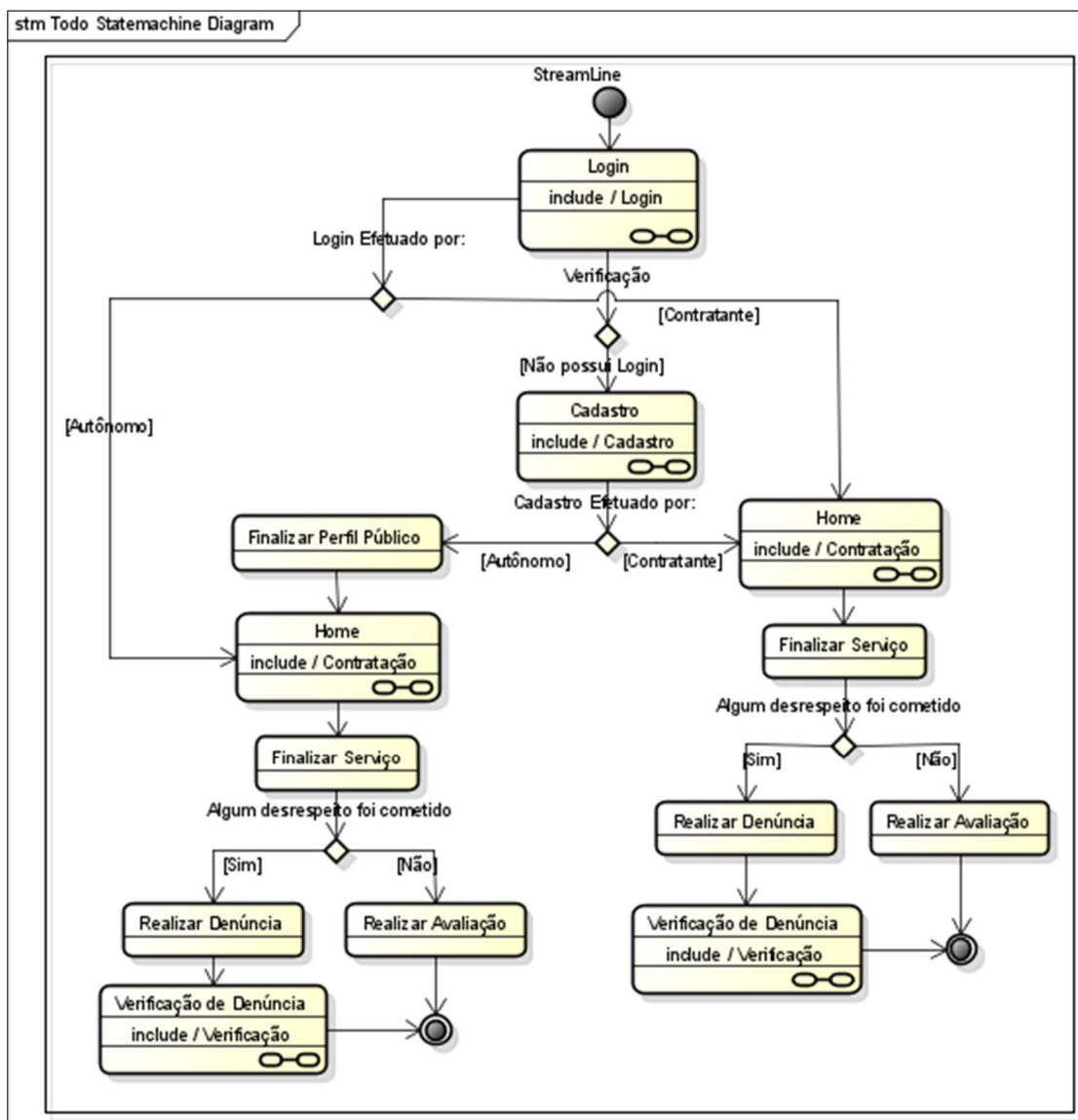
### 3.2.4 Diagrama de Estados

De forma mais detalhada, os diagramas de estados apresentados nas Figuras 65, 66, 67, 68, 69 e 70, demonstram o funcionamento do site, sendo separado pelo diagrama principal, encontrado na Figura 65, e as submáquinas de Login, Cadastro, Home do Autônomo, Home do Contratante e Verificação de Denúncia, que podem ser visualizadas nas Figuras 66, 67, 68, 69 e 70.

É possível observar na Figura 65 que o sistema é iniciado pelo login, onde se verifica se o usuário é um autônomo, no qual será redirecionado para a home própria para os profissionais, um contratante, que será dirigido para a home própria para os clientes, ou se não possui um login, que então será conduzido a se cadastrar, que por sua vez ao ser finalizado também possui a separação, o contratante será redirecionado para a mesma página home, entretanto o autônomo é levado a página de finalização de perfil público, onde terminará de colocar seus dados que apareceram aos clientes, após o autônomo terminar de inserir seus dados e o cliente de navegar pela home, ambos finalizarão seus serviços, onde terão a opção de realizar uma denúncia que encaminhará para a verificação, ou realizar a avaliação, que levará ao fim do programa.



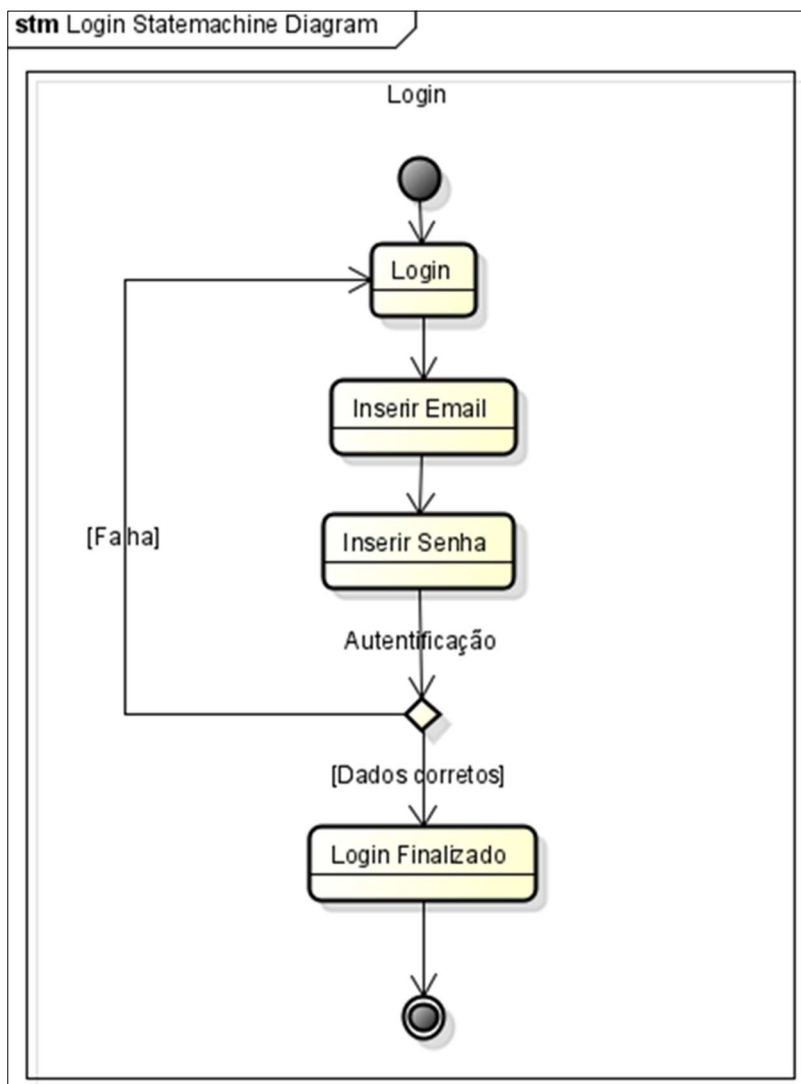
Figura 65 - Diagrama de Estados Principal



Fonte: Autoria própria, 2022.

A submáquina vista na Figura 66 descreve o Login, que se inicia pela própria página de login, onde o usuário irá inserir seu e-mail e senha, para em seguida passar pela autenticação, no qual em caso de falha retornará a página de login, e caso os dados forem inseridos corretamente o login será finalizado, trazendo o final do fluxo.

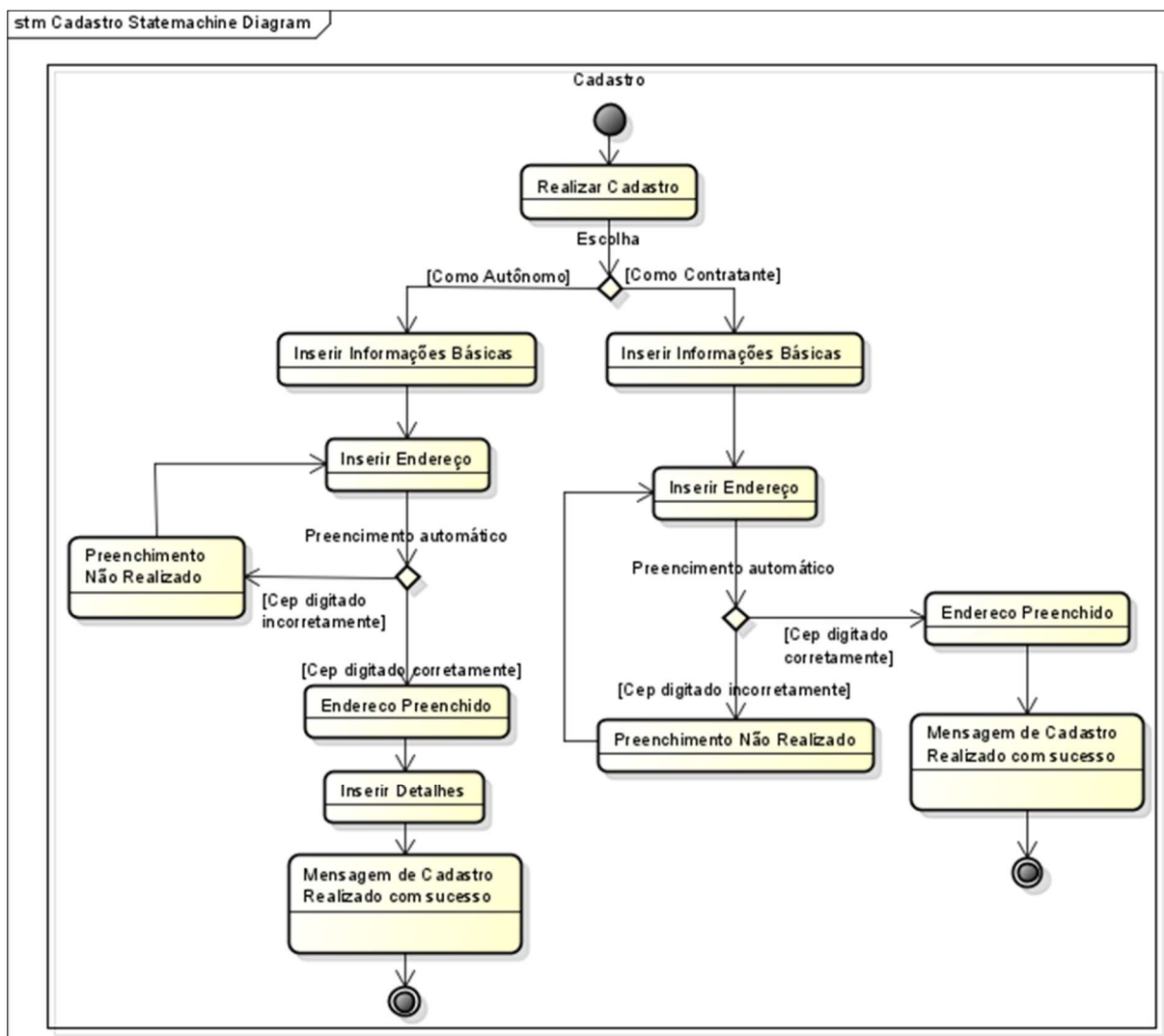
Figura 66 - Diagrama de Estados: Submáquina Login



Fonte: Autoria própria, 2022.

É ilustrado na Figura 67 o cadastro, que se inicia na página de cadastro, no qual é possível escolher se cadastrar como autônomo ou contratante, ambos inserem suas informações básicas, endereços que possuem preenchimento automático com o cep, se preenchido corretamente, caso contrário o preenchimento não é realizado, o forçando a inserir de maneira certa, se o usuário for um cliente logo em seguida de terminar a inserção do endereço aparecerá uma mensagem de sucesso e o fluxo se encerra, se ele for um autônomo será necessário inserir detalhes referente a sua profissão e após aparecerá a mensagem levando ao final do fluxo.

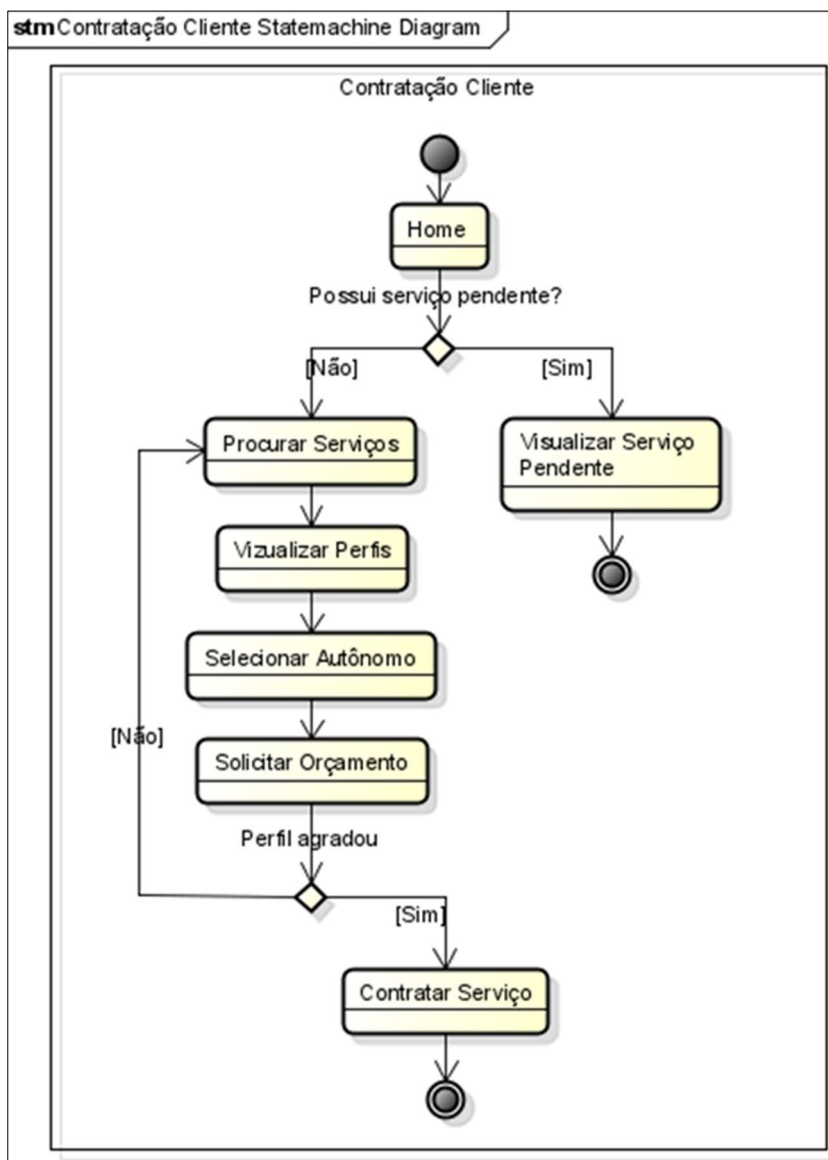
Figura 67 - Diagrama de Estados: Submáquina Cadastro



Fonte: Autoria própria, 2022.

O fluxo de contratação do cliente, mostrado na Figura 68, é iniciado na Home, na hipótese de haver serviço pendente o fluxo é finalizado, caso o cliente não possua ele deve procurar o serviço desejado, onde irá visualizar os perfis e selecionar o que atender as expectativas, solicitando um orçamento o sistema o calculará de acordo com o preço fixo do profissional e as informações fornecidas pelo cliente, caso o preço e o autônomo o agrade ele contratará o serviço o levando para o fim do fluxo, se não ele retornará a procurar serviços.

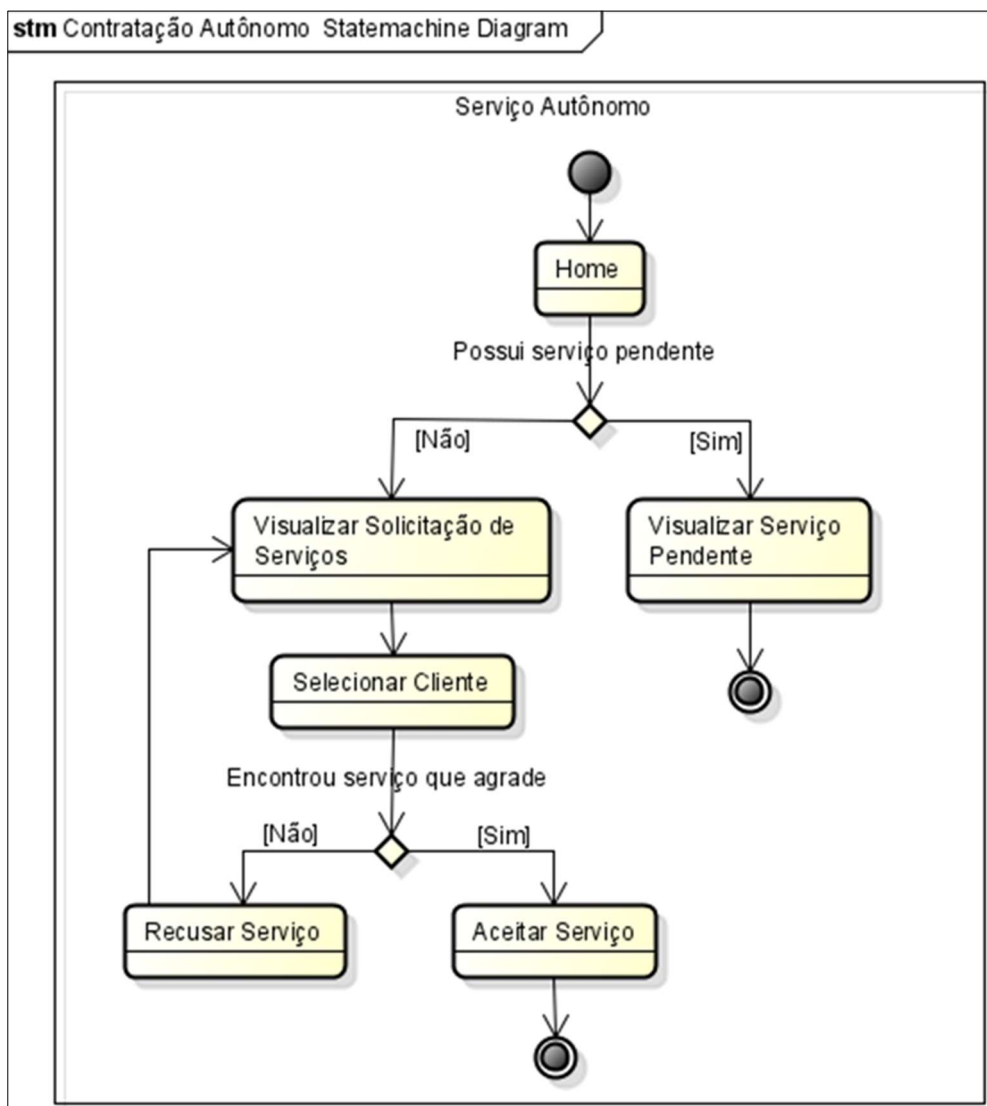
Figura 68 - Diagrama de Estados: Submáquina Contratação Cliente



Fonte: Autoria própria, 2022.

Vemos na Figura 69 a submáquina do serviço do autônomo, que é iniciada na home, onde se verificará se o autônomo possui serviço pendente, na possibilidade de se ter o fluxo é finalizado, se não ele é direcionado a visualização de serviços, onde o profissional irá selecionar um cliente e analisará seu perfil, se o agradar ele aceitará o serviço, levando ao encerramento do fluxo, se não o recusará, o guiando novamente a visualização de solicitações pendentes.

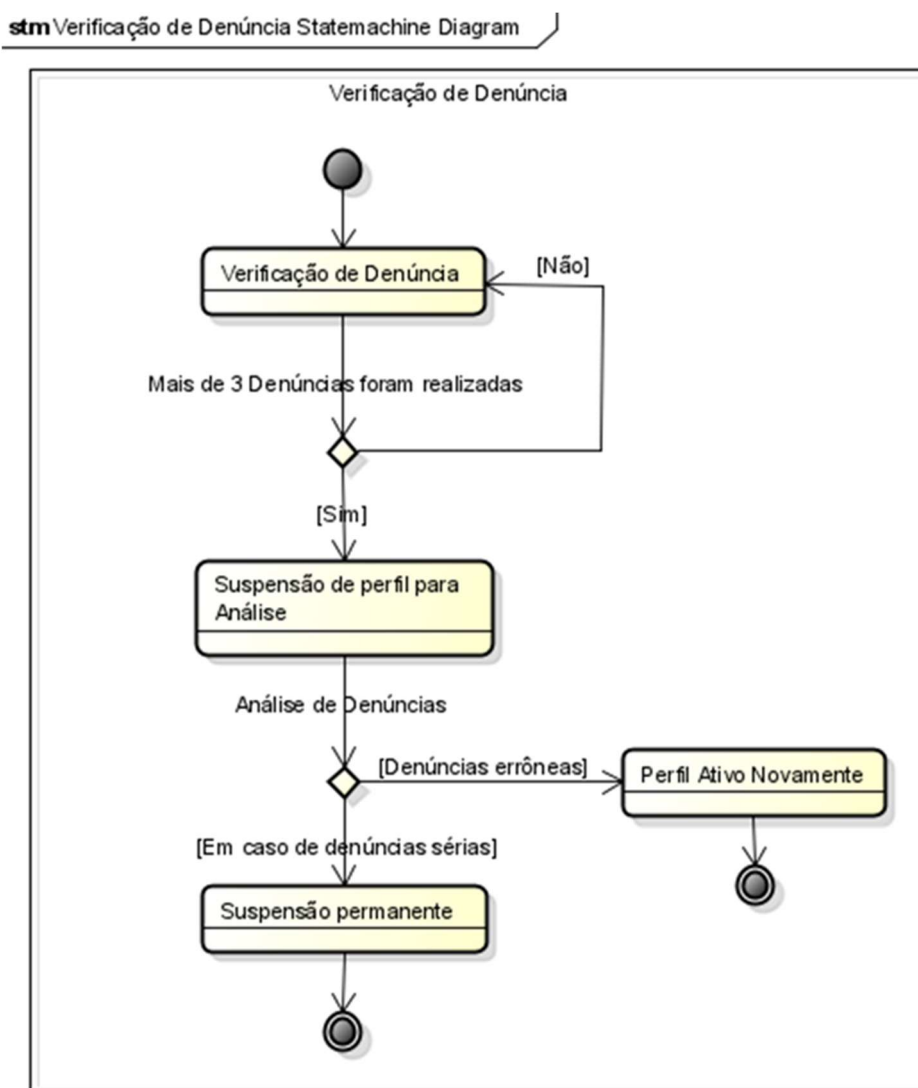
Figura 69 - Diagrama de Estados: Submáquina Serviço Autônomo



Fonte: Autoria própria, 2022.

Na Figura 70 temos a submáquina de verificação de denúncia, que se inicia pela própria verificação, que em caso de mais de 3 denúncias no perfil do usuário terá uma suspensão, que passará por uma análise, se elas forem condizentes o perfil será suspenso permanentemente, caso errôneas o perfil será ativo, caso possua menos de 3 denúncias ele retornará para a verificação.

Figura 70 - Diagrama de Estados: Submáquina Verificação Denúncia



Fonte: Autoria própria, 2022.

### 3.3 Logo StreamLine

Para a caracterização do projeto StreamLine, criamos uma mascote como logo, chamada Ely, representando as buscas por serviços e contratações, com o símbolo de localização a cima de sua cabeça, assim como exibido na Figura 71. As cores de Ely foram selecionadas de acordo com a paleta de cores decidida para o design do site StreamLine.

Figura 71 - Logo StreamLine: Ely Neutra



Fonte: Autoria própria, 2022.

O nome Streamline significa linha de fluxo, a qual expressa resistência com as dificuldades que tentam regredir o avanço de um indivíduo. A partir desse conceito, o nome Ely de nossa mascote, surge de uma transformação do próprio nome

StreamLine, como um apelido curto com a mesma entonação. Na Figura 72 podemos observar Ely com expressões de surpresa.

**Figura 72 - Logo StreamLine: Ely Surpresa**

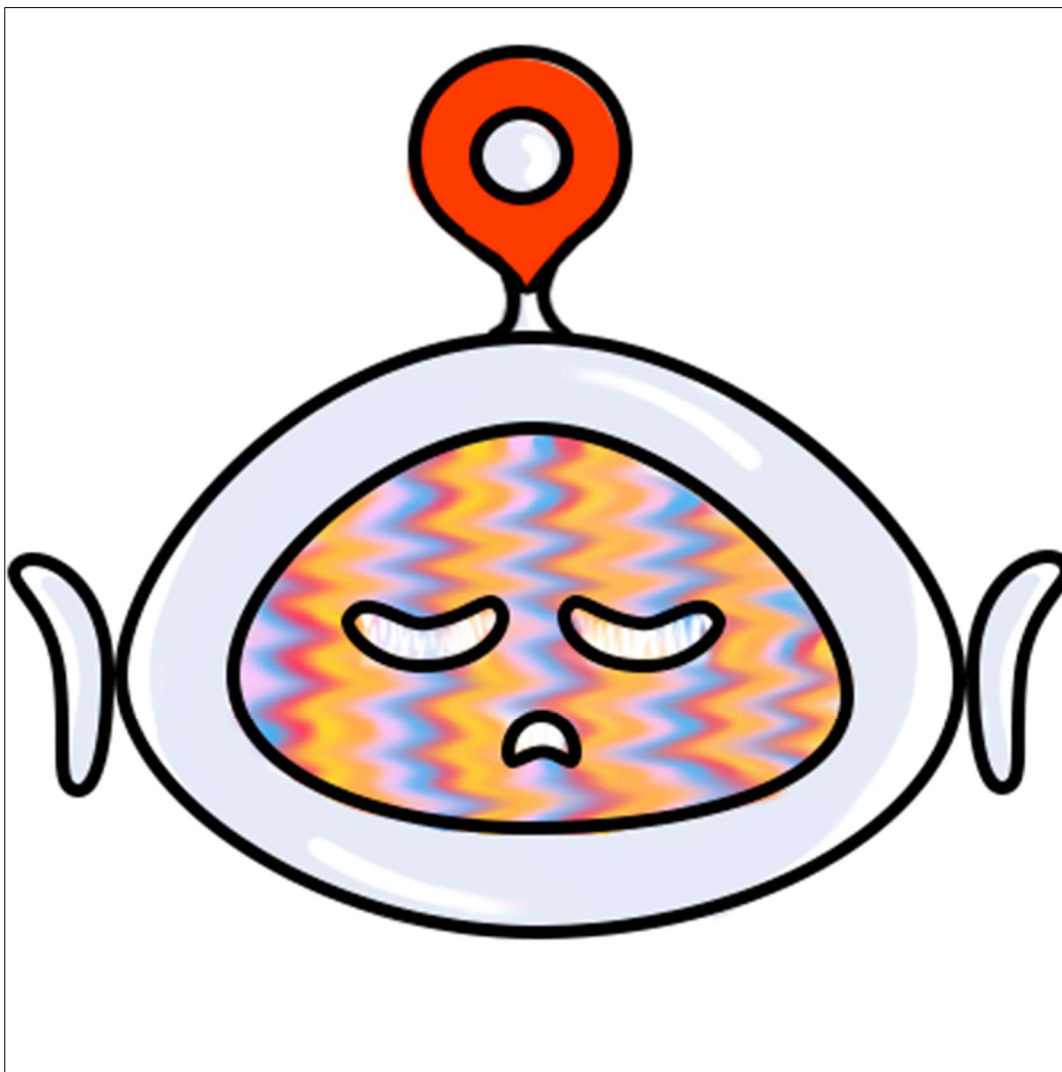


Fonte: Autoria própria, 2022.



Na Figura 73, Ely é apresentada com expressões de aborrecimento, encontrando-se visualmente em nosso site quando o profissional autônomo não dispor pendência de serviços contratados.

**Figura 73 - Logo StreamLine: Ely Aborrecida**



Fonte: Autoria própria, 2022.

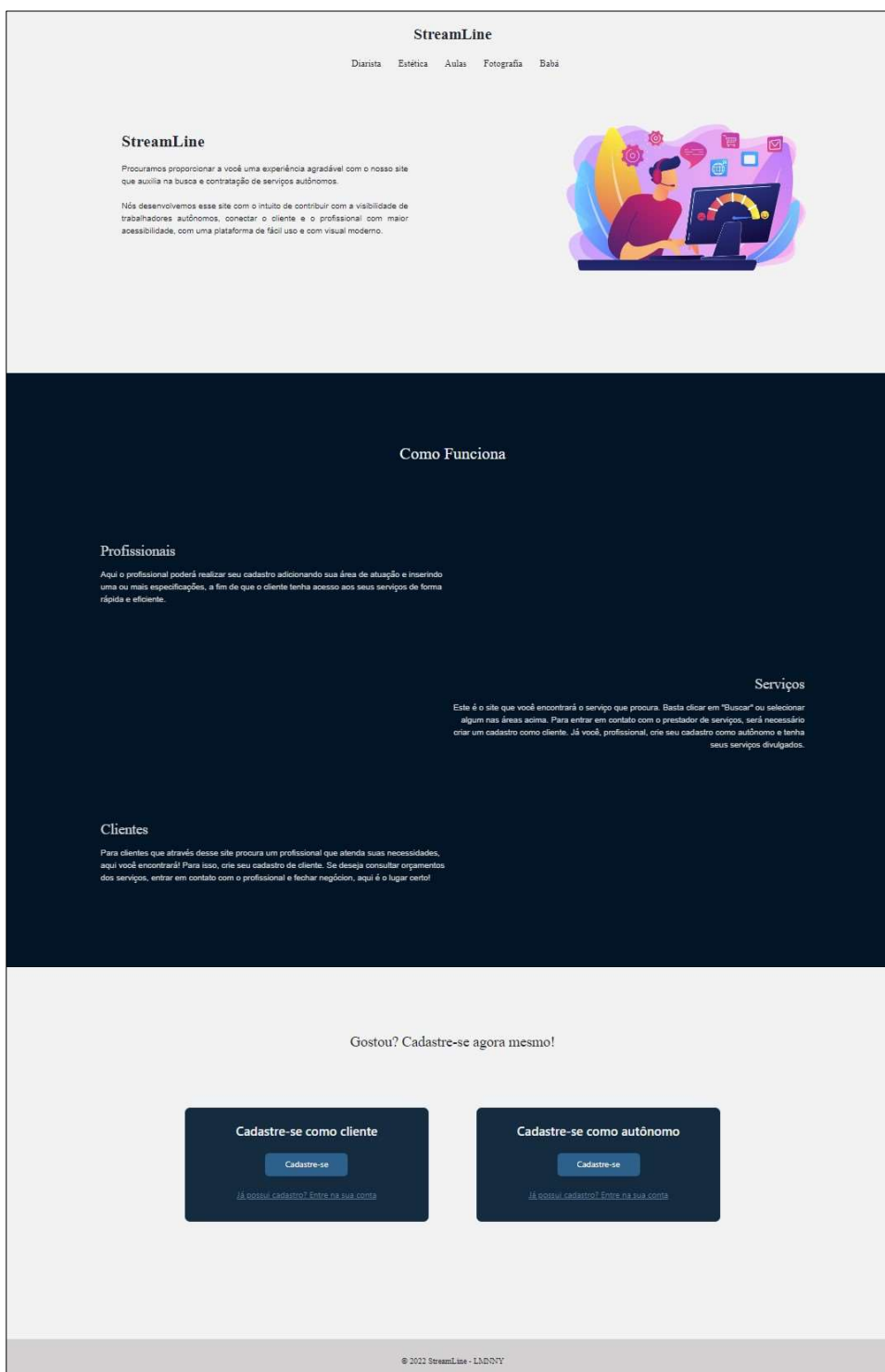
### **3.4 Telas do Site StreamLine**

O site StreamLine é composto por telas que colaboram com o funcionamento de nosso projeto, essas telas contém informações e funcionalidades que auxiliam o usuário na necessidade de busca do mesmo, seja por um profissional autônomo ou um contratante. Nossas telas são divididas em página inicial, página cadastro, página login, página perfil e página trabalhos.

### 3.4.1 Página Inicial

Na página inicial do site StreamLine, o usuário possui acesso a informações sobre nosso projeto, como o intuito, conceito de como funciona e opções de cadastro ou login ao fim da página, como demonstrado na Figura 74:

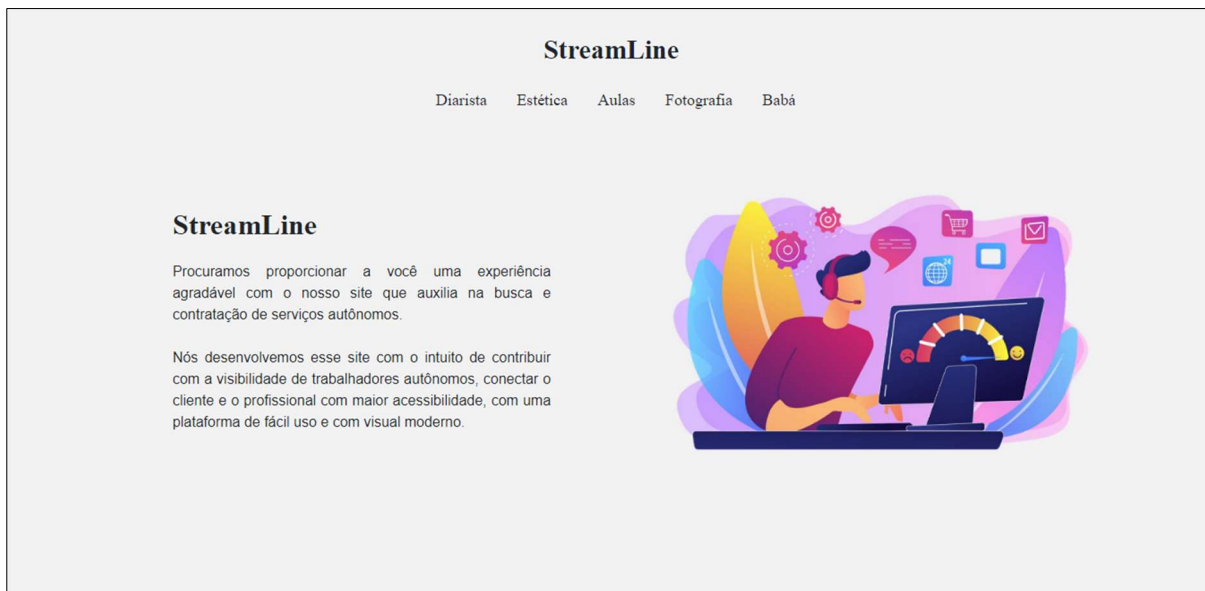
**Figura 74 - Página Inicial Completa StreamLine**



Fonte: Autoria própria, 2022.

Para melhor visualização separamos essa tela inicial em 3 partes, a primeira parte informa o usuário sobre o intuito de nosso site, como na Figura X:

**Figura 75 - Página Inicial: Intuito**



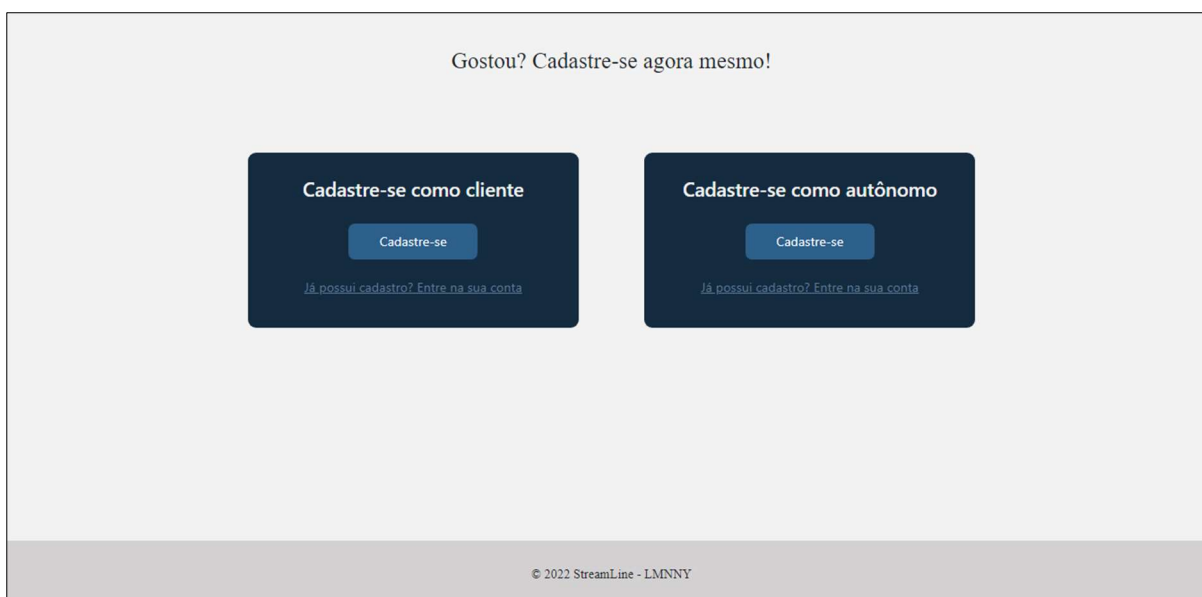
Fonte: Autoria própria, 2022.

A segunda parte da página inicial, como apresentada na Figura 76, auxilia o usuário a entender como funciona nosso sistema de busca e contratação dos serviços autônomos.

**Figura 76 - Página Inicial: Funcionamento**

Fonte: Autoria própria, 2022.

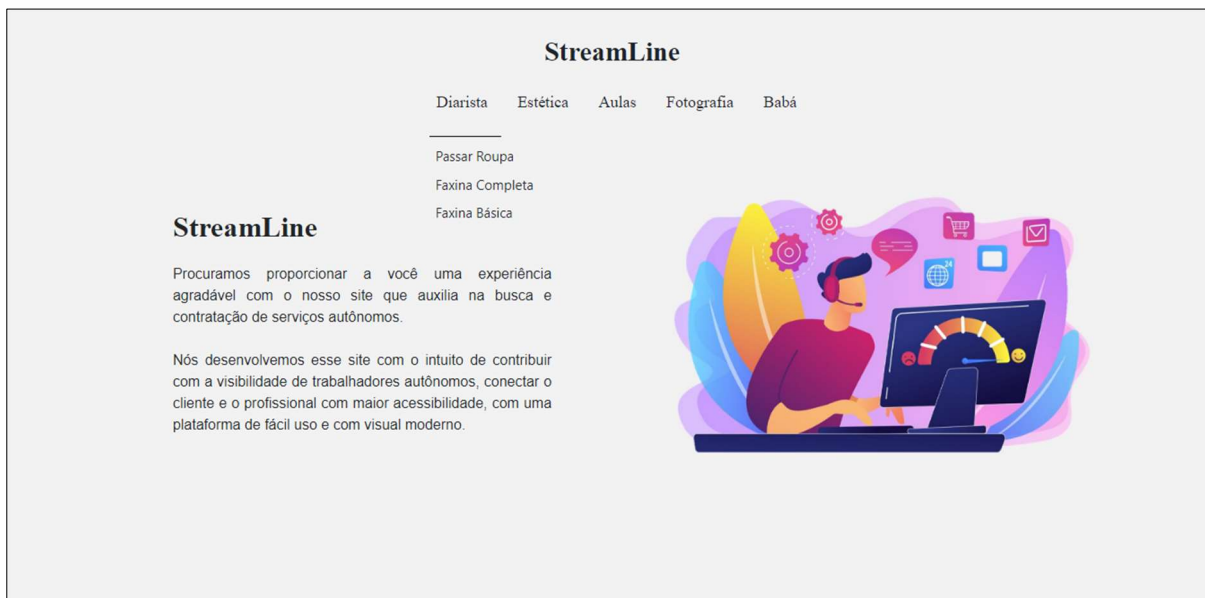
Já a terceira e última parte da página inicial, interage com o usuário permitindo que o mesmo escolha entre se cadastrar ou logar, como cliente ou profissional autônomo, como é exibido na Figura 77:

**Figura 77 - Página Inicial: Cadastro e Login**

Fonte: Autoria própria, 2022.

Ainda na página inicial, temos categorias de trabalho, em que o usuário pode selecionar para especificar sua busca pelo serviço autônomo, como a categoria diaristas, com serviços domésticos evidenciado na Figura 78:

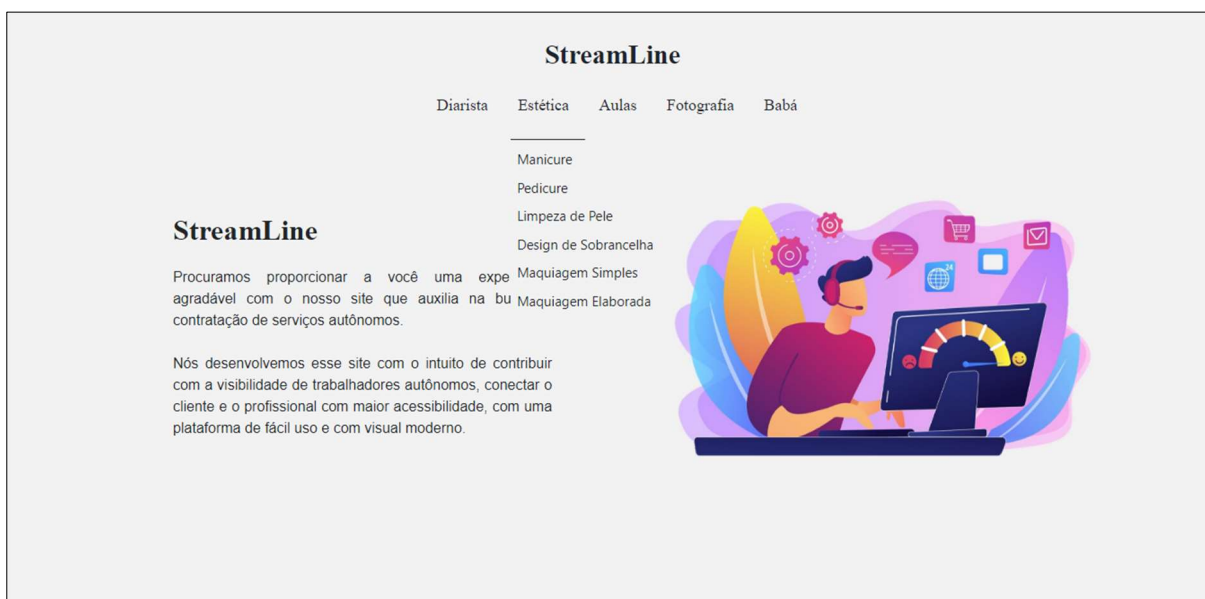
**Figura 78 - Página Inicial: Categoria Diarista**



Fonte: Autoria própria, 2022.

Na categoria estéticas, temos serviços autônomos relacionados a área da beleza e cuidados, assim como apresentado na Figura 79:

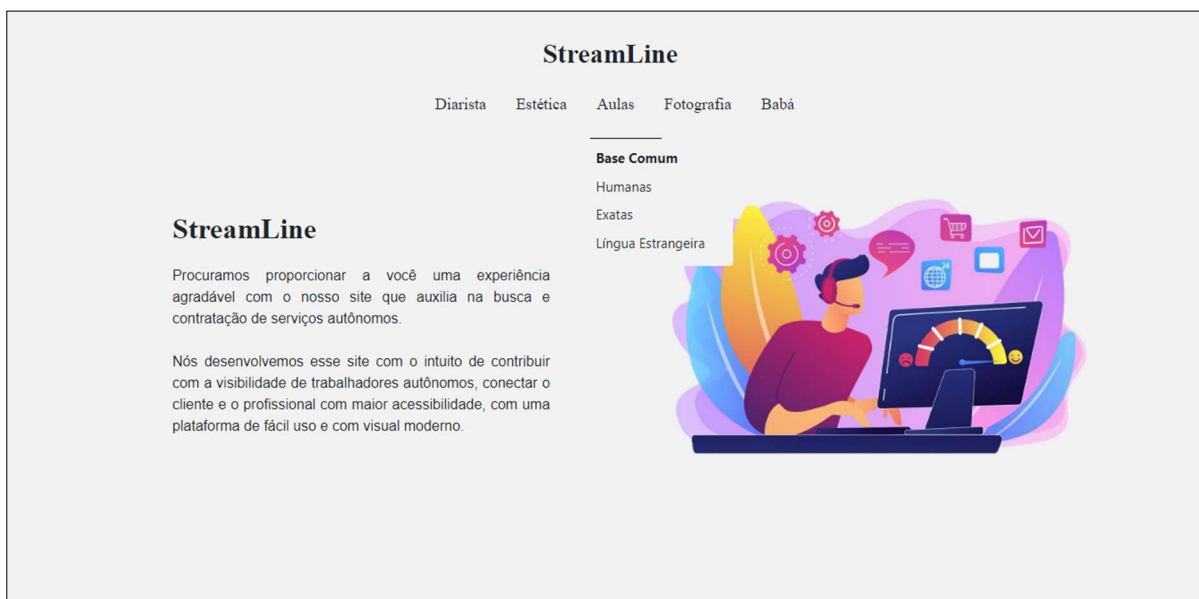
**Figura 79 - Página Inicial: Categoria Estética**



Fonte: Autoria própria, 2022.

A categoria de serviços autônomos aulas, é pertinente a área de ensino e educação como exibido na Figura 80:

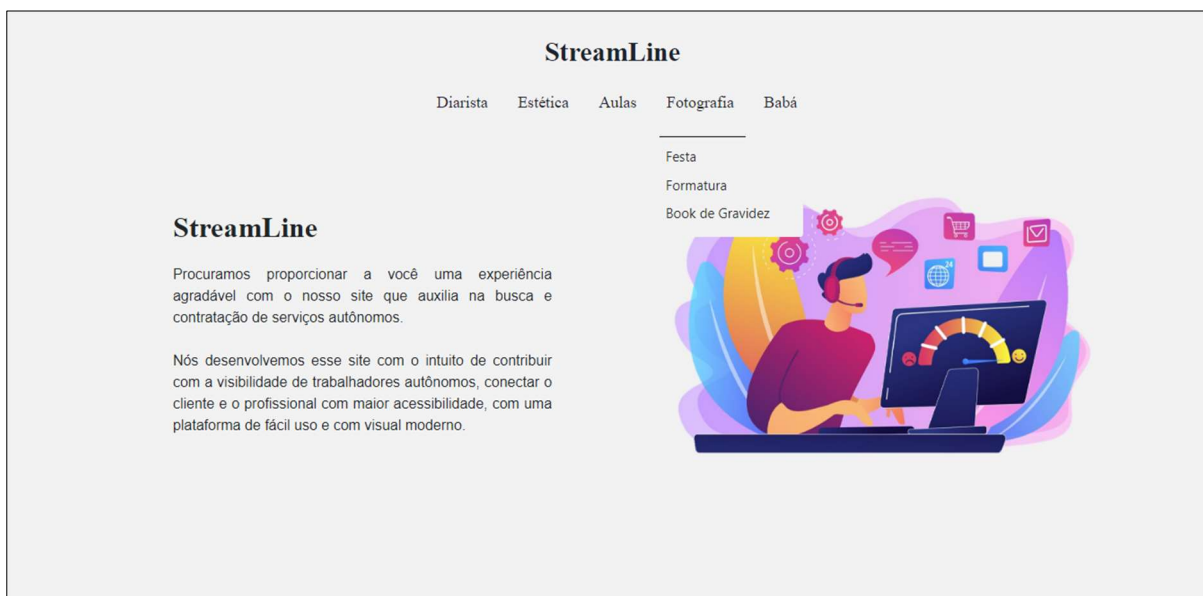
**Figura 80 - Página Inicial: Categoria Aulas**



Fonte: Autoria própria, 2022.

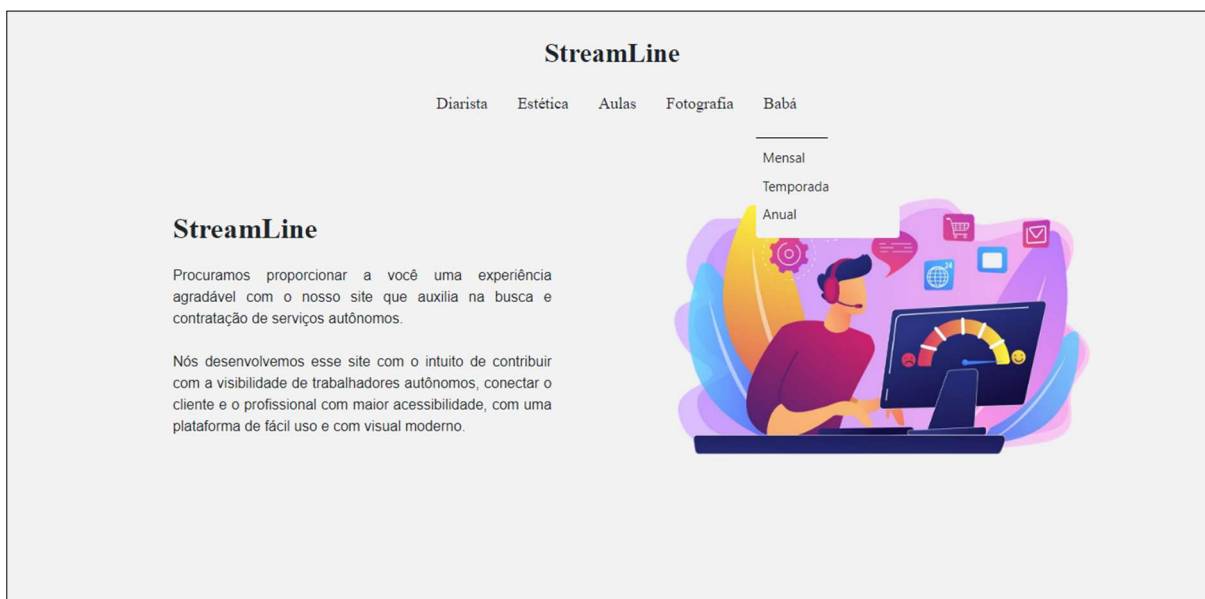
Já na categoria fotografia, como mostrado na Figura 81, os serviços autônomos são pertencentes as áreas comerciais, artísticos e jornalísticos.

**Figura 81 - Página Inicial: Categoria Fotografia**



Fonte: Autoria própria, 2022.

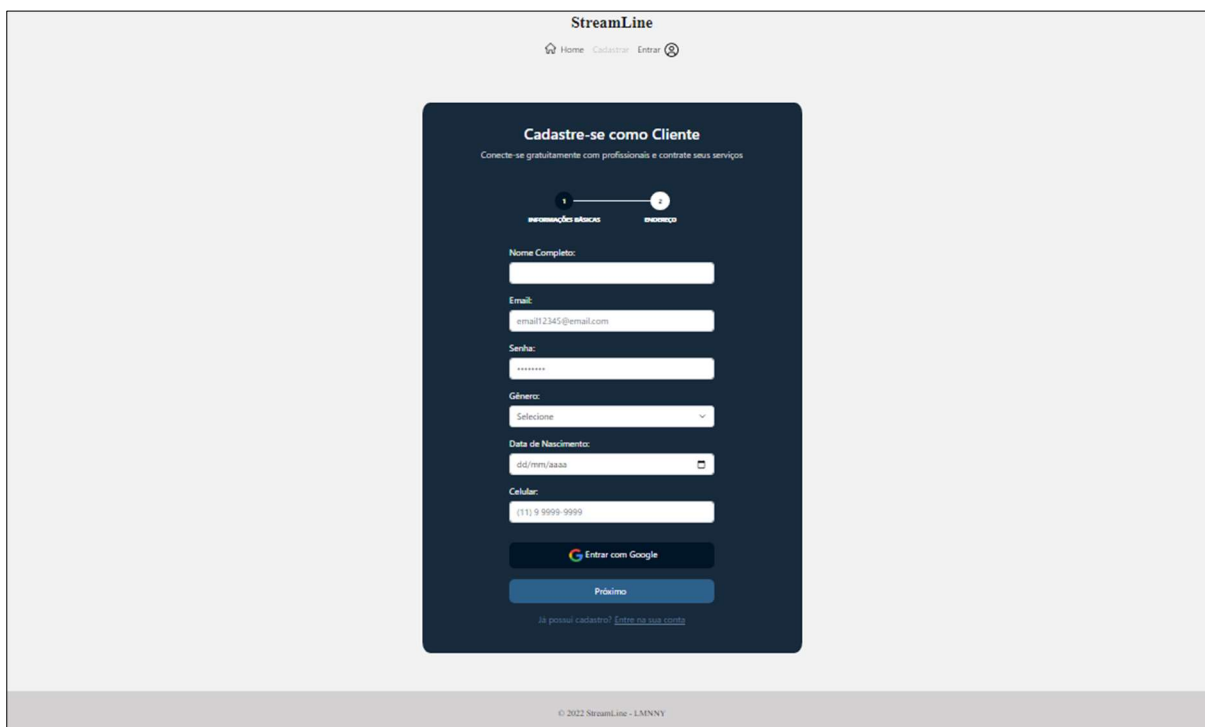
A última categoria de serviços autônomos é a categoria babá, como evidenciada na Figura 82, essa categoria é associada com a área de cuidados infantil, em um período de tempo específico.

**Figura 82 - Página Inicial: Categoria Babá**

Fonte: Autoria própria, 2022.

### 3.4.2 Cadastros

Para realizar o cadastro como cliente, o usuário é direcionado a primeira etapa de cadastro, como mostrado na Figura 83, onde preencherá suas informações básicas, nas quais são: nome completo, e-mail, senha, gênero, data de nascimento e seu celular, tendo a possibilidade de se cadastrar com o Google, ou em caso de já possuir cadastro, efetuar o *login*. Vale ressaltar que para o usuário se cadastrar, será obrigatório ser maior de 18 anos.

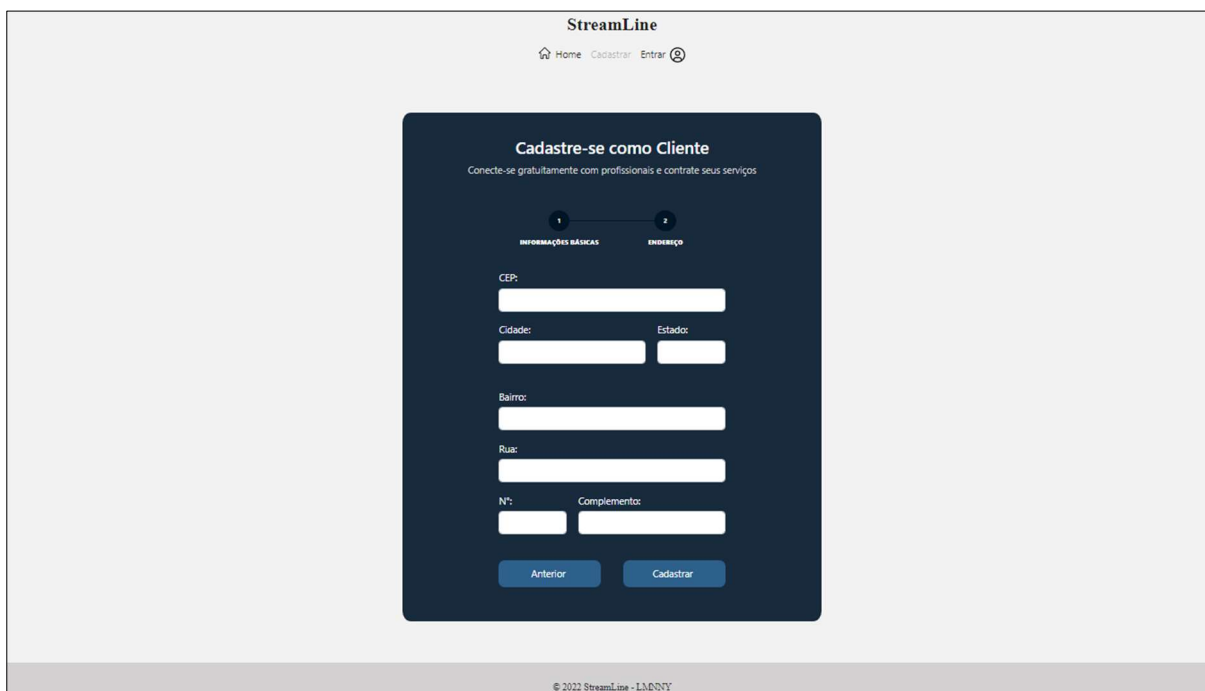
**Figura 83 - Cadastro como Cliente: Informações Básicas**

The image shows a web registration form for 'StreamLine'. At the top, there is a navigation bar with 'Home', 'Cadastrar', and 'Entrar' links. The main heading is 'Cadastre-se como Cliente' with a subtext 'Conecte-se gratuitamente com profissionais e contrate seus serviços'. A progress indicator shows two steps: '1. informações básicas' (active) and '2. endereço'. The form fields include: 'Nome Completo:' (text input), 'Email:' (text input with 'email12345@email.com'), 'Senha:' (password input with masked characters), 'Gênero:' (dropdown menu with 'Selecione'), 'Data de Nascimento:' (calendar icon and 'dd/mm/aaaa' format), and 'Celular:' (text input with '(11) 9 9999-9999'). Below the fields are buttons for 'Entrar com Google' and 'Próximo'. A footer link says 'já possui cadastro? [Entre na sua conta](#)'. The footer text is '© 2022 StreamLine - LMNNY'.

Fonte: Autoria própria, 2022.

Ao clicar em próximo, o usuário será encaminhado a segunda etapa, a de endereço, como demonstra a Figura 84, em que o cliente deverá informar seu CEP, para que desta forma, os demais campos de cidade, estado, bairro e rua, sejam preenchidos automaticamente, em caso de cep incorreto os campos não serão preenchidos, e eles não possuem forma de serem colocados manualmente. Depois deverá ser informado o número da residência e seu complemento. Tendo a opção de cadastrar e voltar a página anterior.



**Figura 84 - Cadastro como Cliente: Endereço**

The image shows a web interface for StreamLine. At the top, the logo 'StreamLine' is centered, with navigation links for 'Home', 'Cadastrar', and 'Entrar'. Below this is a dark blue registration form titled 'Cadastre-se como Cliente'. The form includes a progress indicator with two steps: 'INFORMAÇÕES BÁSICAS' (completed) and 'ENDEREÇO' (current step). The form fields are: CEP (with a text input), Cidade (with a text input) and Estado (with a dropdown menu), Bairro (with a text input), Rua (with a text input), N° (with a text input) and Complemento (with a text input). At the bottom of the form are two buttons: 'Anterior' and 'Cadastrar'. The footer of the page contains the copyright notice '© 2022 StreamLine - LMDNY'.

Fonte: Autoria própria, 2022.

A realização do cadastrado como autônomo, possui a primeira etapa de informações básicas, como ilustrado na Figura 85, onde será necessário o preenchimento do nome completo, e-mail, senha, gênero e data de nascimento, onde para o usuário se cadastrar, será obrigatório ser maior de 18 anos. Possuindo a possibilidade de se cadastrar com o Google, ou na hipótese de já ter cadastro, entrar em sua conta.

**Figura 85 - Cadastro como Autônomo: Informações Básicas**

The image shows a web interface for StreamLine. At the top, there is a navigation bar with 'Home', 'Cadastrar', and 'Entrar' (with a user icon). The main content area features a dark blue card titled 'Cadastre-se como Autônomo' with the subtitle 'Conecte-se gratuitamente com clientes que contratarão seus serviços'. A progress indicator shows three steps: '1. INFORMAÇÕES BÁSICAS' (active), '2. ENDEREÇO', and '3. DETALHES'. The form fields include: 'Nome Completo' (text input), 'Email' (text input with 'email12345@email.com'), 'Senha' (password input with masked characters), 'Gênero' (dropdown menu with 'Selecione'), and 'Data de Nascimento' (date picker with 'dd/mm/aaaa'). Below the form are two buttons: 'Entrar com Google' and 'Próximo'. At the bottom of the card, there is a link: 'Já possui cadastro? [Entre na sua conta](#)'. The footer of the page contains the copyright notice '© 2022 StreamLine - LMNNY'.

Fonte: Autoria própria, 2022.

Ao avançar, o autônomo será redirecionado a etapa de endereço, como mostra a Figura 86, onde será informado o CEP, que assim como no cadastro de cliente, o campo de cidade, estado, bairro e rua, serão preenchidos automaticamente, e em caso do cep ser informado incorretamente, os campos não serão preenchidos, e eles não podem ser colocados manualmente. Em seguida deverá ser informado o número da residência e seu complemento. Tendo a opção de avançar ou voltar a página anterior.

**Figura 86 - Cadastro como Autônomo: Endereço**

The image shows a web form titled "Cadastro como Autônomo" on the StreamLine website. The form is divided into three steps: 1. INFORMAÇÕES BÁSICAS, 2. ENDEREÇO, and 3. DETALHES. The current step is "ENDEREÇO", which is highlighted with a blue circle and the number 2. The form contains the following fields: CEP (a single-line text input), Cidade (a single-line text input), Estado (a single-line text input), Bairro (a single-line text input), Rua (a single-line text input), Nº (a single-line text input), and Complemento (a single-line text input). At the bottom of the form, there are two buttons: "Anterior" and "Próximo". The StreamLine logo is at the top, and the copyright notice "© 2022 StreamLine - LMNNY" is at the bottom.

Fonte: Autoria própria, 2022.

Avançando para a última etapa, como pode ser observado na Figura 87, temos os detalhes, que são as informações referentes ao trabalho do autônomo, onde será informado o telefone, área de atuação, em que o autônomo escolherá conforme as opções predeterminadas pelo site, as especificações, também predeterminadas, o preço base, tempo de experiência e se o número informado se refere a anos, meses ou dias. É necessário para a finalização, selecionar a caixa de seleção, que representa a concordância do usuário com os termos de política e privacidade do site, possuindo por fim, os botões de finalização ou para retornar a etapa anterior.

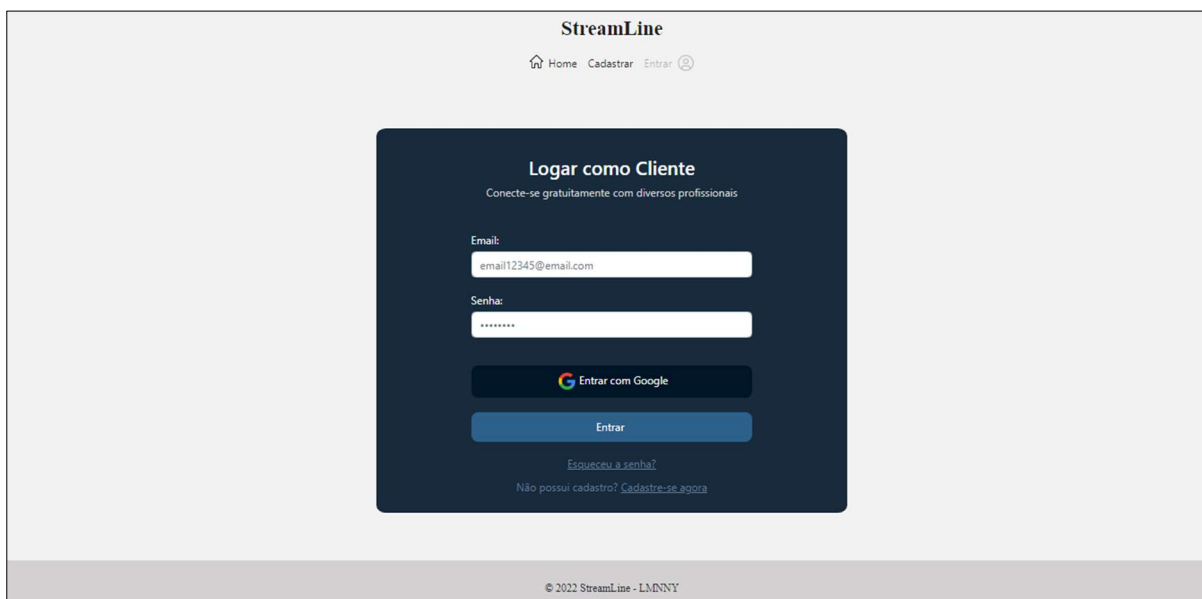
Figura 87 - Cadastro como Autônomo: Detalhes

The image shows a web interface for 'StreamLine' with a navigation bar containing 'Home', 'Cadastrar', and 'Entrar' with a user icon. The main content is a dark blue registration form titled 'Cadastre-se como Autônomo' with the subtitle 'Conecte-se gratuitamente com clientes que contratarão seus serviços'. A progress indicator shows three steps: 1. INFORMAÇÕES BÁSICAS, 2. ENDEREÇO, and 3. DETALHES (the current step). The form fields include: 'Telefone' with a pre-filled value '(11) 9 9999-9999'; 'Área de Atuação:' with a dropdown menu set to 'Diarista'; 'Selecionar Especificações:' with an empty text input; 'Preço base:' with a pre-filled value '0,00' and a note 'Números quebrados devem ser separados por ponto (.)'; 'Tempo de Experiência:' with an empty text input and a 'Ano(s)' label; a checkbox for 'Concordo com os termos e política de privacidade.'; and two buttons: 'Anterior' and 'Cadastrar'. At the bottom, there is a link: 'Já possui cadastro? [Entre na sua conta](#)'.

Fonte: Autoria própria, 2022.

### 3.4.3 Login

Para efetuar o *login* o usuário deverá informar seu e-mail e senha, possuindo a possibilidade de realizar o *login* com o Google. Tendo onde clicar em caso de esquecimento de senha e em caso de não possuir cadastro. Sendo dividido entre cliente e autônomo, a Figura 88 ilustra o *login* como cliente.

**Figura 88 - Login Cliente**

StreamLine

Home Cadastrar Entrar

### Logar como Cliente

Conecte-se gratuitamente com diversos profissionais

Email:

Senha:

[Entrar com Google](#)

[Entrar](#)

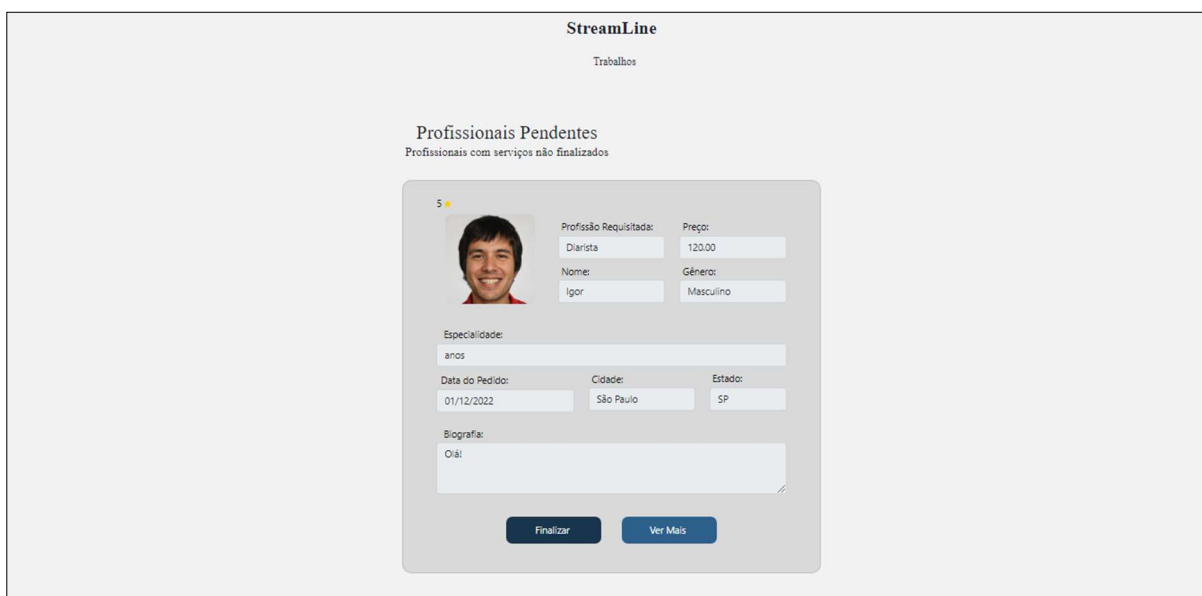
[Esqueceu a senha?](#)

[Não possui cadastro? Cadastre-se agora](#)

© 2022 StreamLine - LMNNY

Fonte: Autoria própria, 2022.

Após realizar o *login* como usuário cliente, terá acesso a página principal do cliente, nela é possível visualizar no início os perfis de profissionais pendentes, ou seja, perfis de profissionais que o serviço ainda não foi finalizado pelo cliente, como visualizado na Figura 89. Se as pendências de serviços não forem finalizadas pelo cliente, não é possível que o mesmo tente contratar outro profissional em seguida.

**Figura 89 - Página Principal Cliente: Profissionais Pendentes**


StreamLine

Trabalhos

### Profissionais Pendentes

Profissionais com serviços não finalizados

5



Profissão Requisitada:  Preço:

Nome:  Gênero:

Especialidade:

Data do Pedido:  Cidade:  Estado:

Biografia:

[Finalizar](#) [Ver Mais](#)

Fonte: Autoria própria, 2022.

Em seguida, temos ao meio da página, os perfis de profissionais que estão disponíveis para oferecer seus serviços, identificado como Principais Autônomos, como apresentado na Figura 90:

**Figura 90 - Página Principal Cliente: Principais Autônomos**

Principais Autônomos

Ver Mais

0 ★

Profissão: Diarista Preço: 120,00

Nome: Igor Gênero: Masculino

Especialidade: anos

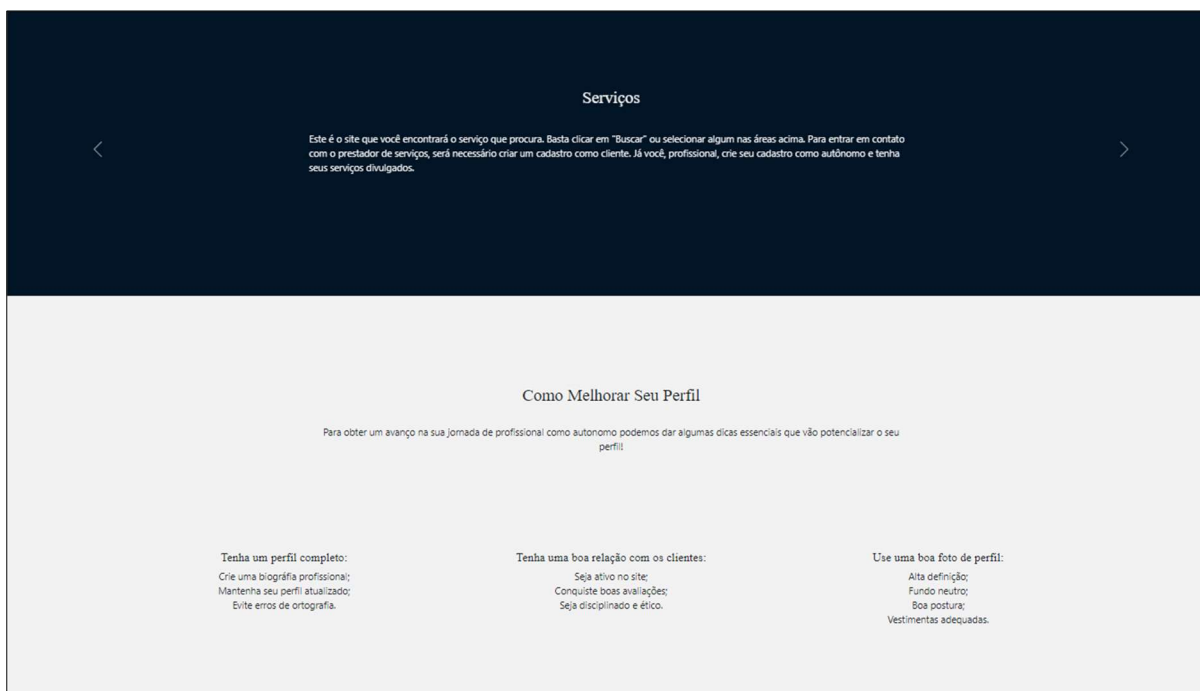
Cidade: São Paulo Estado: SP

Biografia: Olá!

Descartar Ver Mais

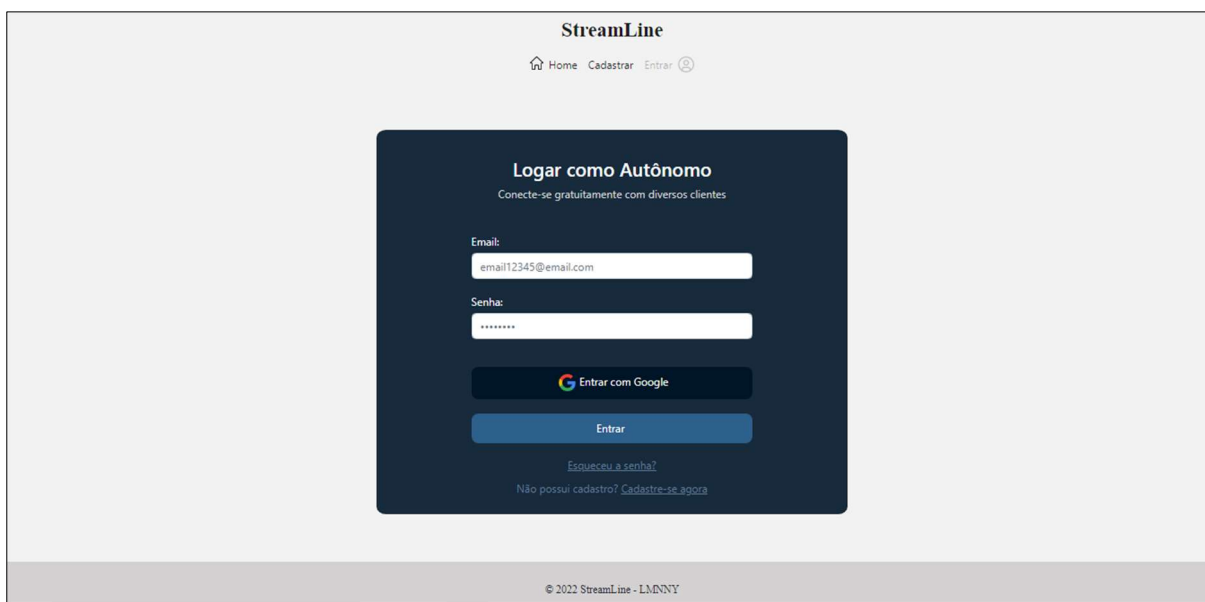
Fonte: Autoria própria, 2022.

Por fim, na Figura 91, é obtemos informações sobre o funcionamento do site e instruções de como o usuário pode melhorar seu perfil.

**Figura 91 - Página Principal Cliente: Funcionamento e Instruções**

Fonte: Autoria própria, 2022.

A Figura 92 mostra o *login* como autônomo, que possui as mesmas funcionalidades que o *Login* de cliente ilustrado na Figura 88. Sendo necessário o preenchimento do e-mail e senha.

**Figura 92 - Login Autônomo**

Fonte: Autoria própria, 2022.

Após realizar o *login* como usuário autônomo, terá acesso a página principal do autônomo, sendo possível visualizar no início, os perfis de clientes pendentes, ou seja, perfis de clientes no qual o serviço ainda não foi finalizado pelo autônomo, como demonstrado na Figura 93. Se as pendências de serviços não forem finalizadas pelo profissional, não é possível que o mesmo tente prestar outro serviço, em seguida.

**Figura 93 - Página Principal Autônomo: Clientes Pendentes**


StreamLine

Trabalhos

### Clientes Pendentes

Clientes com serviço não finalizado

3 ★



Profissão Requisitada:	Preço:
Diarista	13.00
Nome:	Gênero:
Roger Guedes	Masculino

Especialidade:

anos

Data do Pedido:	Cidade:	Estado:
01/12/2022	43	SP

Biografia:

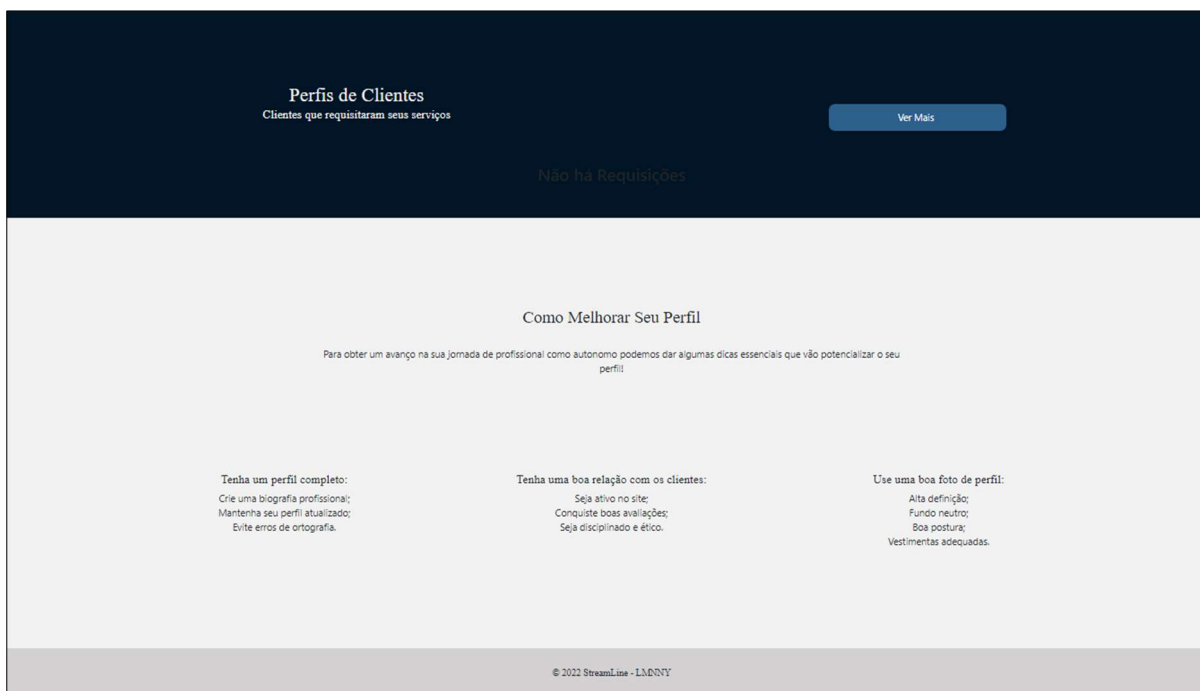
Olá!

Finalizar Ver Mais

Fonte: Autoria própria, 2022.

Em continuidade, na Figura 94 é ilustrado os perfis de clientes que requisitaram os serviços do usuário autônomo, identificado como Perfis de Clientes, e ao final da página obtemos instruções de como o usuário pode melhorar seu perfil.

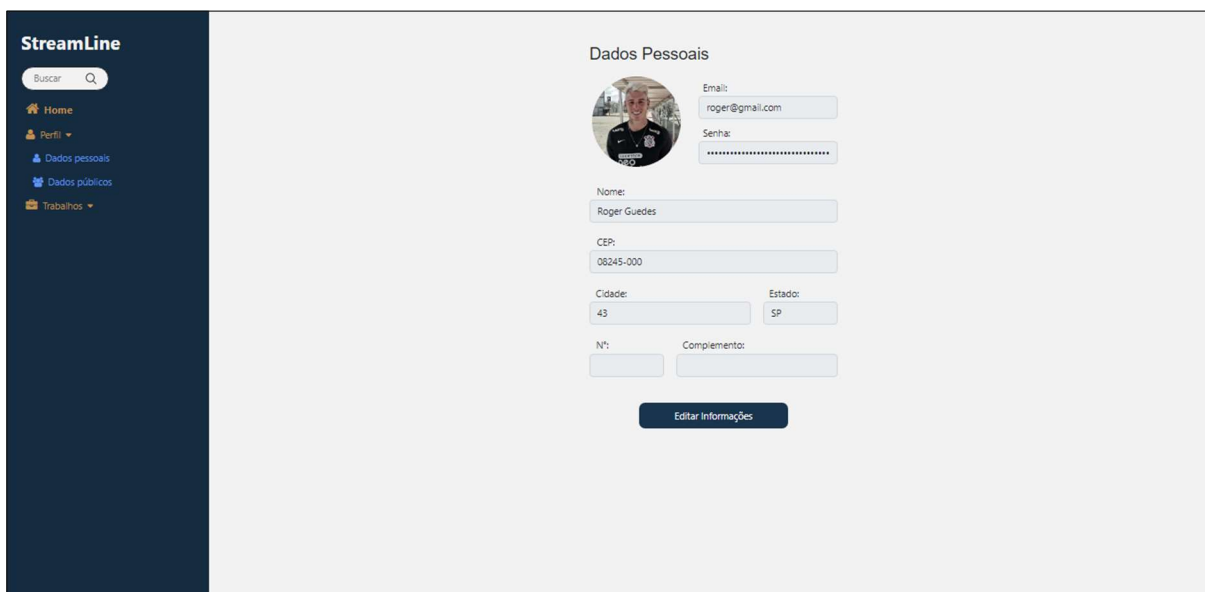


**Figura 94 - Página Principal Autônomo: Perfis de Clientes e Instruções**

Fonte: Autoria própria, 2022.

### 3.4.4 Perfil

As configurações do perfil são divididas entre dados pessoais, que se referem aos dados privados do usuário, que não serão mostrados para as demais pessoas, e os dados públicos, que aparecerão para os outros usuários, ambas possuindo botão para a alteração de dados. Caso o usuário seja um cliente, a Figura 95 ilustra a tela de seus dados pessoais, que se referem a foto, e-mail, senha, nome, cep, cidade, estado, número da residência e o complemento.

**Figura 95 - Perfil Cliente: Dados Pessoais**

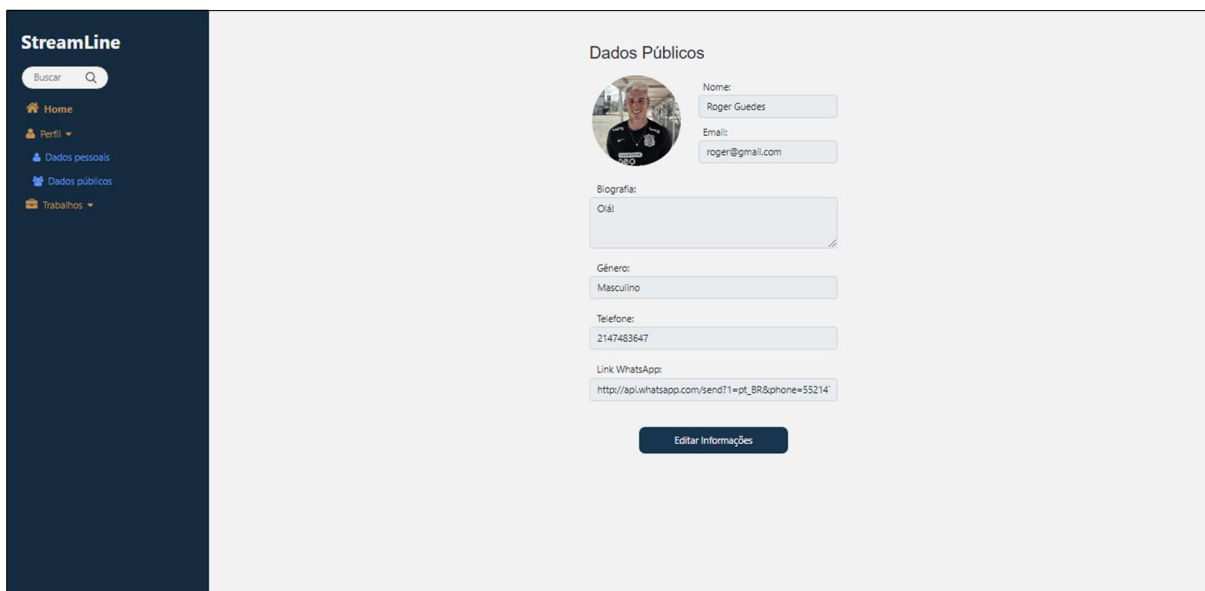
The screenshot shows the 'Dados Pessoais' (Personal Data) section of a user profile in the StreamLine system. The left sidebar contains navigation options: Home, Perfil, Dados pessoais, Dados públicos, and Trabalhos. The main content area displays the following fields:

- Email:** roger@gmail.com
- Senha:** [Redacted]
- Nome:** Roger Guedes
- CEP:** 08245-000
- Cidade:** 43
- Estado:** SP
- N°:** [Redacted]
- Complemento:** [Redacted]

An 'Editar Informações' (Edit Information) button is located at the bottom of the form.

Fonte: Autoria própria, 2022.

A Figura 96 mostra a tela dos dados públicos do cliente, que se refere a foto, nome, e-mail, biografia, gênero, telefone e link do *WhatsApp*.

**Figura 96 - Perfil Cliente: Dados Públicos**

The screenshot shows the 'Dados Públicos' (Public Data) section of a user profile in the StreamLine system. The left sidebar contains navigation options: Home, Perfil, Dados pessoais, Dados públicos, and Trabalhos. The main content area displays the following fields:

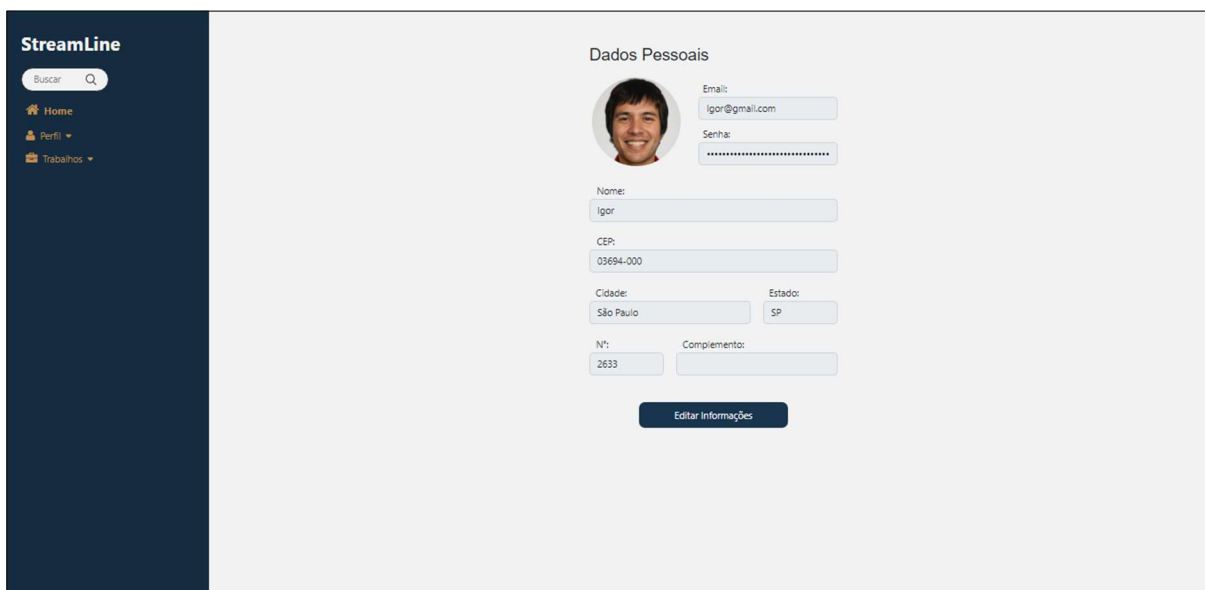
- Nome:** Roger Guedes
- Email:** roger@gmail.com
- Biografia:** Olá!
- Gênero:** Masculino
- Telefone:** 2147483647
- Link WhatsApp:** [http://api.whatsapp.com/send?1=pt\\_BR&phone=55214](http://api.whatsapp.com/send?1=pt_BR&phone=55214)

An 'Editar Informações' (Edit Information) button is located at the bottom of the form.

Fonte: Autoria própria, 2022.

Os dados pessoais do autônomo, mostrados na Figura 97, assim como os do cliente, informam o e-mail, senha, nome, cep, cidade, estado, número da residência e o complemento, sendo possível a alteração deles.

Figura 97 - Perfil Autônomo: Dados Pessoais



The screenshot shows the 'Dados Pessoais' (Personal Data) section of a user profile on the StreamLine platform. The left sidebar contains navigation options: Home, Perfil, and Trabalhos. The main content area displays the following fields:

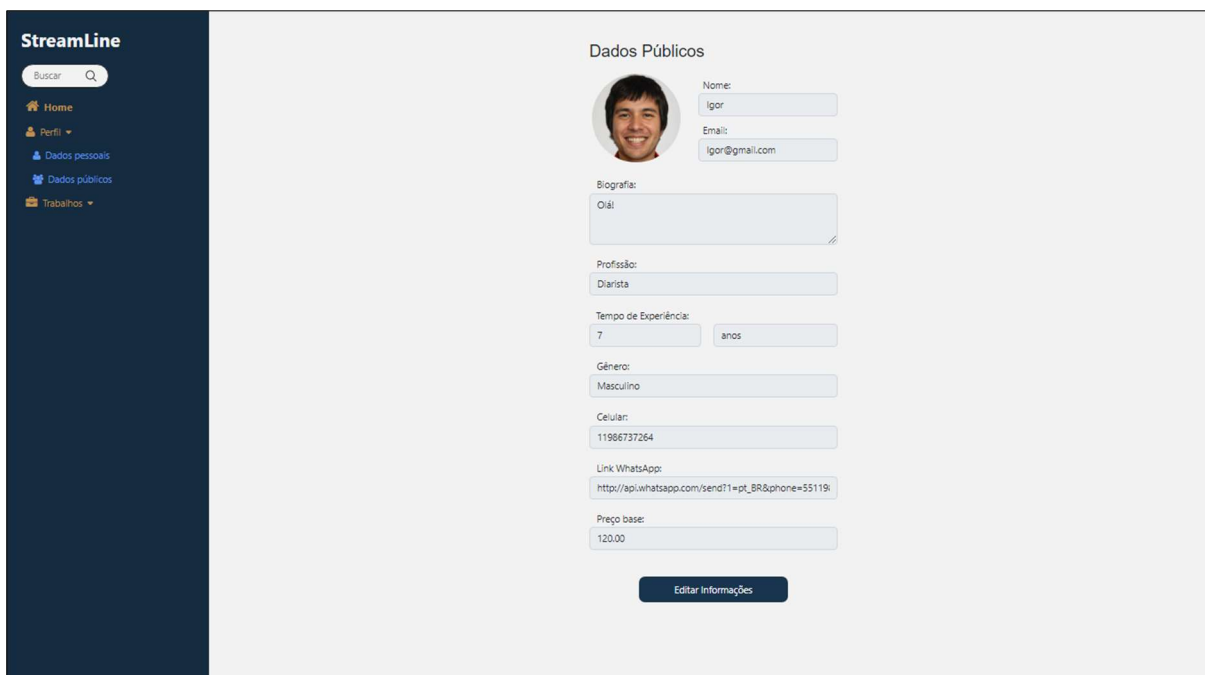
- Nome:** Igor
- CEP:** 03694-000
- Cidade:** São Paulo
- Estado:** SP
- Nº:** 2633
- Complemento:** (empty)
- Email:** Igor@gmail.com
- Senha:** (masked with dots)

An 'Editar Informações' (Edit Information) button is located at the bottom of the form.

Fonte: Autoria própria, 2022.

Na Figura 98, é possível observar os dados públicos do autônomo, que se resumem na foto, nome, e-mail, biografia, tempo de experiência, especificação do tempo, gênero, celular, link do *WhatsApp*, e preço base.

Figura 98 - Perfil Autônomo: Dados Público



The screenshot shows the 'Dados Públicos' (Public Data) section of a user profile on the StreamLine platform. The left sidebar contains navigation options: Home, Perfil, Dados pessoais, Dados públicos, and Trabalhos. The main content area displays the following fields:

- Nome:** Igor
- Email:** Igor@gmail.com
- Biografia:** Olá
- Profissão:** Diarista
- Tempo de Experiência:** 7 anos
- Gênero:** Masculino
- Celular:** 11986737264
- Link WhatsApp:** [http://api.whatsapp.com/send?l=pt\\_BR&phone=55119](http://api.whatsapp.com/send?l=pt_BR&phone=55119)
- Preço base:** 120.00

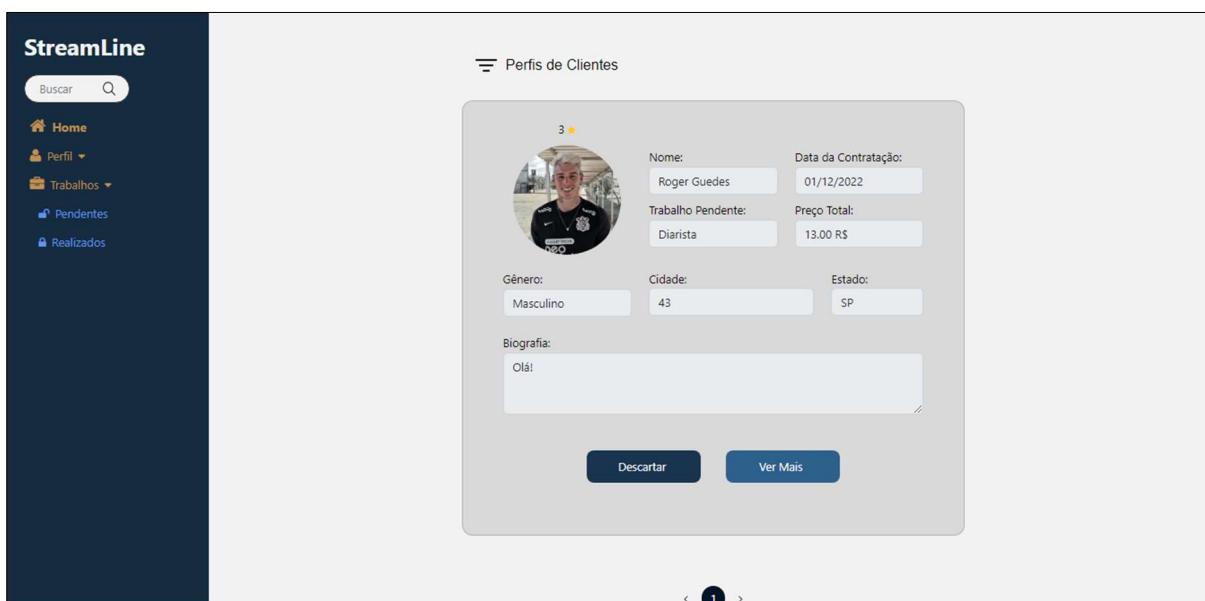
An 'Editar Informações' (Edit Information) button is located at the bottom of the form.

Fonte: Autoria própria, 2022.

### 3.4.5 Trabalhos

Os trabalhos são separados em pendentes e realizados, é demonstrado na Figura 99 a tela de trabalhos pendentes na visão do autônomo, nesta tela estarão os dados do cliente e do serviço requisitado, sendo eles a média de avaliações, a foto, nome, data de contratação, o trabalho pendente, preço total, gênero, cidade, estado e a biografia. Tendo a opção de filtrar os perfis, descartar o serviço ou de ver mais sobre o cliente.

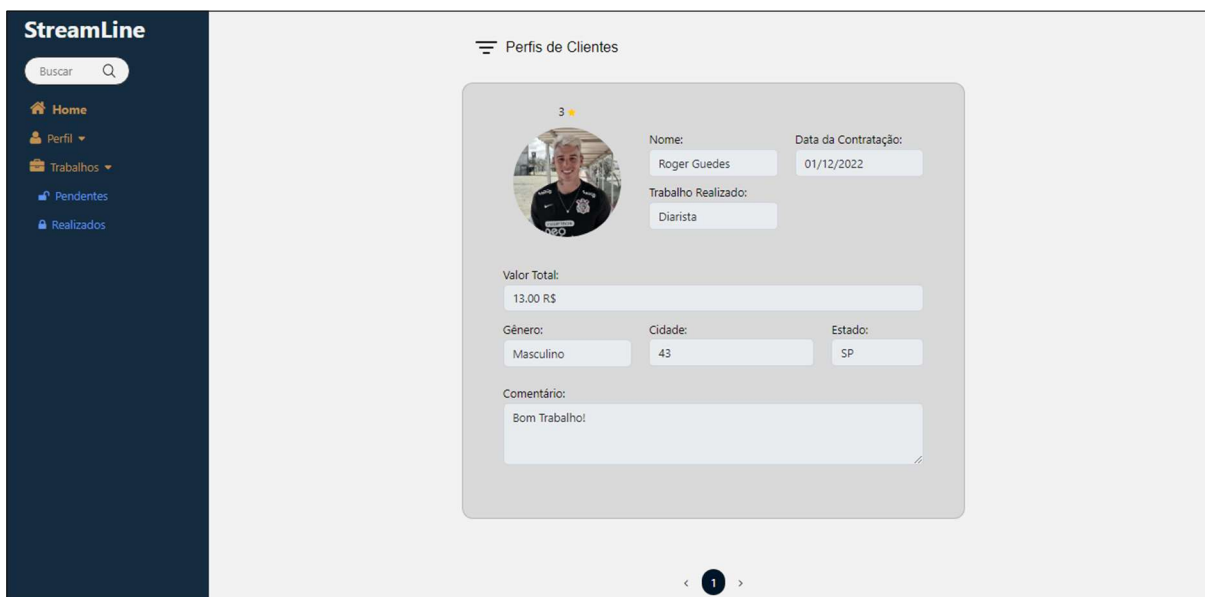
**Figura 99 - Trabalhos Pendentes: Visão como Autônomo**



Fonte: Autoria própria, 2022.

Na Figura 100 é possível observar os trabalhos realizados na visão do autônomo, nela é possível ver as informações dos clientes, nos quais os serviços já foram prestados, sendo elas a média de avaliações, a foto, nome, data de contratação, o trabalho realizado, valor total, gênero, cidade, estado e o comentário da avaliação. Tendo a opção de filtrar os perfis.

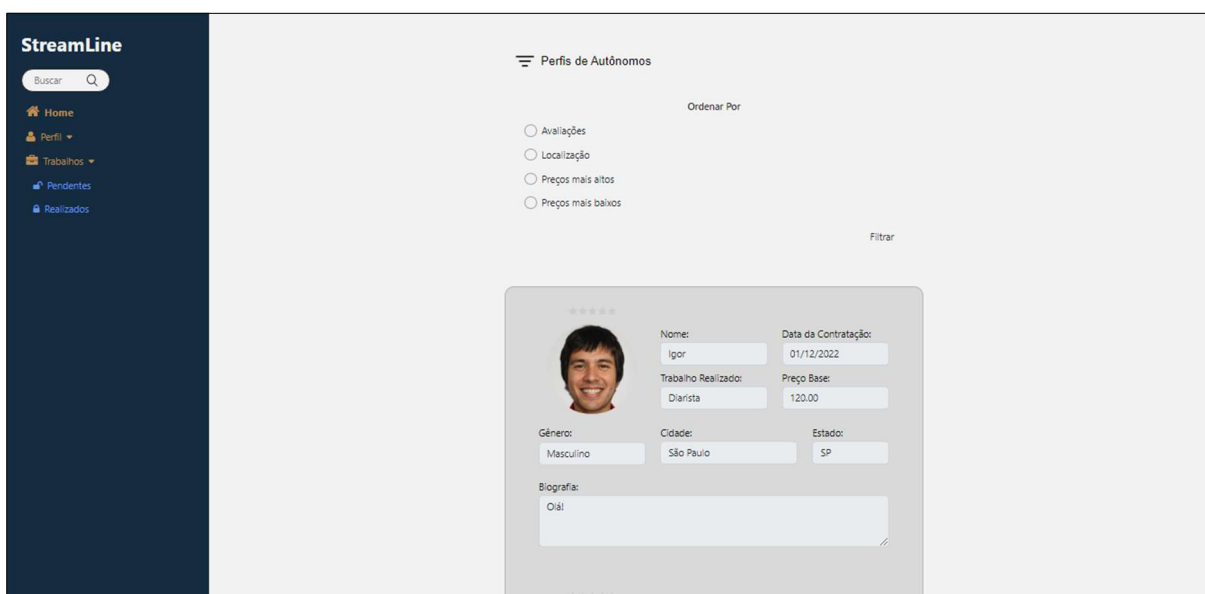
**Figura 100 - Trabalhos Realizados: Visão como Autônomo**



Fonte: Autoria própria, 2022.

Na visão do cliente, os trabalhos também são separados em pendentes e realizados, como exibe a Figura 101, a tela de trabalhos realizados na visão do cliente mostra os dados dos autônomos e dos serviços finalizados, sendo eles a média de avaliações, a foto, nome, data de contratação, o trabalho realizado, preço base, gênero, cidade, estado e a biografia. Tendo a opção de filtrar os perfis por avaliações, localização, preços mais altos e mais baixos.

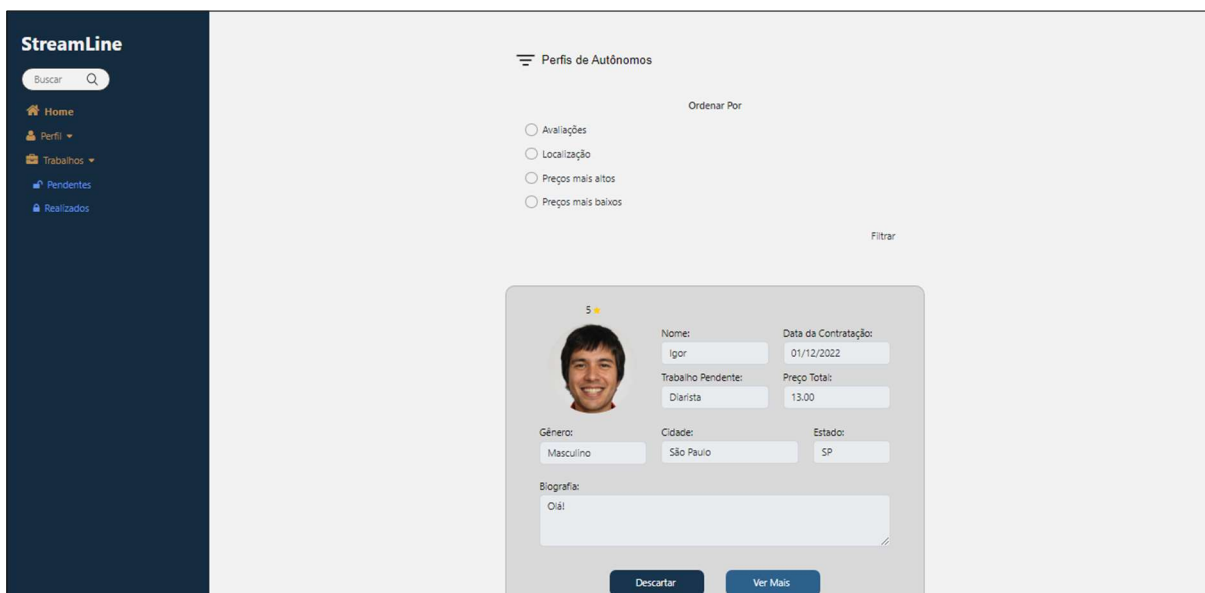
**Figura 101 - Trabalhos Realizados: Visão como Cliente**



Fonte: Autoria própria, 2022.

É possível ver, na Figura 102, os trabalhos pendentes na visão do cliente, que consiste nos serviços ainda não finalizados, sendo eles a média de avaliações, a foto, nome, data de contratação, o trabalho pendente, preço total, gênero, cidade, estado e a biografia. Tendo a opção de filtrar os perfis, descartar o autônomo ou de ver mais sobre ele.

**Figura 102 - Trabalhos Pendentes: Visão como Cliente**



Fonte: Autoria própria, 2022.

Ao clicar em ver mais, o cliente é levado a página onde serão mostrados os detalhes sobre o autônomo e o serviço, como ilustra a Figura 103, onde é possível ver a média de avaliações, a foto do profissional, nome, biografia, gênero, a profissão que o cliente requisitou, e-mail, especialidade, telefone, preço base, tempo de serviço, data de cadastro, idade, data de contratação, cidade e estado. Possuindo um ícone do WhatsApp, que é o link do WhatsApp do Autônomo.

**Figura 103 - Dados do Autônomo Parte 1: Visão como Cliente**

StreamLine  
 Perfil

Perfis de Autônomos

0 ★

Nome do Profissional:  
Igor

Biografia:  
Olá!

Gênero:  
Masculino

Profissão Requisitada:  
Diarista

Email:  
igor@gmail.com

Especialidade:  
anos

Telefone:  
11906737264

Preço base:  
120,00

Whatsapp:

Tempo de serviço:  
2,00

Data de Cadastro:  
05/12/2022

Idade:  
34 anos

Data da Contratação:  
01/12/2022

Cidade:  
São Paulo

Estado:  
SP

Ver Mais

Fonte: Autoria própria, 2022.

Rolando a tela para baixo é possível ver uma avaliação que o autônomo recebeu, ilustrado na Figura 104, que consiste na média do cliente que o avaliou, no nome, data da realização do trabalho e comentário, tendo a opção de ver mais avaliações, cancelar e voltar a página anterior ou finalizar o serviço.

**Figura 104 - Dados do Autônomo Parte 2: Visão como Cliente**

Data da Contratação:  
01/12/2022

Cidade:  
São Paulo

Estado:  
SP

Avaliações: [Ver Mais](#)

5 ★

Nome:  
Roger Guedes

Realização do Trabalho:  
01/12/2022

Comentário:  
Bom serviço

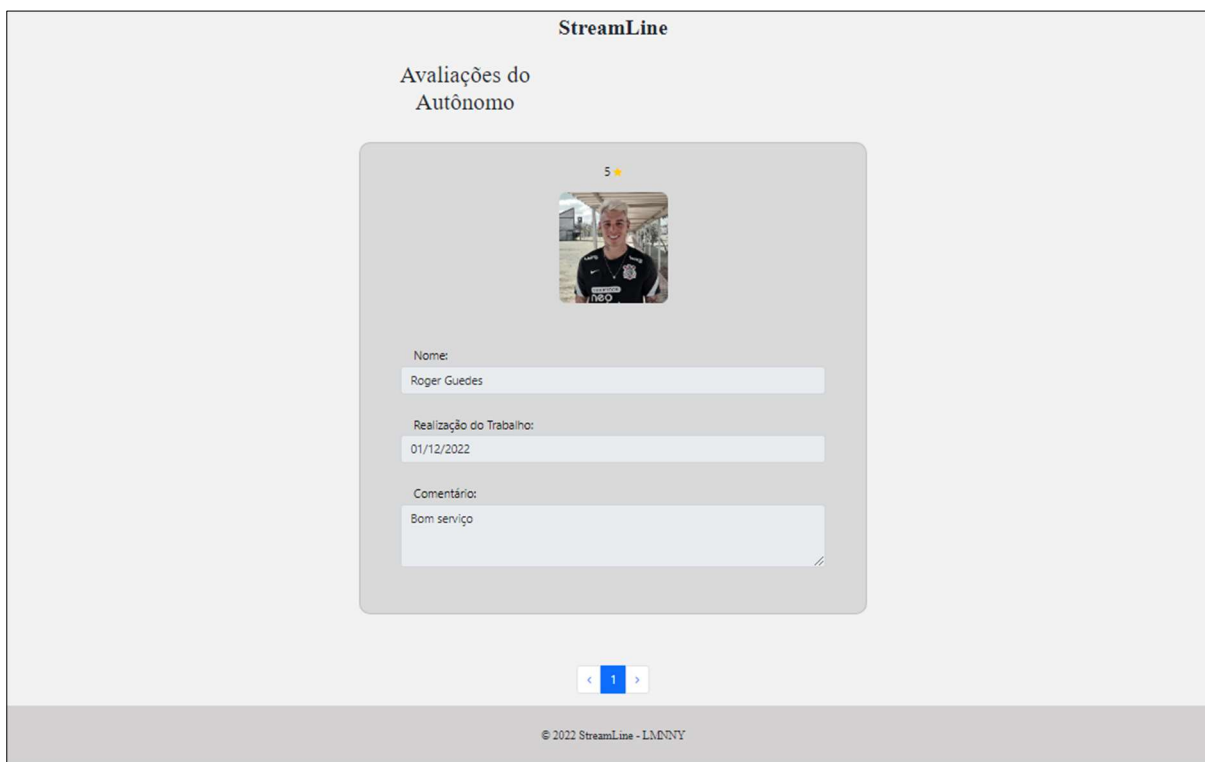
Cancelar [Finalizar Serviço](#)

© 2022 StreamLine - LMDNY

Fonte: Autoria própria, 2022.

Ao clicar em ver mais, o cliente é redirecionado a uma página com as avaliações do autônomo, que consiste na média do cliente que o avaliou, na foto, nome, data da realização do serviço e no comentário, como mostra a Figura 105.

**Figura 105 - Dados da Avaliação do Autônomo: Visão como Cliente**



Fonte: Autoria própria, 2022.

Se o Cliente selecionar para finalizar o serviço, ele é direcionado a página de avaliação, demonstrada na Figura 106, que contém um botão para retornar a página anterior, botão para denunciar e para finalizar a avaliação. Os dados informados sobre o autônomo e o serviço são: foto, nome, data da contratação, profissão requisitada, especialidade, tempo de serviço e o valor do orçamento. As informações necessárias para avaliar o profissional são: quantidade de estrelas e o comentário sobre o serviço prestado.



**Figura 106 - Avaliar Autônomo: Visão como Cliente**

StreamLine

Perfil

**Avaliar Profissional** Voltar

Avalie este serviço para poder contratar outros

Nome do Profissional: Igor

Data da Contratação: 01/12/2022

Profissão Requisitada: Estética

Especialidade: Limpeza de Pele

Tempo de serviço: 2

Valor do Orçamento: 122.00

Selecione a quantidade de estrelas: \*

Deixe seu comentário:

Denunciar

Finalizar Avaliação

Fonte: Autoria própria, 2022.

Caso o cliente queira denunciar o autônomo, ele irá para a página de denúncia, apresentada na Figura 107, que consiste no motivo da denúncia e na reclamação. A página possui os dados do autônomo, que são: foto, nome, data da contratação, profissão requisitada, especialidade, tempo de serviço e o valor do orçamento, tendo um botão para retornar a página anterior e o de finalizar denúncia.

**Figura 107 - Denunciar Autônomo: Visão como Cliente**

StreamLine

Perfil

**Denunciar Profissional** Voltar

Faça sua denúncia com responsabilidade

Nome do Profissional: Igor

Data da Contratação: 25/11/2022

Profissão Requisitada: Estética

Especialidade: Limpeza de Pele

Tempo de serviço: 3:00

Valor do Orçamento: 122.00

Qual é o motivo da denúncia?

Deixe aqui sua reclamação:

Finalizar Denúncia

Fonte: Autoria própria, 2022.

Caso o profissional selecione para finalizar o serviço, assim como para o cliente, ele é direcionado a página de avaliação, como ilustra a Figura 108, que informa os dados sobre o cliente e o serviço, sendo eles: foto, nome, data da contratação, profissão requisitada, especialidade, tempo de serviço e o valor do orçamento. As informações necessárias para a avaliar o cliente são: quantidade de estrelas e o comentário sobre o serviço prestado. Depois do preenchimento ele poderá finalizar a avaliação, ou denunciar o cliente, ou clicar para retornar a página anterior.

**Figura 108 - Avaliar Cliente: Visão como Autônomo**

A imagem mostra a interface de avaliação de cliente no sistema StreamLine. No topo, há o logotipo 'StreamLine' e o ícone de perfil. O título principal é 'Avaliar Cliente', com um botão 'Voltar' à direita. Abaixo do título, há uma instrução: 'Avalie este cliente para voltar a aceitar serviços'. O formulário de avaliação contém:

- Foto do cliente:
- Nome do Cliente: Roger Guedes
- Data de Contratação: 22/11/2022
- Profissão Requisitada: Estética
- Especialidade: Limpeza de Pele
- Tempo de serviço: 3:00
- Valor do Orçamento: 125.00
- Seleção de estrelas: 5 estrelas (5 pontos)
- Deixe seu comentário: Campo de texto para o comentário
- Botões: 'Denunciar' e 'Finalizar Avaliação'

Na base da página, há o rodapé: © 2022 StreamLine - L.MENNY

Fonte: Autoria própria, 2022.

Se o autônomo quiser denunciar o cliente, ele será direcionado a página de denúncia, exibida na Figura 109, onde preencherá o motivo da denúncia e a reclamação. A página também possui os dados do cliente, que consistem em: foto, nome, data da contratação, profissão requisitada, especialidade, tempo de serviço e o valor do orçamento, tendo um botão para retornar a página anterior e o de finalizar denúncia.

**Figura 109 - Denunciar Cliente: Visão como Autônomo**

The screenshot shows a web interface for reporting a client. At the top, it says 'StreamLine' and 'Perfil'. The main heading is 'Denunciar Cliente' with the subtitle 'Faça sua denúncia com responsabilidade'. A 'Voltar' button is in the top right. The form contains a client profile picture, name 'Roger Guedes', and contract date '25/11/2022'. It lists requested profession 'Estética', specialty 'Limpeza de Pele', service time '3.00', and budget value '125.00'. A text area for the complaint is labeled 'Deixe aqui sua reclamação:'. A 'Finalizar Denúncia' button is at the bottom right. The footer reads '© 2022 StreamLine - L.MENY'.

StreamLine

Perfil

**Denunciar Cliente**

Faça sua denúncia com responsabilidade

Voltar

Nome do Cliente:  
Roger Guedes

Data da Contratação:  
25/11/2022

Profissão Requisitada:  
Estética

Qual é o motivo da denúncia?

Especialidade:  
Limpeza de Pele

Deixe aqui sua reclamação:

Tempo de serviço:  
3.00

Valor do Orçamento:  
125.00

Finalizar Denúncia

© 2022 StreamLine - L.MENY

Fonte: Autoria própria, 2022.

## 4 CONCLUSÃO

Com o entendimento sobre UML e desenvolvimento de estudo foi possível realizar um planejamento de forma completa do projeto. A abordagem relacional e os diagramas de caso de uso, atividade e estado tornaram possível entender de forma clara e objetiva a lógica do sistema de site StreamLine. Com o levantamento de requisitos e dicionário de dados, conseguimos realizar a gestão de todas as tabelas e atributos de nosso banco de dados, prezando pela segurança e armazenamento de todos os dados de usuários.

Considerando a conexão entre o prestador de serviço e o cliente, disponibilizamos dados legítimos oferecidos pelo usuário, com a autorização do próprio, afim de que a busca pelo trabalhador autônomo seja mais fácil, por meio de características, habilidades e preferências. Por questões de segurança, estabelecemos dados privados que serão disponibilizados apenas para o usuário que os forneceu, a fins de contato, caso tenha problemas futuros.

Com as linguagens de programação, desenvolvemos na prática as funções do site, proporcionando uma experiência satisfatória para nossos usuários, permitindo que os mesmos possam usufruir da facilidade de disponibilizar o serviço que carece.

Com plataformas para design digital e habilidades artística criamos nossa mascote e também logo, Ely, que concede a harmonia e graciosidade de nosso site StreamLine.

Observando a situação problema de indivíduos que sentem dificuldade em buscar profissionais autônomos, e profissionais com problemas empregatícios, podemos concluir que nosso site *web* auxiliará esses usuários a solucionarem grande parte desse problema, evitando a busca de profissionais por meios lentos, através de comunicação verbal com familiares ou conhecidos até que a informação chegue ao remetente. Pretendemos continuar avançando com o projeto StreamLine futuramente, por acreditarmos fielmente na eficiência e crescimento de StreamLine.

## REFERÊNCIAS

- ALBINO, João Pedro et al. DESIGN DE INTERFACES PARA WEB BASEADOS NO SISTEMA DE GRADE DO BOOTSTRAP 3 INTERFACE DESIGN FOR WEB BASED ON GRID SYSTEM BOOTSTRAP 3. 2015.
- ÁVILA, Michel Leite de. Uma Ferramenta de Apoio à Normalização de Tabelas Baseada na Análise dos Dados. 2007.
- BARBOSA, Eduardo Pinto; DE ALVARENGA, Geoflavia Guilarducci. Uma Proposta para Ensino de Dicionário de Dados em Projetos de Bancos de Dados.
- BOOCH, Grady. UML: guia do usuário. Elsevier Brasil, 2006.
- CHICHINELLI, Micheli. A IMPORTÂNCIA DAS TÉCNICAS DE LEVANTAMENTO DE REQUISITOS NO PROCESSO DE DESENVOLVIMENTO DE SOFTWARE. Revista Empreenda UniToledo Gestão, Tecnologia e Gastronomia, v. 1, n. 1, 2017.
- COELHO, Luis Fernando. Gerador de código HTML baseado em dicionário de dados utilizando banco de dados. 2006.
- DALL'OGGIO, Pablo. PHP Programando com orientação a Objetos. Novatec Editora, 2018.
- DAS NEVES CÂMARA, Álvaro Luiz Panarra; PINHEIRO, Rômulo Silva; ALMEIDA, Paulo Roberto Bastos. GERAÇÃO AUTOMÁTICA DE CÓDIGO COM BASE NO DIAGRAMA DE CLASSE DA UML. Revista Traços, v. 13, n. 27, 2017.
- DATE, Christopher J. Introdução a sistemas de bancos de dados. Elsevier Brasil, 2004.
- DE ALMEIDA, Vinicius Coelho. Uso da linguagem OCL no contexto de diagramas de classe da UML e programas em java. 2006.
- DE MELLO, Rafael Maiani. Técnica para Inspeção de Diagramas de Atividades. 2011. Tese de Doutorado. Universidade Federal do Rio de Janeiro.
- DIMES, Troy. JavaScript: Um Guia para Aprender a Linguagem de Programação JavaScript. Babelcube Inc., 2015.
- DURÃES, Amanda Dias Oliveira et al. Ferramenta para o auxílio da aprendizagem no paradigma de orientação a objetos. 2021.
- EIS, Diego; FERREIRA, Elcio. HTML5 e CSS3 com farinha e pimenta. Lulu. com, 2012.
- ELMASRI, Ramez et al. Sistemas de banco de dados. 2005.
- FERNANDES, Lúcio Alves. Uma linguagem de modelagem e uma ferramenta CASE para apoiar o projeto lógico de banco de dados relacionais. 2017. Dissertação de Mestrado. Universidade Federal de Pernambuco.
- FERREIRA, Jeferson; MARTINS, Eliane. Fluxo de Exceções Intraprocedimentais a partir do Diagrama de Atividades da UML 2.0. Instituto de Computação Universidade Estadual de Campinas, Campinas, São Paulo, 2010.
- FOWLER, Martin. UML Essencial: um breve guia para linguagem padrão. Bookman editora, 2014.
- FRANÇA, Cicero Tadeu Pereira Lima; JÚNIOR, Joaquim Celestino. Banco de Dados. 2014.

GONZAGA, Flávio S.; BIRCKAN, Guilherme. Curso de PHP e MySQL. Florianópolis, outubro, 2000.

GUEDES, Gilleanes TA. UML 2-Uma abordagem prática. Novatec Editora, 2018.

JARIA, Amanda Macedo et al. IFISIO: UM APLICATIVO DE GERENCIAMENTO DE CLÍNICAS FISIOTERÁPICAS COM O FOCO NA EVOLUÇÃO DO PACIENTE PEDIÁTRICO.

LARMAN, Craig. Utilizando UML e padrões. Bookman Editora, 2000.

LISBOA FILHO, Jugurta; IOCHPE, Cirano. Modelagem de Bancos de Dados Geográficos. In: Anais do XX Congresso Brasileiro de Cartografia. 2001.

MACÁRIO, Carla Geovana do N.; BALDO, Stefano Monteiro. O modelo relacional. Instituto de Computação Unicamp. Campinas, p. 1-15, 2005.

MACÁRIO, CG do N.; CHAIM, Marcos. Uso da linguagem de especificação SDL como alternativa ao diagrama de estados proposto pela linguagem UML. Embrapa Informática Agropecuária, 2003.

MACHADO, Felipe Nery Rodrigues. Análise e Gestão de Requisitos de Software—Onde nascem os sistemas. Saraiva Educação SA, 2018.

MAZZA, Lucas. HTML5 e CSS3: domine a web do futuro. Editora Casa do Código, 2014.

NAKAZATO, Karen Miyuki. TÉCNICA DE LEITURA PARA INSPEÇÃO DE DIAGRAMAS DE ESTADOS COM BASE EM DIAGRAMAS DE ATIVIDADES ESPECIFICANDO OS CASOS DE USO DO SOFTWARE. 2013. Tese de Doutorado. Universidade Federal do Rio de Janeiro.

OLIVEIRA, George Moreno de. Desenvolvimento e avaliação do plugin para o Figma para Documentação de Acessibilidade para Interfaces-DAI. 2022.

PRESCOTT, Preston. Programação em JavaScript. Babelcube Inc., 2016.

RAMAKRISHNAN, Raghu; GEHRKE, Johannes. Sistemas de gerenciamento de banco de dados-3. AMGH Editora, 2008.

ROB, Peter; CORONEL, Carlos. Sistemas de banco de dados. Projeto, implementação e, 2011.

SANTANCHÈ, André; CAVOTO, Patrícia. Normalização. 2013.

STEIN JÚNIOR, Altair Von; MALUCELLI, Andreia; BASTOS, Laudelino Cordeiro. Especificação de sistema de informação de microáreas de risco utilizando a abordagem orientada a objetos. Revista Eletrônica de Enfermagem, v. 11, n. 4, p. 866-76, 2009.

SILVA, André Luiz. G-LAB. 2009.

SILVA, Arthur de Almeida Pereira da. Design Responsivo: técnicas, frameworks e ferramentas. 2014.

SILVA, Maurício Samy. CSS3: desenvolva aplicações web profissionais com uso dos poderosos recursos de estilização das CSS3. Novatec Editora, 2019.

SILVA, Maurício Samy. Fundamentos de HTML5 e CSS3. Novatec Editora, 2015.

SILVA, Maurício Samy. JavaScript-Guia do Programador: Guia completo das funcionalidades de linguagem JavaScript. Novatec Editora, 2010.

SOUZA, Glaycow Silveira Silva e. DESENVOLVIMENTO DE UMA FERRAMENTA PARA ANÁLISE E EXTRAÇÃO DE MÉTRICAS DE QUALIDADE DE DIAGRAMAS DE CLASSE UML BASEADO NO QUALITY MODEL OBJECT ORIENTED DESIGN. 2018. Orientado por Professor M. Sc. Jackson Gomes de Souza.

TORRES, Victor Monteiro. HTML e seus Componentes. Revista Ada Lovelace, v. 2, p. 99-101, 2018.